CONTRIBUTIONS TO THE
MECHANISMS AND SIMULATION OF
MINERAL PROCESSING OPERATIONS

BY

JAN STEPHANUS JAKOB VAN DEVENTER

A dissertation submitted in fulfilment of the requirements for the
degree of Doctor of Engineering at the University of Stellenbosch

Promoter: Prof. L. Lorenzen

The University of Stellenbosch
August 1999
To Tiny, George and Ben

for their love and support

during the 19 years

of this research.
Declaration

I hereby certify that this dissertation is my own original work, except where specifically acknowledged in the text. Although the published papers contained in this submission have been produced in collaboration with co-workers, my own contribution has been stated explicitly in each case. Some of the research has been submitted in part as theses by co-workers, but the collection of papers in their published form have not been submitted at any other university.

J.S.J. VAN DEVENTER

27 August 1999
Abstract

This collection of 101 published papers is based on 19 years of research, mainly in South Africa and recently also in Australia. The main thrust of this work was the simulation of mineral processing operations where conventional modelling was inadequate owing to their ill-defined nature. These papers presented some of the first applications of knowledge based systems and neural networks to mineral processing problems.

A new methodology was proposed for modelling ill-defined kinetic processes by relating rates to process conditions via non-parametric methods. The kinetic and equilibrium parameters could be related to adjustment factors if reference conditions changed. These methods were applied to batch and continuous flotation, leaching and adsorption systems as well as pyrometallurgical processes. In the case of continuous process data intrinsic reaction rates could be back-calculated and then related to process conditions via a neural net. It was explained how the configuration of mineral processing circuits could be optimised by a two-stage linear programming method where the constraints were determined interactively by a knowledge based system or neural network. In the case of non-linear system constraints neural nets were shown to detect and locate gross errors efficiently in material balancing problems. Several novel hybrid neural net architectures were proposed which allowed the integration of parametric fundamental knowledge with non-parametric heuristic knowledge. Furthermore, it was shown how neural nets could be used to extract knowledge from historical process data records.

Pioneering work was conducted on the application of textural image analysis to flotation froth surfaces and ore particles on a conveyor belt. It was possible to relate image features to metallurgical performance via neural net architectures. Different types of topological maps were used to classify froth types, to track the profile of changes in process and flotation conditions, and to relate froth class to concentrate grade and recovery. Perturbations in the comminution circuit were reflected in perturbations in the froth image features and metallurgical performance. It was also shown that the spray
angle of a hydrocyclone as determined by image analysis could be used as input to a soft sensor for predicting size distribution.

A model was developed to combine transport phenomena in the froth phase with fluid flow behaviour in order to optimise mechanical cells and flotation columns. Rate constants for the various sub-processes were determined by fitting models to concentration profiles measured at industrial plants. It was shown that the behaviour of the froth phase during the flotation of sulphides, gold and uranium was affected by galvanic interaction between metallic iron and the individual mineral species. A conceptual model was developed for the flotation behaviour of free gold in the presence of refractory sulphides for different oxidative conditions and different sequences of reagent addition.

The use of diagnostic leaching to evaluate the efficiency of mineral processing operations was explained in detail. A first attempt was made to relate diagnostic leaching data to mineral liberation via semi-empirical equations and neural nets. It was explained how the liberation patterns of different types of gold ore could be distinguished by using topological maps. The mechanisms of selected sundry processes such as electrokinetic solid-liquid separation, flow splitting, induced aeration and jet reactors were also investigated.
Samevatting

Hierdie versameling van 101 publikasies is gebaseer op navorsing oor 19 jaar, hoofsaaklik in Suid-Afrika en onlangs ook in Australië. Die fokuspunt van hierdie werk is die simulasiest van mineraal prosesserings operasies waar konvensionele modellering ontoereikend is vanweë swak gedefinieerdheid. Hierdie publikasies toon van die heel eerste toepassings van kennis gebaseerde stelsels en neurale netwerke op mineraal prosesserings probleme.

'n Nuwe metodiek is voorgestel vir die modellering van swak gedefinieerde kinetiese prosesse deur die reaksie tempo’s te verbind met proses toestande via nie-parametriese tegnieke. Die kinetiese en ekwilibrium parameters is in verband gebring met verstellingsfakte indien die verwydingstoestande verander het. Hierdie metodes is toegespas op enkellading en kontinue flottasie, loging en adsorpsie stelsels, asook op pirometallurgiese prosesse. In die geval van kontinue proses data kon die intrinsieke reaksie snelhede terugberekend word en in verband gebring word met proses toestande via 'n neurale net. 'n Twee stadium lineêre programmerings metode is voorgestel vir die optimale ontwerp van mineraal prosesserings aanlegte deur die beperkings interaktief te bepaal met 'n kennis gebaseerde stelsel of neurale netwerk. In die geval van nie-lineêre stelsel beperkings is getoon dat neurale nette effektief buitegewone foute kon uitken en opspoor in massabalans probleme. Verskeie nuwe hibriede neurale net argitekture is voorgestel om parametriese fundamentele kennis te integreer met nie-parametriese heuristiese kennis. Verder is getoon hoe neurale nette gebruik kan word om kennis te ekstraheer uit historiese rekords van proses data.

Pioniers werk is gedoen om tekstuur gebaseerde beeldanalise tegnieke toe te pas op flottasie skuim oppervlaktes en erts partikels op 'n vervoerband. Dit is moontlik om beeld kenmerke in verband te stel met metallurgiese doel treffendheid via neurale netwerk argitekture. Verskeie topologiese afbeeldings is gebruik om skuim tipes te klasifiseer, om die profiele van veranderings in proses en flottasie toestande te volg, asook om die skuim klas in verband te bring met herwinning en konsentraat graad. Versteurings in die
groottereduksie aanleg is afgebeeld deur ooreenkomstige versteurings in die skuim kenmerke en metallurgiese gedrag. Daar is ook getoon hoe die sproei hoek van 'n hidrosikloon soos bepaal deur beeldanalise, gebruik kan word as inset tot 'n sagte sensor om partikelgrootte verspreiding te voorspel.

'n Model is ontwikkel om oordragsverskynsels in die skuimfase te combineer met vloeipatrone om sodoende mekaniese flottasie selle en kolomme te optimeer. Snelheidskonstantes vir die verskeie sub-prosesse is bepaal deur modelle te pas op konsentrasie profiele soos gemeet op industriële aanlegte. Daar is getoon dat die gedrag van die skuim fase in die flottasie van sulfiede, goud en uraan beïnvloed word deur galvaniese interaksie tussen yster metaal en die onderskeie mineraal spesies. 'n Konseptuele model is voorgestel vir die flottasie gedrag van vrye goud in die teenwoordigheid van weerbarstige sulfiedes onder verschillende toestande van oksidasie en volgorde van reagens byvoeging.

'n Gedetailleerde verduideliking is gegee vir die aanwending van diagnostiese loging om die doeltreffendheid van mineraal prosesserings operasies te evalueer. Diagnostiese loging data is vir die eerste keer in verband gebring met mineraal vrystelling met semi­-empiriële vergelykings en neurale nette. Daar is verduidelik hoe die vrystellingspatrone van verskeie goud erts tipes onderskei kan word deur 'n topologiese afbeelding. Die werking van verskeie prosesse soos elektrokinetiese vloei/vastestof skeiding, vloei verdeling, geïnduseerde belugting en spuit reaktore is ook ondersoek.
Acknowledgements

My family has been most supportive of my research career, which has been very rewarding, but which has also involved considerable personal sacrifice. Nobody knows this better than my wife Tiny, who deserves the first place in a long list of acknowledgements. The love, support and patience that Tiny and my sons, George and Ben, have shown over the 19 years of this research work are greatly appreciated. Without their understanding and sacrifice my contribution to science and engineering would have been far less.

I wish to thank Leon Lorenzen for his technical and managerial contribution over many years, and for the fact that he has encouraged me to submit this dissertation. Successful research is possible only when the contributions of team members are integrated. Certain team members play a more pivotal role in contributing ideas and integrating knowledge. Without the original ideas, dedication and hard work by Markus Reuter, Chris Aldrich and Derick Moolman over many years this research would not have reached the same level of advancement. Their efforts to develop artificial intelligence techniques and computer vision systems are gratefully appreciated.

In addition, the contributions and hard work by following co-workers and research students have been substantial and are highly appreciated: Deon Annandale, Keith Anthony (deceased), Etienne Barnard, Morne Bezuidenhout, Gideon Botes, Dee Bradshaw, André Burger, Frikkie Cloete, Robert Dunne, Jacques Eksteen, Craig Goodall, Jim Green, Brent Loftus, Peter Marais, Gerhard Muller, Danie Nieuwoudt, Deon Oosthuizen, Kurt Petersen, Johan Rademan, Victor Ross, Marius Sinclair, Cathy McInnes, Greg Schmitz, Wayne Stange, Sam Swaminathan, Adam Teague, Wimpie van der Merwe, Tjaart van der Walt, Wynand van Dyk, Ross Veitch, Wayne Wilmans.
Most of the research work presented in this submission has been conducted in the Department of Chemical Engineering at the University of Stellenbosch. Since the end of 1995 this work has been continued in the Department of Chemical Engineering at the University of Melbourne. The infrastructural and financial support of both universities is gratefully acknowledged.

Funding from the following companies and institutions since 1981 is especially acknowledged: Anglo American Corporation, Anglovaal, Australian Research Council, De Beers, the former Foundation for Research Development, Gencor, Impala Platinum, Kenwalt, Mintek, Mount Isa Mines, Sasol and Stawell Gold Mines.
# CONTENTS

Declaration i
Abstract ii
Samevatting iv
Acknowledgements vi

1. Scope of Submission 1
2. Personal Role in Research 4
3. List of Publications Included in this Submission 9
4. List of Publications Not Included in this Submission 18
5. General Statement of Impact 37
6. Summary of Submitted Publications 40
   6.1 Process Synthesis 40
   6.2 Knowledge Based Systems 42
   6.3 Neural Networks: Theoretical Concepts 44
   6.4 Neural Networks: Applications 47
   6.5 Computer Vision 50
   6.6 Froth Flotation and Interfacial Phenomena 53
   6.7 Fluid Mechanics 56
   6.8 Diagnostic Leaching 57

7. Publications on Process Synthesis
8. Publications on Knowledge Based Systems
9. Publications on Neural Networks: Theoretical Concepts
10. Publications on Neural Networks: Applications
11. Publications on Computer Vision
12. Publications on Froth Flotation and Interfacial Phenomena
13. Publications on Fluid Mechanics
14. Publications on Diagnostic Leaching
VOLUME I
1. Scope of Submission

In 1981 Jannie van Deventer started a major research programme on mineral processing at the University of Stellenbosch. From the beginning the research has been directed at both mineral processing problems as well as hydrometallurgy and pyrometallurgy. This is unusual, because most researchers specialise only in one of these areas. The main unit operation studied in mineral processing has been froth flotation, with an emphasis on the froth phase rather than the pulp phase as investigated by most other researchers. Adsorption onto activated carbon and leaching of gold ores have been the main focus of the hydrometallurgical research, while the pyrometallurgical work has emphasised carbothermic reduction of metal oxides. Both the chemical mechanisms governing these systems as well as their mathematical modelling have been studied.

Even in the early 1980’s it became evident that conventional mathematical modelling techniques have limited value in simulating these complex systems which are to a large extent ill-defined. Often the relationship between variables is convoluted and difficult to establish from industrial process data. It is for this reason that artificial intelligence techniques were developed in collaboration with Markus Reuter at the end of the 1980’s. With the rapid development of computer technology and advances in the field of neural networks the initial work on hybrid knowledge based systems was superseded by the development of special hybrid neural network architectures. These techniques were applied to mineral processing operations, hydrometallurgical unit processes as well as selected high-temperature processes. It was also shown that neural nets could be applied to a wide range of other ill-defined process engineering problems.

In many mineral processing operations such as froth flotation visual information contributes to operational decision making. Nevertheless, at the time no formal method was available to capture this knowledge on-line. The development of computer vision techniques for froth flotation, the spray angle of a hydrocyclone and ore particles on a conveyor belt was consequently a natural extension of the work on artificial intelligence techniques. The thrust of the contribution is encompassed in the integration of textural computer vision methods with neural network architectures.
Multidisciplinary research usually yields new knowledge at the interface of disciplines. As a result of the research on leaching of refractory gold ores and the work on mineral processing, new methods were developed for interpreting diagnostic leaching data in terms of mineral liberation. It was also logical to apply neural nets to identify this ill-defined relationship between diagnostic leaching behaviour and the liberation of the host minerals in terms of particle size distribution.

For the purpose of this submission it was decided to divide the published research into two categories, as listed in Sections 3 and 4. The papers on the mechanisms of mineral processing operations and their modelling are listed in Section 3. Most of these papers discuss the development of artificial intelligence and computer vision techniques specifically to meet the needs of mineral processing problems. However, some hydrometallurgical and high temperature applications are also discussed in these papers in order to demonstrate general applicability. Only the papers listed in Section 3 should be considered to formally constitute this submission. Many of the presentations listed in Section 4 are relevant to this submission, but are not available in a published form and overlap in any case with the papers in Section 3. The papers on hydrometallurgy, pyrometallurgy and waste processing listed in Section 4 do not form part of this submission, but are nevertheless listed in order to give a complete profile of the research by Jannie van Deventer. These papers also help to explain the rationale behind some of the work on artificial intelligence discussed in the submission.

The research papers included in this submission are placed in the following categories: (1) Process synthesis, (2) Knowledge based systems, (3) Neural networks: Theoretical concepts, (4) Neural networks: Applications, (5) Computer vision, (6) Froth flotation and interfacial phenomena, (7) Fluid mechanics, and (8) Diagnostic leaching. This categorisation is not absolute and a certain degree of overlap is inevitable. The content and context of these papers are summarised in Section 6 of this submission. Copies of the actual papers are given within each of the categories mentioned above in Sections 7 to 14. The personal role of Jannie van Deventer in the submitted research is explained in Section 2, especially in view of the fact that all the work has been conducted in collaboration with postgraduate students and co-workers over 19 years. Whereas Section 6 explains the significance of each submitted research paper,
Section 5 summarises the impact of the entire submission in terms of a citation count and the transfer of research results to industry. At the end of Section 4 a list of postgraduate students who have graduated from this research is provided. If nothing else, it is evident that the research has contributed substantially to the advanced education of professionals.
2. Personal Role in Research

Unless stated to the contrary below, all research was planned, initiated and managed by Jannie van Deventer (JvD) in his role as supervisor of research students. Upon graduation many of these ex-students have continued to collaborate with JvD, so that papers have appeared with these ex-students over many years. In the majority of papers included in this submission the actual research work was conducted by the students and co-workers under the guidance of JvD. Unless the entire paper was written by JvD, the student was usually placed as the first author. There are a few papers such as numbers 3, 5, 10, 67, 74 and 75 (listed in Section 3) which were written solely by JvD with the co-workers placed as the first authors. As many of the co-authors of the papers listed in Section 3 were not Anglophone JvD had to do substantial rewriting of many papers. Therefore, the contribution of JvD to most of the listed papers was substantial in terms of planning, structuring, interpretation and editing. Communication with the editors of journals, textbooks and conference proceedings in the process of submission of papers was conducted by JvD for more than 80% of the papers, except for the following papers, where JvD has been closely involved in such communication: 15, 21, 27, 35, 47, 48, 53, 54, 55, 57, 60, 83, 85, 86, 90, 95, 96 and 101 (as listed in Section 3). While the composition of the research team has always been in a state of transition over the 19 years of this work, JvD has clearly provided the leadership and continuity. The respective contributions of JvD and co-workers to ideas in each of the research categories of Section 3 will be explained below.

Process Synthesis

Jim Green initiated the work on the application of linear programming to the optimisation of flotation circuits. In 1984 and 1985 JvD supervised Markus Reuter and the principles eventually published were the result of a true team effort. In 1990 JvD solely proposed and planned the work on gravity separation circuits executed by Keith Anthony. Based on the plant experience of Johan Rademan during his project supervised by JvD and Leon Lorenzen, JvD proposed in 1993 that an off-line simulation programme for the repulp section should be developed. The programming was conducted by Johan Rademan, but the suggestions by JvD were instrumental in achieving a positive outcome.
Knowledge Based Systems

As an extension of the work on process synthesis, JvD and Markus Reuter in a team effort identified the need for artificial intelligence to capture heuristic knowledge in process systems during 1987. The ideas for the overall knowledge based structure were proposed more by Markus Reuter, while JvD played a key role in integrating this structure with processing systems and reaction kinetics during 1990. The use of a knowledge base to determine the constraints in a two-stage linear programming system was a natural sequence to the work on process synthesis, for which JvD and Markus Reuter could claim equal credit. JvD and Markus Reuter developed the generalised kinetic model for incorporation in the knowledge base as a true team effort from 1989 to 1991. The development of a knowledge based structure for gold leaching in columns was proposed and planned by JvD in 1990, although the programming was conducted by Wimpie van der Merwe.

Neural Networks: Theoretical Concepts and Applications

Tjaart van der Walt suggested in 1990 that the knowledge based approach should be replaced by neural nets as part of his thesis. In 1991 Markus Reuter suggested independently to JvD that a major research effort should go into neural nets for reaction kinetic problems. Consequently, Tjaart van der Walt spent time with Etienne Barnard at the University of Pretoria during 1990 and 1991 to learn about cutting edge work on neural networks at that stage. Subsequently, numerous papers have appeared on neural nets in which Markus Reuter played a key role together with JvD in directing research. The generalised neural net kinetic rate equation was a culmination of the knowledge based work and new neural network research, for which Markus Reuter and JvD could claim equal credit. The idea to combine heuristic and fundamental knowledge in a neural net architecture was proposed by JvD, while the development and programming of the specific configuration was done by Tjaart van der Walt. JvD proposed the partitioning of the variable space of high-dimensional systems and the back-calculation of rate information from continuous systems. The idea to apply neural nets to metal-slag equilibrium and other high temperature processes was offered by Markus Reuter.

The research on neural networks progressed rapidly when Chris Aldrich joined the team as a mature PhD student in 1991. JvD proposed that neural nets should be used for material balancing and process synthesis by decoupling of recycle streams in circuits. The associated
work on gross error detection and location was suggested by Chris Aldrich, who developed the theory and various network structures in close collaboration with JvD. All the published applications of neural nets to processing problems were initiated and planned by Chris Aldrich and JvD as a true team effort. JvD solely proposed the use of neural nets for solvent extraction systems and the kinetic modelling of thermal decomposition processes.

\textit{Computer Vision}

When it became evident that more heuristic knowledge should be obtained on-line in order to provide adequate information as input into a neural network for prediction of plant performance, JvD proposed the videographic monitoring of processes. In an exchange of ideas between JvD and Wayne Stange during 1992 it became evident that computer vision of flotation froth was an obvious choice. Wayne Stange then also proposed the videographic monitoring of the spray angle of a hydrocyclone. Derick Moolman joined the team in 1993 as a PhD student and made an enormous contribution regarding the development of image analysis tools and the practical implementation of the technology. Chris Aldrich contributed substantially to the integration of neural net theory with image analysis results. This was such a talented team that less need for reliance on the intellectual input from JvD for rapid progress. Nevertheless, JvD played a key role in guiding Derick Moolman and Chris Aldrich to achieve practical and research goals. JvD proposed the direct application of neural nets to pixel data, while Derick Moolman suggested the use of textural methods for feature extraction and Chris Aldrich proposed the application of topological maps to froth features. The progress in this field was the result of a true team effort between Derick Moolman, Chris Aldrich and JvD.

JvD was responsible for supervising the work of Morne Bezuidenhout on several plants in Australia. JvD has been intimately involved in the interpretation of these data sets. JvD has also proposed the separation of long term shifts from short term fluctuations in metallurgical and computer vision data. Together with Chris Aldrich, JvD supervised the work of Kurt Petersen on image analysis of hydrocyclone underflow and particles on a conveyor belt. JvD was closely involved in the modelling of underflow spray angles and made several suggestions for relating such values to particle size data. Although Kurt Petersen proposed the use of variance and range operators for determining particle size distributions, JvD proposed
the integration of this information with an overall simulation programme for closed circuit comminution.

_Froth Flotation and Interfacial Phenomena_

Robert Dunne suggested the flotation problem investigated by JvD and Victor Ross in 1983 and 1984. Robert Dunne was largely responsible for experimental planning in view of a lack of flotation experience by JvD at that stage. However, together with Victor Ross, JvD played a key role in the interpretation of flotation and electrochemical results. The subsequent re-interpretation of results in terms of galvanic interactions was conducted solely by JvD from 1991 to 1995. JvD was invited in 1996 to supervise the thesis of Adam Teague on the flotation of gold ore at the Royal Melbourne Institute of Technology. From 1996 to 1998 JvD solely proposed the entire research programme investigating the behaviour of free gold versus locked gold, and he also played a major role in the interpretation of results. The actual work was conducted by Adam Teague. The methodology of linking the flotation of uranium to its associated mineralogy was proposed by JvD, although Leon Lorenzen participated in the execution of the project.

In 1983 Frikkie Cloete suggested to JvD that the flotation process could be intensified by sparging air into a hydrocyclone. At the same time Robert Dunne alerted JvD to the work already conducted by Jan Miller of Utah. JvD subsequently decided to continue with the design of Jan Miller and involved Frikkie Cloete in the project. JvD played a pivotal role in interpreting results and modifying the original Miller design. JvD solely planned subsequent work in 1987 and 1988 where the distribution of air along the length of a porous cylinder was modified. While Victor Ross was employed by Mintek, frequent interaction with JvD in 1986 and 1987 led to the concept of modelling the mass transport phenomena in flotation froths in terms of rate processes and flow patterns resulting from mechanical design. The ideas in this work were the result of team work by Victor Ross and JvD, but the actual modelling, programming and practical work were conducted solely by Victor Ross. The design of a moving froth bed to support hydrophilic particles, while hydrophobic particles act as film breakers, was proposed by Victor Ross. Together with Leon Lorenzen, JvD supervised this project of Wynand van Dyk. JvD played a key role in research planning, equipment design and the subsequent modelling of results not yet published.
The work on electrokinetic solid/liquid separation was proposed by staff from De Beers Research Laboratories in 1984, but the project was supervised entirely by JvD. The principles of modelling electro-osmotic dewatering were developed by JvD, but the computational work was conducted by Markus Reuter.

Fluid Mechanics

During the Master’s research by André Burger in 1984 it became evident that segregation of coarse particles during flow-splitting was a problem. It was JvD’s initiative that this problem was investigated and he also proposed a final practical solution. The work on induced aeration was initiated and planned entirely by JvD in 1983, but executed by Chris Aldrich. In 1993 JvD proposed and designed a jet reactor system for gold leaching, which was subsequently used by Leon Lorenzen, JvD and co-workers in a test programme. The concept of the high-intensity gas-liquid jet reactors investigated by Gideon Botes was developed solely by JvD in 1994. JvD also played a key role in the interpretation of the mass transfer results.

Diagnostic Leaching

The concept of diagnostic leaching was developed by Leon Lorenzen in collaboration with co-workers at the Anglo American Research Laboratories in the mid 1980’s. Based on his knowledge of mineral liberation models, JvD proposed the application of such models to diagnostic leaching data in 1992. JvD also suggested the concept of attributing some of the leaching to non-liberation phenomena. Leon Lorenzen played a key role in the execution of this research. Whereas JvD proposed in 1995 that neural nets and topological maps should be used to capture the ill-defined relationship between diagnostic leaching data, liberation and size reduction, Chris Aldrich developed the final network architecture, as published in Paper 99.
3. List of Publications Included in this Submission

[ ] Number of citations according to Science Citation Index, excluding self citations.

Process Synthesis

Papers in Refereed Journals


Knowledge Based Systems

Papers in Refereed Journals


Refereed Papers in Conference Proceedings


Non-Refereed Conference Papers


Neural Networks: Theoretical Concepts

Papers in Refereed Journals


Invited Chapters in Textbooks

Refereed Papers in Conference Proceedings


Non-Refereed Conference Papers


Neural Networks: Applications

Papers in Refereed Journals


**Invited Chapters in Textbooks**


**Refereed Papers in Conference Proceedings**


**Non-Refereed Conference Papers**

Computer Vision

Papers in Refereed Journals


Invited Chapters in Textbooks


Refereed Papers in Conference Proceedings


Non-Refereed Conference Papers


Froth Flotation and Interfacial Phenomena

Papers in Refereed Journals


Invited Chapters in Textbooks


Refereed Papers in Conference Proceedings


Non-Refereed Conference Papers


Fluid Mechanics

Papers in Refereed Journals


### Diagnostic Leaching

#### Papers in Refereed Journals


#### Refereed Papers in Conference Proceedings


4. List of Publications Not Included in this Submission

[ ] Number of citations according to Science Citation Index, excluding self citations.

PHD THESES

JOURNAL PAPERS


**INVITED CHAPTERS IN TEXTBOOKS**


**REFEREED PAPERS IN CONFERENCE PROCEEDINGS**

Note: The papers listed here were refereed in full and were published as chapters in books.


22


23


NON-REFEREED CONFERENCE PAPERS

Note: The papers listed here were selected on the basis of an abstract only, or were not refereed thoroughly, and may be published elsewhere as indicated.


117. Petersen, F.W. & Van Deventer, J.S.J., "The mechanism of fouling of adsorbents in CIP and RIP by ore particles and organics", Cyanide in Gold Processing: the next 100 years, S.A. Chemical Institute, 40 p. [Sandton, South Africa, 5-6 November 1990].


120. Van Deventer, J.S.J., Petersen, F.W. & Reuter, M.A., "The interaction between solid fines, organics and metal complexes during the treatment of plant effluent in adsorption columns", ACHENA '91: International Meeting on Chemical Engineering and Biotechnology, DEHEMA. [Frankfurt am Main, Germany, 9-15 June 1991].
121. Van Deventer, J.S.J., "The management training of final year chemical engineering students", ACHEMA '91: International Meeting on Chemical Engineering and Biotechnology, DECHEMA. [Frankfurt am Main, Germany, 9-15 June 1991].


146. Van Deventer, J.S.J. & Aldrich, C., "Application of neural nets in the process industries", ACHEMA '94: International Meeting on Chemical Engineering and Biotechnology, DEHEMA. [Frankfurt am Main, Germany, 5-11 June 1994].

147. Van Deventer, J.S.J., "Obstacles to technology transfer in the mineral processing industries", ACHEMA '94: International Meeting on Chemical Engineering and Biotechnology, DEHEMA. [Frankfurt am Main, Germany, 5-11 June 1994].


28


29


193. Van Deventer, J.S.J. & Rees, K.L., "Improved leaching of gold: Does the literature offer a solution?", *Minerals Engineering '97*. [Crowne Plaza Hotel, Santiago, Chile, 30 July to 1 August 1997].


199. Van Deventer, J.S.J., "Soft sensor development in mineral processing and hydrometallurgy", Invited Panel Discussion on Computational Intelligence, 9th IFAC Symposium on Automation in Mining, Mineral and Metal Processing (*MMM'98*). [Cologne, Germany, 1-3 September 1998]


**OTHER PRESENTATIONS AND REPORTS**

**Unpublished Presentations at Smaller Meetings**


208. "Recent advances in the technology of gold extraction in South Africa", Meeting of the Kalgoorlie Branch of the Australasian Institute of Mining and Metallurgy, 20 June 1988.


216. "The application of connectionist computer vision in mineral and process engineering", Department of Chemical Engineering, The University of Melbourne, Australia, 3 February 1995.

217. "How computer vision and neural networks can assist process engineers", Department of Chemical Engineering, Monash University, Melbourne, Australia, 21 March 1996.


220. "Mineral processing in Australia: Ancient art or modern science?", Professorial Inaugural Lecture, The University of Melbourne, Australia, 15 August 1996.


228. "The interrelationship between mineral processing and environmental management", Department of Civil and Environmental Engineering, The University of Melbourne, Australia, 3 April 1998.


UNPUBLISHED RESEARCH REPORTS


PUBLISHED REPORTS ON CONFERENCES


SUPERVISOR FOR COMPLETED DOCTORAL THESES

University of Stellenbosch, South Africa


Royal Melbourne Institute of Technology

15. Teague, "The behaviour of gold during the flotation of refractory ores", Ph.D., July 1999. (Co-supervised with Dr C. Swaminathan) [Sponsor: Stawell Gold Mines]

SUPERVISOR FOR COMPLETED MASTER'S THESES

University of Stellenbosch, South Africa


22. A.J. Burger, "Flottasie van 'n growwe pirieterts in 'n luggeborrelde hidrosikloon" (English: "Flotation of a coarse pyritic ore in an air-sparged hydrocyclone"), M.Eng., December 1986. (Co-supervised by Prof. F.L.D. Cloete) [Sponsor: Mintek]


24. T.P. Hanekom, "Die gedrag van skuimers tydens die flottasie van piriet" (English: "The behaviour of frothers in the flotation of pyrite"), M.Eng., December 1987. [Sponsor: Sasol]


29. F.W. Petersen, "Inhibited mass transfer to porous adsorbents by organic and fine particles", *M.Eng.*, December 1991. [Sponsor: FRD]

30. I.W. van der Merwe, "A knowledge-based system for the simulation of gold leaching", *M.Eng.*, December 1992. (Co-supervised by Dr M.A. Reuter) [Sponsor: FRD]


34. J.J. Eksteen, "The development and technology transfer of an industrial machine vision system for the control of a platinum flotation plant", *M.Eng.*, March 1995. (Co-supervised by Dr C. Aldrich). [Sponsor: Kenwalt]


SUPERVISOR FOR PRESENT DOCTORAL STUDENTS

University of Stellenbosch, South Africa


The University of Melbourne


44. J.G.S. van Jaarsveld, "Immobilisation of pollutants in geopolymers", started in 1996. (Co-supervised by Prof L. Lorenzen) [Sponsor: ARC]


49. Coster, "The fuzzy dependence of metallurgical efficiency on operating variables in smelters", started in 1998. [Sponsor: ARC]


53. R. Louey, "Tramp element detection in molten steel", started in 1999. (Co-supervised with Dr D.E. Langberg & Dr D.R. Swinbourne) [Sponsor: G.K. Williams C.R.C.]
5. General Statement of Impact

The scientific research community considers impact numbers and citation counts as the main indicators of the quality of published research. The population size of a specific scientific discipline determines the probability that a paper be cited, so that impact numbers for journals and citation counts could hardly be compared across discipline boundaries. It is possible to base the selection of a journal as a publication medium mainly on its impact record. However, this does not mean that the paper will then receive adequate exposure in the target discipline. A further complicating factor is that professionals in a more applied discipline may make extensive use of the principles presented in a specific paper without ever citing that paper owing to the fact that they are practitioners who do not write papers. Therefore, despite the importance of citation counts, such data should be interpreted with caution.

Mineral processing, which is the subject of this submission, constitutes a very small research community world-wide. Even the more general field of chemical engineering is small compared with the basic sciences or biotechnology disciplines. The work in this published collection appeared in specialist mineral processing journals, general chemical engineering journals, as well as a range of conference proceedings and textbooks. In each case the publication was aimed at a selected target audience instead of maximising the citation count. A further aspect is that papers are often cited in theses, conference proceedings and textbooks, but these citations are not included in the Science Citation Index count, which refers only to citations in journals.

| Total Citations According to Science Citation Index, Excluding Self-Citations |
|-----------------------------|--------|
| **Subject of Publications** | **Citations** |
| Process Synthesis           | 25     |
| Knowledge Based Systems     | 16     |
| Neural Networks: Theoretical Concepts | 28     |
| Neural Networks: Applications | 22    |
| Computer Vision             | 56     |
| Froth Flotation and Interfacial Phenomena | 26 |
| Fluid Mechanics             | 3      |
| Diagnostic Leaching         | 7      |
| Total of Publications Included in this Submission | 183 |
| Total of Publications Not Included in this Submission | 230 |
| Total of all Publications   | 413    |
Against this background the citation record relevant to the papers included in this submission as given in the table above is most satisfactory. (The citation count for each paper is given as [ ] in Sections 3 and 4 behind each reference.) For all 101 papers in this submission the total non-self citation count is 183, which means 1.81 citations per paper. When only the 52 published journal papers are considered the corresponding citation count is 137, which means 2.63 citations per journal paper. Considering that the impact number for the top chemical engineering journals is around 1.0 and for mineral processing journals it is substantially lower, this citation record is sufficient evidence of the impact of this work in the research community.

One of the most important contributions of this research is not the number of citations, but the postgraduate students who have participated in the work and who have subsequently transferred their knowledge to industry and other research groups. Section 4 shows that of the total of 15 PhD and 26 Master’s graduates supervised by Jannie van Deventer, the work of 10 PhD and 10 Master’s graduates is covered by this submission. Markus Reuter, Chris Aldrich and Leon Lorenzen have subsequently established an own reputation in mineral processing research. Moreover, from discussions with professionals in industry it is apparent that the numerous presentations at conferences and in industry (as listed in Sections 3 and 4) have enhanced the general acceptance of artificial intelligence and image analysis techniques.

Although Section 6 summarises the significance of the individual publications, a brief statement of the theoretical and industrial impact of the research work is given below:

- The two-stage linear programming method of process synthesis has been the basis for a number of subsequent studies by other researchers.

- The many papers on knowledge based systems and neural networks represent a major attempt to introduce these techniques to the process engineering community.

- The generalised kinetic model using knowledge based systems or neural nets is a simple but very useful and original method for simulating ill-defined dynamic systems. Other authors have already based their work on this concept.
• Three of the PhD graduates of this research programme have formed a software business (Crusader Systems) based on the principles of artificial intelligence developed in this research. They have introduced this software to numerous companies and have also commercialised a computer vision system based on the methods developed in this submission. The substantial industrial impact of this work is not reflected at all in the citation count.

• By employing Kurt Petersen, De Beers has recently decided to further develop the image analysis methods for determining the spray angle of a hydrocyclone and the size distribution of particles on a conveyor belt.

• The research on the effect of galvanic interactions on flotation froths is novel and has been used by a number of mining companies in process optimisation.

• Other authors and a number of research students at other universities have recently based their work on the papers discussing the interrelationship between diagnostic leaching data and mineral liberation of gold ores.
6. Summary of Submitted Publications

In this section a summary is presented of all papers listed in Section 3. The categories of research and the reference numbers of papers are the same as those in Section 3.

6.1 Process Synthesis

Papers 1 to 3 discuss the application of linear programming to the optimal design of mineral processing circuits, which was an extension of the work previously published by J.C.A. Green. The methodology proposed by Green was cumbersome and very difficult to apply to practical mineral processing circuits. In Paper 1 a method was proposed where the optimisation of the circuit was based on two sequential linear models, but the structure of the circuit was based on total flow rates instead of component flow rates. Subsequently, a linear programming methodology was proposed which was a substantial improvement on the technique proposed by Green. An advantage of this method was that the only data required by the simulation model were the separation characteristics, i.e. the split factors in the individual processing units, the tailings grade constraints, and the feed rates. This approach could be used in the design phase of a plant where accurate separation information is not available. Once improved separation data are available during the operation of a plant, it is possible to update the simulations and hence the optimisation of the circuit configuration.

Paper 2 was a substantial extension of this approach by incorporation of regrind mills in the circuit, as well as mineral liberation and a distribution of flotation response in the case of modelling of flotation circuits. The novelty of this method was associated with the fact that milling affects the liberation distribution of species and that this could affect the optimisation of the overall mineral processing plant structure. This paper catalysed a substantial number of further process synthesis studies in the mineral processing literature. Usually the simulation of mineral processing circuits incorporating regrind mills is extremely complex. In this paper the action of a regrind mill was incorporated in the bounds on the separation factors, which meant that a change in the degree of mineral liberation or the floatability of a component due to changes in the surface chemistry could be incorporated in the mathematics in a simple manner.
The shortcoming of Papers 1 and 2 was that the separation factors had to be specified, and where changes in the circuit operating conditions were encountered, subsequent changes in the split factors were not considered implicitly. Therefore, no methodology was proposed to change these split factors during the optimisation. Many of the subsequent papers in which Markus Reuter participated considered the use of knowledge based systems, and subsequently, neural networks to relate split factors in processing plants to operating conditions. These methods provided a way in which an updating of split factors could be accomplished during a sequential linear programming approach, as explained in Papers 1 and 2. An example of this methodology was given in Paper 3, where gravity separation was used as an example. Similar to Papers 1 and 2, a sequential linear programming model was used, where the first model optimised the circuit structure based on total flow rates, while the second linear programming model optimised the flow of individual elements and hence the total grade. A knowledge base was used to generate the bounds on separation factors, and in this way, the separation factors could be updated by the knowledge based system. It is also possible to use a neural network to simulate the split factors for processing units in a circuit and hence conduct process synthesis for a non-linear system such as a hydrometallurgical zinc recovery plant (Papers 33 and 37). Such a process simulator could be used to determine the environmental and economic impact of process decisions. This work has recently been extended substantially by Markus Reuter. A further example of the use of neural nets in process synthesis is the decomposition of circuits containing recycle flow streams, as explained in Paper 39.

Paper 4 discussed the application of standard dynamic modelling methodology to the re-pulp section of a leaching plant where a reaction occurred in the re-pulping tanks. In that plant, milling of the material occurred on a stop-start basis so that steady-state was never achieved. Furthermore, virtually no on-line instrumentation was available owing to the harsh conditions on the plant. The proposed approach made use of a dynamic simulation model which prompted the user for additional information, such as valve positions, the time of milling, and an estimation of flow rates. This program provided the operators with an off-line control technique to optimise and stabilise pulp density and hence the consistency of flow into the ultimate leaching autoclaves. Reaction was taking place in the re-pulping tanks, so that an estimation of the reaction kinetics had to be taken into account. This study has wide implications in the sense that many plants in the metallurgical industry are controlled by the intuition of the operators, owing to a lack of suitable on-line instrumentation. This means that
operators often have to guess the level of fluid in tanks as well as the flow rates to reactors. To a large extent process operators are expected to solve a dynamic material balance intuitively without having the assistance of an off-line model available. Further work is currently being conducted to apply this methodology to the high temperature industry.

6.2 Knowledge Based Systems

Papers 5 to 14 discussed the development of knowledge-based systems for mineral processing, hydrometallurgical and pyrometallurgical systems. This work was conducted in the early days of the application of knowledge-based systems to mineral processing problems, so that standard packages were not used. Instead, it was attempted to develop a new methodology in which the intrinsic behaviour of systems could be incorporated within the structure of the knowledge base. At the end of the 1980’s when this work was conducted, an object or Windows based approach was not available in standard software and the few knowledge-based structures that existed were seriously lacking in their ability to incorporate generic process knowledge. Furthermore, the data base structures that are widely available currently were also lacking. This provided the justification for the development of a structure as proposed in these papers. To some extent this work was part of the pioneering effort in getting knowledge-based systems accepted within the metallurgical industry. While substantial software development was associated with the papers listed here, it was soon overtaken by commercial development of knowledge-based systems and databases.

The novelty of the approach presented in this collection of papers lies in the way in which a pivotal condition is selected and how the knowledge base is used to determine adjustment factors in order to change the pivot values. In this way, meta-classes, classes and sub-classes of objects were defined in order to organise the knowledge space. Interaction factors were also calculated within the data base in order to account for the multi-dimensional nature of the data space. A methodology was proposed for the inverse application of the knowledge based system for the purpose of fault diagnosis. A distinction was made between shallow and deep process knowledge. Shallow knowledge was defined in terms of qualitative or heuristic information expressed as objects, labels and rules. These objects were related to adjustment factors via if-then rules. At the time when this work was conducted little attention was paid to deep process knowledge. The overall structure of the knowledge base and the use of an object
oriented approach were not novel, because a similar methodology had been proposed in the chemical engineering literature. However, artificial intelligence and knowledge engineering received very little attention in mineral processing at the time. Moreover, no clear methodology existed to incorporate deep knowledge such as fundamental equations and models, correlations or differential equations in an object oriented knowledge based approach.

In order to overcome the shortcoming of incorporating kinetic data and dynamic behaviour in the knowledge based models a novel methodology was proposed in papers 5 to 14. In essence the kinetics of a reaction were finger-printed by considering the slope of a batch kinetic curve at any position in terms of process conditions. This meant that as process conditions changed during a batch experiment, the splines were described via the objects and adjustment factors of the knowledge based system. Therefore, the usual incorporation of process conditions into the rate constant was replaced by the changes in slope. In this way any complex reaction kinetic system could be simulated without taking the reaction mechanisms into account. This method enabled the data to be the complete data base instead of a curve fitted kinetic model.

By applying this method to a wide range of processes it was found empirically that simple proportional adjustment factors could be used to adjust kinetic curves for different initial concentrations, solid/liquid ratios and process equilibria. This unique method provided a useful strategy of linking deep process knowledge and systems equations with shallow process data. For example, reaction rates used in continuous dynamic material or heat balances were related directly to batch data via the knowledge base. This also meant that the identification of reaction kinetics could be conducted by back-calculation from continuous process data.

Unfortunately this methodology was published only in the mineral processing literature, which has a more limited exposure than the more general chemical engineering literature. Consequently, this methodology has not been actively pursued by other authors. Unfortunately, few other authors have adapted deep knowledge based systems for application to dynamic data sets. Papers 5 to 14 used mainly published data sets to facilitate comparison of model fitting. Papers 5, 6, 9, 10, 11 and 13 explain how this approach was applied to batch leaching, continuous co-current and counter-current leaching of gold ores, as well as adsorption of gold cyanide onto activated carbon in batch reactors, periodically operated counter-current cascades, columns and carbon-in-leach systems. Different process conditions such as cyanide concentrations and different degrees of adsorbent fouling could be taken into
account using adjustment factors through the knowledge base. This approach was computationally more efficient than utilising the more rigorous adsorption and leaching models published by van Deventer and others. Papers 7, 8, 12 and 14 integrate a knowledge based approach with the two stage linear programming methodology proposed in Paper 2 in defining bounds on split factors for flotation circuits. This approach allowed the steady-state simulation and optimisation of circuit configuration by taking reagent concentrations, reagent quality, changes in ore mineralogy, and hence changes in batch flotation kinetics, into account. As explained in Paper 6, the knowledge based kinetic methodology was also applied successfully to the oxidative leaching of sphalerite, the leaching of chalcopyrite at elevated temperature, the leaching of cobalt concentrate, the carbothermic reduction of ilmenite, the decarburisation of liquid iron by carbon dioxide, and the reduction of nickel oxide by hydrogen.

6.3 Neural Networks: Theoretical Concepts

It is not entirely possible to separate the neural net papers dealing with fundamental concepts from those discussing engineering applications. Papers 15 to 29 deal mainly with new theoretical concepts concerning the adaptation of neural network methodology to processing problems. While neural networks are currently used more widely in process engineering, these papers presented some of the first attempts to introduce neural networks to the field of mineral processing. That was indeed the purpose of Papers 17, 24, 25, 26 and 27.

It was a logical step to combine the knowledge based kinetic methodology proposed in Papers 5 to 14 with various neural network architectures. Paper 15 used the same methodology explained in Papers 6 and 7, but a neural net instead of a knowledge base was used to relate the adjustment factors, concentration values and rate data to process conditions. At the time all papers in the process engineering literature dealing with the application of neural nets to reaction kinetics incorporated time as an input. Therefore, intrinsic rate information was not captured. The approach explained in Paper 15 applied a first order kinetic expression to rate data, where the rate "constant" could be a complex function of concentration and process conditions. A pivot set of concentration and rate "constant" data was defined as being independent of process conditions, i.e. the kinetic mechanism did not change. A neural net was then used to relate changes in process conditions such as temperature, density, reagent...
levels, pH and stirring conditions to the adjustment factors. No curve fitting was required, as the kinetic rate equation utilised experimental data directly. The neural net approach of capturing process information allowed the identification of process conditions and hence fault diagnosis if the adjustment factors were known. A procedure was also proposed for incorporating the neural net kinetic equation into the material balance equations for mixed flow reactors, both in the case of macromixing and micromixing. It was shown that this novel approach could be applied to industrial hydrometallurgical and flotation data sets. Due to the useful features of the method proposed in Paper 15 it has been used as a basis for further work by other workers in the field. Nevertheless, most papers that appear on the application of neural nets to kinetic problems still consider rate data in a black box manner without attempting to capture intrinsic rate information.

In industrial plants it is not always appropriate to link batch data to continuous operations owing to differences in operating conditions. Therefore, intrinsic rate information should be extracted from continuous data, as proposed in Papers 16, 24, 28 and 29. The implicit assumption was made that conservation equations could be decoupled and in the case of a simplified dynamic material balance a quantity representing the intrinsic reaction rate could be back-calculated by assuming mixed flow behaviour. Despite the fact that these assumptions were not necessarily valid, the calculated rate information incorporated all the associated non-idealities. The back-calculated rate data, coupled with the associated process conditions, could then serve as training data for a neural net. This methodology was applied successfully to countercurrent adsorption systems involving complex intrinsic reaction kinetics.

The scaling and transformation of input data to reflect trends inherent in the data were also demonstrated in Paper 16. This enhanced the efficiency of training and simulation of the neural net. It was also shown that it is not always efficient to train a neural net over the entire variable space. Instead, a methodology of partitioning the variable space into sub-spaces was proposed, as also explained in Paper 32. A simple perturbation analysis was applied, which used a neural net to estimate first order partial derivatives of a multivariate function. These partial derivatives then quantified the relative effects of the variables at different positions in the variable space, so that sub-spaces of reduced dimensionality could be identified by eliminating less significant variables from certain sub-spaces. As a further simplification, neural nets were then used to identify simple mathematical relations between the rate
information and variables. At the time this approach was essential in order to reduce computational demands when applying neural nets to highly non-linear data spaces. However, with powerful computers and sophisticated neural net architectures commercially available today there is less of a demand for this sub-space approach. Nevertheless, there are still numerous problems in reaction engineering systems where the variable space is less continuous and smooth, so that a partitioning approach may still be preferable.

Papers 18, 19, 20, 22, 23 and 26 proposed a novel method for the detection and location of gross (systematic) errors as well as the reconciliation of inconsistent process data by using neural networks. Paper 23 presented a comprehensive summary of this work and explained the methodology involved. Numerous methods have been published on material balance smoothing and gross error detection. However, most of these methods are based on the assumption that measurement errors are randomly Gaussian with a known covariance matrix and distributed around a zero mean. Likewise, most methods for gross error detection involve statistical tests based on the characteristics of the constrained residuals of the measurement errors. These tests are only useful when the system constraints are linear or could be linearised, which is usually not the case in mineral processing. When a neural net was used to capture the relationship between the properties of the constraint residuals and those of the measurement residuals, this problem of assumptions and non-linearity was overcome. This ill-defined neural net relationship then served as the basis for the estimation of measurement error variances or gross error detection schemes.

In Papers 18 and 19 conventional back-propagation nets with sigmoidal or preferably hyperbolic tangent transfer functions were used to detect gross errors in material balance measurements in hydrocyclone and grinding circuits. In Paper 20 data reconciliation was conducted using a conventional crossbar or Hopfield net with full lateral connections, a modified Hopfield net with a multi-dimensional array of hidden layers, and the latter structure with feedback connections and a direct search method with a systematic reduction in the search space. Paper 22 presented a comparison of network architectures during the detection and location of gross errors in a number of synthetic mineral processing problems. Back propagation, probabilistic, learning vector quantization (LVQ), radial basis and general regression neural nets were found to all detect gross errors very well. However, the more complicated problem of locating the sources of gross errors in a process system was handled best by LVQ nets. Probabilistic nets performed almost as well, but back-propagation nets
were not suitable for this purpose. The reason was that the neural nets had to interpret complex patterns of constraint residuals that were dependent not only on the location of systematic errors, but also on the actual values of the process variables at the time of sampling. There are still problems with the on-line training of neural nets under industrial conditions, so that these methods have not yet been used industrially.

Despite the methods proposed in Papers 15 and 16, few neural net architectures inherently include the functional relationships between variables based on fundamental knowledge. For example, if it is known that two variables have a multiplicative, additive or exponential relationship it makes sense to include this as part of the network structure. This will also allow some differentiation between parametric and non-parametric knowledge. Standard architectures are not suitable for this purpose, which provided the justification for the regression network proposed in Paper 21. This represented a unique approach with substantial potential to link fundamental deep knowledge with shallow heuristic knowledge. It also meant that the regression network had the ability to extrapolate beyond the scope of the training data, which is usually a weakness of neural nets. Paper 21 showed that the regression network far outperformed back-propagation structures and multiple adaptive regression splines. Unfortunately, the training of this type of network was difficult owing to the presence of plateaus adjacent to valleys on the error surface. Paper 21 proposed various optimisation methods to overcome these difficulties in order to train the network optimally. With enhanced computational power available today there is a greater justification for use of the regression network concept than was the case in the early 1990's.

6.4 Neural Networks: Applications

Papers 30 to 44 demonstrated that neural networks could be applied to a wide range of process engineering problems. The objective of these papers was also to enhance the acceptance of neural networks in the wider process engineering community. Although these papers discussed optimal network architectures for specific problems, the aim was not to develop new generic knowledge in artificial intelligence.

Neural nets have only recently found application in the pyrometallurgical community. Paper 30 was one of the first publications addressing this topic and hence has been cited by several
authors. It was shown that neural nets could relate slag properties such as viscosity and metal-slag equilibria to composition, whereas the fundamental and empirical models available in the early 1990's were not able to do so. Today sophisticated slag models are available which have better predictive performance, but they still fail for very complex systems. Provided that sufficient experimental data are available, neural nets could be used to correlate complex equilibrium relationships not amenable to conventional modelling. Paper 37 showed how such trained neural nets could then be used to interpret metallurgical processes. This approach was also used to correlate equilibria in solvent extraction systems (Paper 42) and multicomponent adsorption systems. With more sophisticated neural network architectures becoming available it is recommended that aspects of fundamental equilibrium models should be combined with non-parametric structures, as proposed in Paper 21.

Although most neural networks are not transparent in terms of the structural relationship between variables, Paper 31 demonstrated that backward analysis could be used to analyse the weight matrix of a three layer perceptron. It was shown that the viscosities of crude oils could be correlated with their measured properties at different temperatures. In view of the non-linearity of the system the neural net gave better correlations than any of the published empirical equations. Likewise, Paper 32 showed that ordinary sigmoidal back-propagation neural nets gave better prediction of the classification efficiency of a hydrocyclone than multiple adaptive regression splines (MARS) methods. The shear stress versus shear strain relationship for superplastic materials such as a copper alloy was simulated accurately using a hybrid neural network incorporating an Arrhenius relationship for the effect of temperature. Paper 44 showed that it was important to incorporate such fundamental knowledge, so that hybrid neural nets had better extrapolative ability than conventional architectures. As mentioned before, it was shown that the pre-processing and scaling of data were important. Paper 33 also emphasised this point. For inadequate or sparse data sets a hybrid sub-space approach was recommended.

Papers 33, 36, 37, 39, 40 and 44 explained how the inputs and outputs for an entire processing plant could be related via neural nets in order to construct a plant-wide connectionist simulator. Depending on the geometry of the data space it was important to design an optimal network architecture and to select the best transfer function. Often a hyperbolic transfer function was preferred to a sigmoidal form so that low valued and high valued outputs could be treated equally. In view of the non-linear nature of most plant-wide models, neural
networks yielded a better representation of reality than multi-linear regression models. The following case studies were considered in terms of operating variables: (a) the losses on a gold plant; (b) the consumption of cyanide on a gold plant; (c) the extraction efficiency of a uranium plant; (d) the consumption of reagents on a phosphate flotation plant; (e) the pyrometallurgical recovery of base metals from industrial flue dusts; and (f) the optimisation of a rotary salt flux furnace for the processing of secondary aluminium.

The methodology of representing the intrinsic kinetics of a complex reaction system using a neural net, as explained in Paper 15, was applied in Papers 34, 37, 38 and 44. As demonstrated for superplastic behaviour in Paper 44, it was shown in Papers 34, 38 and 44 that a kinetic neural net had better extrapolative abilities if the highly non-linear effect of temperature was excluded from the net and instead incorporated as an Arrhenius relationship in a hybrid architecture. This confirms the contention that the inputs into a neural net structure should be minimised and that a combination of parametric and non-parametric information structures has superior predictive capabilities compared with a total black-box approach.

A large variety of neural net architectures were used and applied to selected case studies which revealed distinctive geometries of variable space. In Paper 35 radial basis transfer functions were used for the hidden nodes, while the pattern layer representation was constructed solely by self-organisation, with the output layer being trained using standard techniques. This architecture was used to correlate the onset and rate of induced aeration with rheological and design variables. It was shown that conventional back-propagation nets did not deliver a satisfactory result. The concept of a cascaded net structure was also proposed in which a stage-wise prediction of variables was facilitated. A hybrid neural net with a self-organising Kohonen layer was used in Paper 42 to correlate the equilibrium of rare-earth solvent extraction with process conditions. An interesting hybrid structure was developed in Paper 42 to relate a dimensionless number for drop coalescence in liquid/liquid dispersions to operating and solvent parameters for distinctly different classes of solvents. As shown in Papers 43 and 99, a sigmoidal back-propagation neural net with a Kohonen hidden layer was used to relate the liberation of gold from refractory ore components to diagnostic leaching data.

It is often difficult to derive any conclusions from a statistical analysis of an industrial data set in which the relationship between variables is ill-defined. Such a case study is the pressure...
leaching of copper-nickel matte described in Papers 36 and 41. Rather than performing a quantitative prediction of the performance of the process, the outputs were classified in a fuzzy manner as either high or low in order to simplify the data set. Learning vector quantization (LVQ) and back-propagation nets were used to extract knowledge from the historical process data and thereby determining the relative importance of process variables.

In this work it became apparent that the determination of residence time and hence the time delay between processing units from sparse data sets were problematic. Although this problem was again addressed in Paper 56, a generalised methodology has been developed since in the Ph.D. work by Bezuidenhout.

6.5 Computer Vision

The aim of the work on knowledge based systems and neural networks was to utilise ill-defined information for predictive purposes. Therefore, it was a natural extension of this work to develop computer vision methods for mineral processing operations and to link such information via neural networks to process performance. At the time when this work was initiated in the early 1990’s the few available studies were either very computationally intensive (Woodburn’s early work on coal froths) or very simplistic by considering only average grey level. The numerous studies on computer vision in the electronic engineering literature had focused mainly on artificial and well defined examples at that stage. Papers 45 to 70 represent a major effort to develop and transfer computer vision technology to mineral processing. There is necessarily an overlap with the previous sections on neural network applications, because this work represented the first comprehensive attempt to integrate computer vision and neural network principles in a mineral processing application.

Paper 58 presented a comprehensive overview of computer vision and how neural networks could be applied either directly to image data or indirectly to features extracted from such data. When neural nets are used for image compression at the level of raw pixel data a significant reduction in storage space and an increase in the level of computation are attained (Paper 47). Morphological edge filtering was also shown to be computationally more efficient when using neural networks. The analogy of machine vision systems with biological systems was also explained in terms of the processing time, the size of the feature space and the level of information (Papers 46 and 58).
An important extension of this work was the introduction of textural image feature extraction techniques. This was the first time that spatial and neighbouring grey level dependence matrix methods were applied to flotation froth images, as explained in many of the publications, including Papers 49 to 52. Although the analytical use of the SGLDM and NGLDM features was limited owing to the inherent difficulties in the interpretation of feature sets of high dimension, they effectively captured knowledge of the process reflected in the structure of the flotation froth. The ratio of the Energy features of the SGLDM method calculated at 90 degree angles was shown to be indicative of the average bubble shape and could identify elliptical bubbles. A thresholding method for calculating froth mobility and stability parameters was also proposed. Paper 50, which has been cited frequently, showed that the second moment of the NGLDM method correlated well with the flow speed of the froth. Depending on the froth type, one or more textural features could be related directly to the coarseness of the froth image and hence the average bubble size.

In many of the papers neural nets were used to relate the image features of the froth to flotation performance. Initially, it was attempted to classify the froth in categories which corresponded to those descriptions used by the operators. Subsequently, the image features based on textural analysis were used for this classification. For example, Papers 49 and 59 demonstrated that textural froth features could be combined via an LVQ neural net to classify
froths. It was also shown that without froth classification the textural image features could be related to process conditions and metallurgical performance via a simple neural net in the case of batch flotation (Paper 50).

In a series of publications (Papers 48, 51, 52, 53, 56, 59, 60, 64, 65, 66, 68, 70) froth image features were projected onto a two-dimensional mapping to obtain a visualisation of the differences between froth classes in space. It was shown that pre-allocated classes of froth occupy reasonably different positions in such a mapping. The degree of dispersion on such a map and consequently the separation between froth types were quantified (Papers 51, 52 and 59). Initially simple Kohonen self organising maps (SOM) were used. However, for applications where it was not possible to train the net on all possible process conditions prior to implementation, fuzzy adaptive resonance theory maps (ARTMAP) were used as explained in Paper 48. In Papers 53 and 60 Sammon maps were used due to their more faithful representation of the structure of the original data space. However, as explained in Paper 60, conventional Sammon maps have the shortcoming that when new data are evaluated, the entire map has to be reconstructed. Hidden target mapping neural nets were used in Paper 60 to overcome this shortcoming. In many of these papers it was shown how a two or three-dimensional mapping could be used to track changes in process conditions. In that way operators could be provided with a tool which helps to visualise complex multivariate changes and relate those to metallurgical performance. It was shown in Papers 53 and 60 that the position on a map was related not only to the type of froth and the set of operating conditions, but also the recovery and grade of the froth.

During application of a computer vision system at mines in South Africa and Australia it was found that perturbations in processing units such as comminution preceding the flotation circuit were reflected in perturbations in the froth image features. Papers 56, 65, 66, 67, 68 and 70 showed that long residence times between the points of perturbations, the position of the froth monitor and the points of on-stream analysis did not always allow a direct correlation between froth features and metallurgical performance as measured by the taking of lumped samples. However, it was possible to relate short term fluctuations in image features to short term fluctuations in metallurgical performance. Paper 56 proposed a methodology whereby such short term perturbations and long term shifts in the base line could be distinguished on an empirical basis. Recent unpublished work has significantly advanced this methodology by devising a novel filter which eliminates shifts in the base line of data sets. As
illustrated in Paper 56, SOM plots of the sign of perturbations presented a meaningful classification of froth types.

None of the papers mentioned above discussed the closing of a control loop during the monitoring of industrial flotation froths. This is a daunting task in view of the complexity of flotation systems. However, further industrial work by Moolman has succeeded in achieving this goal through the integration of a sophisticated software platform with the computer vision system. That system marketed by Crusader Systems in South Africa is now being used commercially. In that respect the above work has made an impact on industrial practice in addition to the contribution to the body of knowledge.

In an extension of the work on froth monitoring, Papers 55 and 62 proposed a method for quantifying the underflow spray angle of a hydrocyclone as an indicator of changes in cut size. In further work it was shown that the spray angle in combination with information such as inlet pressure, flow rate and pulp density could be used in a neural network as a soft sensor for prediction of particle size distribution. Even today very little is known about the relationship between spray angle profiles and the fluid behaviour in a hydrocyclone, so that this work presented a pioneering effort in the field. Although a few papers have been published on the image analysis of material on conveyor belts this is a very complex problem that has not been resolved satisfactorily. Paper 57 presented a novel textural pattern method for recognition of types of ore and projected particle size distribution on a conveyor belt using range and variance operators. Although more sophisticated methods have been published, this textural method offered a simple and computationally efficient method which could be incorporated in a commercial system. It was explained how changes in the textural features could be related to changes in operating conditions.

6.6 Froth Flotation and Interfacial Phenomena

Papers 71, 76, 77, 78, 79, 80, 81, 82, 88 and 89 discussed chemical aspects of froth flotation with an emphasis on electrochemistry, galvanic interactions, reagent-mineral interactions and mineralogy. Engineering aspects of flotation, such as the flow behaviour in an air-sparged hydrocyclone, mass transport in conventional and column froths, and the development of a moving froth bed system, were discussed in Papers 73, 74, 83, 84, 85, 86, 87, 90 and 91.
limited effort on solid-liquid separation by electrokinetic means was presented in Papers 72 and 75.

Despite the wealth of information on flotation chemistry available in the literature, Paper 71 made a valuable contribution by showing that rest potential and Eh measurements should be interpreted with care, especially when pulp and solution values are compared. This paper was one of the few that used rest potential values to draw conclusions about xanthate speciation at the mineral surface. It also explained the relationship between rest potentials and pulp potential. This work laid the foundation for Papers 76, 77, 80 and 82 in which it was demonstrated how changes in the milling conditions affect galvanic interactions and subsequently froth behaviour. In contrast with other papers which related galvanic interactions only to pulp phenomena, this was the first time that froth behaviour was related to galvanic interactions. It had been argued conventionally that a reduction in potential caused by the addition of metallic iron should reduce the floatability of sulphides. On the other hand, it had been well known that an overload of hydrophobic gangue should reduce froth stability. These papers showed that the addition of iron reduced the recovery of undesirable hydrophobic particles and hence yielded an improved froth with higher grades and recovery. Paper 80 showed that these phenomena had an interesting effect on the behaviour of free gold during flotation. Paper 89 presented a study in which uranium containing minerals were selectively destroyed in order to investigate their relative floatabilities with respect to uranium recovery. The methodology in this study was analogous to Papers 78, 79, 80 and 81 which considered gold flotation. It was found in Paper 89 that small additions of iron promoted the recovery of pyrite and hence uranium, but higher additions of iron had a detrimental effect. Even when oleic acid was used to enhance the flotation of oxides small additions of iron had a positive effect. This effect could not be explained in terms of electrochemical theory as in the case of sulphides. It is evident that the above papers provide a basis for future work in this field.

Many papers have appeared on the flotation of gold ores, but no other studies have distinguished analytically between free and lock gold in this context. Papers 78, 79, 80 and 81 presented recent results which showed that a large percentage of free gold was recovered by entrainment. Therefore, the optimal grade/recovery combination for gold was shown to be a compromise between factors promoting entrainment and factors promoting the hydrophobicity of sulphide minerals. The flotation of free gold was affected largely by
physical constraints such as the shape and size of particles, the degree of water and gangue transport to the froth, the froth stability and the extent of bubble loading of sulphides which could provide a barrier towards the hydrophobic bubble attachment of free gold. Quantitative XRD was combined with diagnostic leaching to distinguish between free and locked gold in the individual minerals. The specific ore studied contained a large percentage of gold associated with pyrrhotite instead of pyrite, which is unusual. Consequently, a compromise had to be reached between oxidising conditions promoting pyrite recovery and reducing conditions promoting pyrrhotite recovery. Paper 81 was one of few studies which investigated the sequence of copper addition in relation to xanthate conditioning. When copper sulphate was added after the collector the recovery of free gold increased significantly with a concomitant increase in the total gold recovery.

Although the concept of the air-sparged hydrocyclone was developed by Jan Miller, there was a lack of data on its applicability to specific ores and the associated reagent consumption. Papers 73 and 83 compared the recovery of pyrite in the air-sparged hydrocyclone with that in a batch cell and concluded that satisfactory results were obtained for a rougher operation, but that reagent consumption was three times that of conventional flotation. Pilot scale experiments at the University of Stellenbosch also showed that important improvements could be made to the original design. Papers 73, 74 and 91 demonstrated that an underflow spigot in which the flow was restricted by an orifice was more successful than an annular opening in preventing blockage by wood chips in the slurry. In an interesting modification of the original design non-porous sections were inserted at various positions along the length of the cylinder. Papers 74 and 91 showed that improved performance was obtained when the lower section of the porous cylinder was sealed off for certain ranges of operating conditions.

Papers 84, 85, 86 and 87 presented different features of a mathematical model to simulate the behaviour of flotation froths on the basis of mass transport phenomena by taking account of the rates of flotation, entrainment and drainage of various solid species and water. This methodology could be applied to mechanical or column flotation cells in order to simulate the effect of changes in design parameters on flotation efficiency. For example, Paper 87 explained that smaller mechanical cells should be used or additional concentrate launders should be provided for enhanced recovery of coarse particles. A two dimensional regionalised streamline model for the froth was used to relate flow patterns to transport phenomena. The model was calibrated by fitting concentration profiles to measured values. Test work on the
Rougher, cleaner and recleaner cells of pyrite and fluorspar flotation plants demonstrated the general validity of the method. Paper 86 showed that the extent of detachment of floating species could be inferred from the concentration profiles, which allowed the calculation (Paper 84) of the optimal froth depth. The approach presented for flotation columns in Paper 85 served as the basis for a number of studies by other authors. Current work at the University of Melbourne is aimed at the integration of these mass transport phenomena in the froth phase with froth image features determined by computer vision.

Coarse hydrophobic particles are poorly recovered in conventional flotation. As an extension of an invention by Victor Ross, Paper 90 proposed a novel moving froth bed system in which the ore particles were fed from the top onto the froth bed. The hydrophobic particles acted as film breakers and were recovered as a sink product, while the hydrophilic particles were supported by the froth bed and were recovered as overflow product. As expected, the mass/projected area ratio was an important determinant of the recovery efficiency of particles. More papers will in due course appear on this work.

As a one off project, ultrafine kimberlite slurries which could not be dewatered by conventional means were subjected to electrokinetic dewatering techniques. Paper 72 presented the results of both electrophoretic and electro-osmotic experiments, with the latter being more energy efficient than the former. At the time very little information was available on the industrial application and efficiencies of these methods. Paper 75 presented an extension of this work in which a temperature dependent mathematical model was developed for electro-osmosis and applied to two kimberlite samples and a coal slurry. It was shown that a gap existed in the literature in terms of temperature-conductivity relationships for slurries.

6.7 Fluid Mechanics

This section is a collection of a number of ad hoc studies on fluid flow phenomena which supported the major thrust of the research discussed in the rest of the papers. For example, the segregation of particles of different size, density and composition in a flow-splitting device is commonplace in mineral processing applications. Paper 92 presented the first detailed quantification of this phenomenon and explained why a Y-piece is preferred to a T-piece when splitting a slurry.
Induced aeration is used frequently in flotation cells and bioleaching reactors. In the mid 1980's little information was available to predict the onset of such aeration and the rate of aeration in terms of design and rheological variables. Papers 35, 93 and 94 presented results and correlations to fill this gap in the body of knowledge. Paper 93 showed that at low viscosities an increase in viscosity resulted in an increase in aeration rate, while the opposite was true for highly viscous fluids. Paper 94 explained the effects of fluid density and the presence of suspended solids on aeration rate.

In many mineral processing operations there is justification for process intensification. The work on the air-sparged hydrocyclone was such an example. In Paper 95 a high velocity impinging stream cross flow reactor was proposed for the enhancement of leaching of gold ores. Although not proven conclusively, it was suggested that micro-cracking on the surface of particles caused by the impinging stream of leaching solution could increase gold dissolution from refractory ores. Similarly, Paper 96 explained how the configuration of liquid and gas nozzles in a compact reactor could affect the efficiency of mass transfer. This work on high intensity jet reactors was a new contribution to the field, as little has been published in this regard. As expected, the angle of impingement between the gas and liquid streams was the most important aspect to affect the efficiency of mass transfer. A mechanism of bubble formation and break-up in a reactor operated in the bubble mode was proposed and a design methodology for such reactors was also formulated.

6.8 Diagnostic Leaching

Diagnostic leaching is a methodology which bridges the gap between mineral processing and hydrometallurgy. It applies selective destruction of minerals by leaching to determine the amount of gold associated with each mineral group. This information could then be used to evaluate the efficiency of mineral processing operations such as flotation. Papers 97 to 101 do not claim to have invented the technique, which was pioneered by Leon Lorenzen and co-workers. However, Papers 97 and 100 demonstrated the industrial application of diagnostic leaching and discussed factors affecting the selectivity of mineral destruction. At the time the purpose of these papers was to transfer the technology to the wider mineral processing community.
Papers 98 and 101 presented a pioneering effort to relate mineral liberation to diagnostic leaching results. The King liberation model was used as a simple representation of gold liberation with changes in particle size. Consequently, the grain size of gold was back-calculated from leaching data as a mineralogical parameter. Even with little size reduction some gold leaching was measured, so that it was postulated that a fraction of leachable gold could not be attributed directly to liberation. This simple first attempt has since been refined and has served as the basis for a number of further liberation models published by other authors. In a further development Paper 99 explained how neural nets could be used to relate diagnostic leaching data to size reduction information. In contrast with Paper 98 where a specific functional form was assumed for the liberation model, the non-parametric nature of the neural net in Paper 99 eliminated the need for a particular functional relationship. A Kohonen feature map was constructed to project the multi-dimensional gold liberation problem onto a two-dimensional space in order to quantify the differences in liberation behaviour between different ore types. A measure of clustering and the centres of gravity for the positions of unmilled and milled ores on the topological maps were used in this characterisation of ores. In a novel neural net architecture the unmilled ore characteristics as determined partially by diagnostic leaching were summarised in a Kohonen layer and then related to the liberation data via a back-propagation structure (Paper 43). Further work is being conducted at the University of Melbourne to relate mineral liberation and diagnostic leaching data for refractory and preg-robbing gold ores.
Section 7  Publications on

Process Synthesis
OPTIMAL DESIGN OF MINERAL SEPARATION CIRCUITS 
BY USE OF LINEAR PROGRAMMING

M. A. REUTER and J. S. J. VAN DEVENTER
Department of Metallurgical Engineering, University of Stellenbosch, Stellenbosch 7600, South Africa

J. C. A. GREEN†
Department of Chemical Engineering, University of Stellenbosch, Stellenbosch 7600, South Africa

and

M. SINCLAIR
Department of Applied Mathematics, University of Stellenbosch, Stellenbosch 7600, South Africa

(First received 13 May 1987; accepted in revised form 11 September 1987)

Abstract—A model is proposed for the simulation of mineral separation plants using linear programming. A mineral separation plant is described here by two linear models, one being a subset of the other. These models are solved in succession, the first producing the total flow rates between the banks and the circuit configuration, and the second the flow rates of mineral species. Circuit configurations produced by the simulation model are similar to those encountered in industry. This suggests the feasibility of applying the model in the design of separation plants in practice. The grades and recoveries obtained also closely resemble those of similar separation plants in industry. The only data required by the simulation models are the separation characteristics, tailings grade constraints and the feed rates.

INTRODUCTION

Mineral separation processes often consist of a number of similar units which are grouped into banks and interconnected in a predefined manner. In some of these processes no transformation takes place in the particle size distributions. This means that the feed is simply split into different fractions on the basis of one or more properties such as density, magnetic susceptibility, conductivity, surface properties and fluorescence. For example, a gravity concentration plant may comprise a number of cones or spirals which are grouped into rougher, scavenger and cleaner banks, in order to split a feed into concentrate, middlings and tailings streams.

Many papers have been published on the simulation of flotation plants, which separate minerals on the basis of hydrophobicity. However, very few papers deal with the design and simulation of other mineral separation circuits such as gravity concentrators (Burt, 1984).

When designing mineral separation plants, it is attempted to produce an optimal configuration as well as optimal operating conditions. In this paper a general optimisation routine is proposed that may be used to simulate any mineral separation plant comprising a number of banks.

Most flotation plant simulators and optimisation routines are based on numerical methods. Mehrotra and Kapur (1974) optimised the mean residence time of mineral species and obtained an optimal structure of flotation cells from a generalized circuit by direct search methods. The circuit structures thus produced do not correspond to circuit configurations generally encountered on an industrial plant. King (1976) developed the NIM simulator, which consisted of mass balance equations defined for mineral classes in a fixed structure. The system of equations, which include a flotation model, was solved iteratively, producing mineral class flow rates and flotation bank retention times. Sutherland (1977) applied a pacer type optimisation to simulate flotation circuits which could include regrind mills in their structure.

Williams and Meloy (1983) and Williams et al. (1986) employed a technique introduced by Meloy (1983) called “circuit analysis” to simulate mineral separation processes with or without multiple feeds. In this approach it was assumed that individual unit operations are linear. A selectivity function defined the split of the feed into concentrate and tailings. The ensuing equations were linear algebraic equations that could be solved without the use of any numerical methods, and could be applied to countercurrent, cocurrent and multiple-feed processes.

Green et al. (1985) developed mass balance equations that defined the flow rates of mineral species between the flotation banks in terms of split factors, as suggested by Nagiev (1957). The mass balance equations of a fixed structure produced a coefficient matrix which yielded bounds on the flow rates via Ostrowski’s theorem. Subsequently, the flow rates of the mineral

†Present address: Department of Quantitative Management, University of South Africa, Pretoria, South Africa.
species could be computed using parametric linear programming. As an extension to this work, Green (1984) linearised the mass balance equations by considering the valuable mineral species Cu only. The resulting linear model could be solved using linear programming, and the separation characteristics of the Cu in each bank were given as bounds. This produced an optimal structure based on Cu and optimal Cu flow rates between the banks. Subsequently, given the process structure, an iterative procedure was used to calculate the flow rates of the remaining mineral species. It is noteworthy that the Cu flow rates after this iterative procedure were not the same as those produced by the linear programming procedure. This is obvious, since the computed structure is based on Cu only and not on all the mineral species in the circuit. Furthermore, the circuits produced by this method are difficult to apply in practice owing to their complexity. These difficulties are overcome in the simulation routine proposed in this paper.

SIMULATION ROUTINE

The mass balance equations proposed here are an extension of those defined by Green (1984). It is clear that the mass balance equations for a system of \( k \) mineral species are highly non-linear. These non-linear equations must be linearised in order to apply linear programming methods. This is accomplished by rewriting them as two sets of linear equations, which form the basis of two linear simulation models, one being a subset of the other.

The first linear simulation model produces an optimal configuration and optimal total flow rates between the banks of separators. This solution is a function of the mineral separation in each bank, the plant size and constraints on the grades. The second linear simulation model produces the flow rates of the remaining mineral species subject to the first solution. This model will be discussed separately below.

In contrast to most other simulators, this simulator includes the separation characteristics of a mineral species at a particular bank as bounds and not as a model included in the mass balance equations. This is a realistic approach if it is considered that most simulation models for mineral separators are not fundamental, but empirical, and suitable for a particular application only. For example, the operation of gravity concentration cones is represented mostly by separation curves and not by fundamental models, which are, if they exist, based on numerous assumptions (Sivamohan and Forssberg, 1985). The separation characteristics of a gravity concentration cone can then be deduced, within bounds, from these separation curves for a particular type of cone.

The two linear simulation models must both be solved with the use of linear programming. They will be discussed separately below.

The Formulation of Linear Programming Simulation Model 1

This model will be discussed with reference to the different constraints imposed on the system.

**Mass balance constraints.** The mass balance constraints define the total flow rates between the banks of mineral separators. Consider the two separator banks \( i \) and \( j \) in Fig. 1. The middlings stream \( z \) could recycle around any bank \( j \), but would not constitute a product from the circuit. This is the case in a circuit of gravity separation cones (Alexis, 1978; Van der Spuy, 1982).

From Fig. 1 the following four steady-state mass balance equations may be derived:

\[
\begin{align*}
\sum_{i,i \neq j} r_{mi} + \sum_{i,i \neq j} r_{yi} + \sum_{i} r_{zi} + u_i &= m_i + y_i + z_i \quad (1) \\
\sum_{i,i \neq j} r_{mi} + b_j &= m_j \quad (2) \\
\sum_{i} r_{zi} &= z_j \quad (3) \\
\sum_{i,i \neq j} r_{yi} + a_j &= y_j \quad (4)
\end{align*}
\]

**External constraints.** These constraints include the following:

(a) The total flow rates from the separator banks may not exceed certain values due to pipe diameter and pumping speed limitations [constraints (5)–(7) below].

(b) The flow to a flotation bank must not be too low or too high, since this affects the retention time (cell volume/flow rate) and hence the separation characteristics of the separator [constraints (5)–(8) below].

\[
\begin{align*}
\sum_{i,j} r_{mi} + \sum_{j} r_{yi} + \sum_{j} r_{zi} &
\end{align*}
\]

![Fig. 1. Steady-state total flow rates from bank \( j \) to \( i \).](Stellenbosch University https://scholar.sun.ac.za)
Optimal design of mineral separation circuits

Note: (a) and (b) must be considered together.

(c) The designer must establish which separators produce concentrate and which produce tailings [constraints (9) and (10)].

(d) Excluding certain recycle streams. For example, a recycle of final product back to the rougher bank would be pointless. Conventional flotation does not produce a middlings stream, although this could be possible in column flotation cells and banks. If the design objective is to produce novel circuit configurations, these constraints may be neglected [constraints (11)–(13) below].

(e) Due to production limitations, the feed rate to the plant cannot exceed certain limits. Note that the feed can be a multiple feed, i.e. it may be fed to all j separator banks, if required [constraint (14) below].

(f) Grade constraints, in other words the product must be of a particular grade [constraint (15) below].

The above may be summarized by the following constraints:

\[ m_j^L \leq m_j \leq m_j^U \]  \hspace{1cm} (5)

\[ z_j^L \leq z_j \leq z_j^U \]  \hspace{1cm} (6)

\[ y_j^L \leq y_j \leq y_j^U \]  \hspace{1cm} (7)

\[ \tau_j = \left[ V_j \mu_j s_j \right] / \left[ m_j + z_j + y_j \right]. \]  \hspace{1cm} (8)

For the production of concentrate and tailings:

\[ a_j^L \leq a_j \leq a_j^U \]  \hspace{1cm} (9)

\[ b_j^L \leq b_j \leq b_j^U \]  \hspace{1cm} (10)

For particular recycle streams \( ij \) that are not permissible, the following three constraints are necessary:

\[ r_{mij} = 0 \]  \hspace{1cm} (11)

\[ r_{zij} = 0 \]  \hspace{1cm} (12)

\[ r_{yij} = 0. \]  \hspace{1cm} (13)

For the (multiple) feed to the plant, the following constraint may be formulated:

\[ u_j^L \leq u_j \leq u_j^U. \]  \hspace{1cm} (14)

The bounds on the tailings grade are defined by

\[ (G^T_j)^L \leq \frac{m_j}{m_j^U} \leq (G^T_j)^U. \]  \hspace{1cm} (15)

Separator constraints. The separation of the feed into concentrate, middlings and tailings characterizes this constraint, which may be established either by the application of theoretical models, separation curves, empirical models or from plant experience.

The separation characteristics of the separator \( j \) may be defined in terms of the separation factors \( y_{mkj} \) and \( z_{mkj} \) as follows:

\[ y_{mkj} = \frac{y_{kj}}{m_{kj}} \]  \hspace{1cm} (16)

\[ z_{mkj} = \frac{z_{kj}}{m_{kj}}. \]  \hspace{1cm} (17)

Due to variations in the conditions in separator bank \( j \), the separation factors may vary between an upper and lower bound. Hence, if the separation factors are not known precisely, constraints (18) and (19) may replace equalities (16) and (17):

\[ (y_{mkj})^L m_{kj} \leq y_{kj} \leq (y_{mkj})^U m_{kj} \]  \hspace{1cm} (18)

\[ (z_{mkj})^L m_{kj} \leq z_{kj} \leq (z_{mkj})^U m_{kj}. \]  \hspace{1cm} (19)

The separation characteristics affect the magnitude of the total streams from the separator. The equations which define the separation into total streams are

\[ y_j = \left( \sum_k G^T_j y_{mkj} \right) m_j \]  \hspace{1cm} (20)

\[ z_j = \left( \sum_k G^T_j z_{mkj} \right) m_j. \]  \hspace{1cm} (21)

If \( G^T_j, y_{mkj} \) and \( z_{mkj} \) are bounded [constraints (15)–(19)], the following two constraints may replace constraints (20) and (21):

\[ (YM_{kj})^L m_j \leq y_j \leq (YM_{kj})^U m_j \]  \hspace{1cm} (22)

\[ (ZM_{kj})^L m_j \leq z_j \leq (ZM_{kj})^U m_j. \]  \hspace{1cm} (23)

Object function. Given constraints (1)–(7), (9)–(14), (20) and (21), or (22) and (23), the objective of this simulation routine is to maximize the total concentrate recovery. This can be stated as

\[ OBJ = \sum_j w_j a_j \]  \hspace{1cm} (24)

where OBJ is the production cost of total concentrate; and \( w_j \) is the cost to produce the total concentrate \( a_j \), which depends on current operating costs (e.g. 100 monetary units/tonne).

Linear Programming Simulation Model I. The linear simulation model will consist of eqs (1)–(7), (9)–(14), (20) and (21), or (22) and (23), and (24). This model produces the optimal structure, the optimal total inter-bank flow rates and the optimal concentrate and tailings flow rates.

The Formulation of Linear Programming Simulation Model II

The minerals being concentrated in a separator are denoted as species \( k \). The manner in which a species \( k \) is defined depends on the type of separation circuit under consideration. In a flotation circuit, species may be defined on the basis of the percentage mineral content.
in a particle. In a gravity separation plant density may be used to distinguish between species.

The flow rates of the mineral species $k$ are simulated by the linear model which will be developed in this section. Most of the constraints of this second model are gleaned from the solution of Linear Programming Simulation Model I.

**Mass balance constraints.** The structure of the mass balance constraints for species $k$ in the separator circuit is similar to that developed for Linear Programming Simulation Model I. Mass balance constraints produced by Linear Programming Simulation Model I must also be included. The same separation factors and grade constraints used in Linear Programming Simulation Model I are used here.

From Fig. 2, the following four mass balance equations may be derived for species $k$ flowing between separators bank $i$ and $j$ (variables denoted by an $'$ have been evaluated by Linear Programming Simulation Model 1):

$$
\sum_{j \neq i} \sigma_{ij} m_{kj} + \sum_{j \neq i} e_{ij} z_{kj} + \sum_{j \neq i} c_{ij} y_{kj} + u_{ki} = m_{ki} + y_{ki} + z_{ki} \quad (25)
$$

$$
\sum_{j \neq i} \sigma_{ij} m_{kj} + \beta_{ij} m_{kj} = m_{kj} \quad (26)
$$

$$
\sum_{i} e_{ij} z_{kj} = z_{kj} \quad (27)
$$

**External constraints.** The only external constraint imposed on this linear simulation model is the grade constraint on the flow rates of the tailings, i.e. constraint (15).

$$
\sum_{i} \sigma_{ij} y_{kj} + \alpha_{ij} y_{kj} = y_{kj} \quad (28)
$$

The bounds on the feed rate are given by

$$
u_{ij}^L \leq u_{ij} \leq u_{ij}^U \quad (29)
$$

The following five constraints are produced as a result of Linear Programming Simulation Model I:

$$
m_{kj}^* = \sum_{k} m_{kj} \quad (30)
$$

$$
z_{kj}^* = \sum_{k} z_{kj} \quad (31)
$$

$$
y_{kj}^* = \sum_{k} y_{kj} \quad (32)
$$

$$
b_{kj}^* = \sum_{k} \beta_{ij} m_{kj} \quad (33)
$$

$$
u_{kj}^* = \sum_{k} u_{kj} \quad (35)
$$

*Fig. 2. Steady-state flow rates of species $k$ between banks $i$ and $j.*
Separator constraints. These constraints are similar to those imposed on Linear Programming Simulation Model I [constraints (16) and (17) or (18) and (19)].

Objective function. Given the constraints (15)–(19), and (25)–(35), the objective is to maximize the recovery of the valuable mineral species $k$. This can be stated as

$$\text{OBJM} = \sum_{j,k} w_{kj} a_{kj}$$  \hspace{1cm} (36)

where OBJM is the total earnings due to the recovery of the valuable mineral species $k$ from banks $j$; and $w_{kj}$ is the price weight for mineral species $k$ at bank $j$, which is based on current mineral prices (e.g. 200 monetary units/tonne).

Linear Programming Simulation Model II. This model consists of eqs (15)–(19), and (25)–(36).

APPLICATION ALGORITHM

This section will explain the application of the linear programming simulation models defined above.

(a) Establish the separation characteristics of the separator $j$ by applying the defining eqs (16) and (17). The separation characteristics of a separator may be established from pilot plant data, separation curves, theoretical and empirical separation models, or separation plant experience. For example, the flotation rate ($k_{ij}$), the number of flotation cells ($N_j$) and the retention time ($\tau_j$) may be related to the separation factor $y_{mj}$ (Lynch et al., 1981) by the theoretical flotation model:

$$y_{mj} = [(1 + k_{ij} \tau_j)^{N_j} - 1].$$  \hspace{1cm} (37)

Multiple linear regression models, which define the separation factors, may also be implemented. These may even be included in the simulation models.

If in the course of batch flotation or pilot flotation plant work it was established that $k_{ij}$, $\tau_j$ and $N_j$ vary between certain limits for a particular bank $j$, it is clear that an upper and lower bound for $y_{mj}$ may be established for constraint (18). Hence constraint (18) replaces equality (16) in this instance. A similar approach will produce a constraint for the separation factor $z_{mj}$.

(b) Determine the bounds on constraint (15). This depends on the production grade requirements and product specifications set by the designer.

(c) Define the constraints given by eqs (20) and (21), or (22) and (23), using the constraints developed in (a) and (b) above.

(d) Define the bounds imposed on the feed rate to the separation plant as well as the bounds on the product flow rates from each separator bank $j$ [constraints (14) and (5)–(7)].

(e) Establish the bounds imposed on the recycle streams, i.e. to exclude those recycle streams that are not feasible. For example, middlings streams in flotation cells are not a general feature of industrial flotation plants.

(f) Use a linear programming package to solve Linear Programming Simulation Model I.

(g) From the solution of the Linear Programming Simulation Model I obtain the structural parameters and total flow rates and define the constraints given by eqs (25)–(35).

(h) The same tailings grade constraints [constraint (15)] and separation constraints [constraints (16) and (17), or (18) and (19)] used in Linear Programming Simulation Model I are used in the formulation of Linear Programming Simulation Model II.

Note: It is not a prerequisite to define the grade constraints here, since they have in fact been included in the formulation of Linear Programming Simulation Model I.

(i) Use a linear programming package to solve Linear Programming Simulation Model II in order to produce the flow rates of mineral species $k$.

(j) From the solution of the Linear Programming Simulation Model I and Linear Programming Simulation Model II the retention time and the flotation rate constants may be established, which ensure optimal recovery. This may be achieved by using the definition of the retention time [eq. (8)] and a separation model [e.g. eq. (37)].

SAMPLE PROBLEMS

Four solutions produced by the simulation model will be discussed, demonstrating its applicability during the design of separation plants.

Examples 1–3 demonstrate the simulation of flotation plants and example 4 demonstrates on application to a gravity separation plant. Note that in the formulation of the simulation model for a flotation circuit, the middlings streams are excluded.

For the purpose of illustration, the use of Linear Programming Simulation Models I and II will be explained in example 1 in more detail than in the case in examples 2–4.

Example 1

This example demonstrates a very simple rougher-cleaner flotation circuit $\{j, i: 1, 2\}$, which separates species $k = 1$ (100% chalcopyrite), species $k = 2$ (90% silica, 10% chalcopyrite) and species $k = 3$ (100% silica) into concentrate and tailings streams. The feed flow rates to the rougher bank $(j = 1)$ of species $1$–$3$ are $u_{11} = 6$ t/h, $u_{21} = 3$ t/h and $u_{31} = 291$ t/h, respectively. These data are taken from Lynch et al. (1981).

By the application of eq. (37) and taking variations of the retention time, flotation rates and the number of cells in a bank into consideration, the following bounds on the separation factors for species $1$–$3$ at the rougher bank $(j = 1)$ could be produced: $1 \leq y_{m11} \leq 24.0$, $0.01 \leq y_{m21} \leq 1$ and $0.1 \leq y_{m31} \leq 0.3$. For the cleaner bank $(j = 2)$ the separation factors are: $3.1 \leq y_{m12} \leq 4.0$, $0 \leq y_{m22} \leq 0.127$ and $0 \leq y_{m32} \leq 0.006$, respectively. The tailings grade constraints [constraint (15)] for mineral species $k$ at each bank $j$
are defined as the following (designer's choice):

\[
0 \leq \frac{m_{11}}{m_1} \leq 0.001 \quad 0.010 \leq \frac{m_{12}}{m_2} \leq 0.0158
\]
\[
0 \leq \frac{m_{21}}{m_1} \leq 0.0095 \quad 0.005 \leq \frac{m_{22}}{m_2} \leq 0.0265
\]
\[
0.832 \leq \frac{m_{31}}{m_1} \leq 0.992 \quad 0.950 \leq \frac{m_{32}}{m_2} \leq 0.96.
\]

Subject to the tailing grades defined above and the separation factors, constraint (22) for banks \( j = 1 \) and \( j = 2 \) could be defined as follows:

\[
0.0832m_1 \leq y_1 \leq 0.33m_1
\]
\[
0.0310m_2 \leq y_2 \leq 0.072m_2.
\]

From the above data the following simulation models can be constructed for a two-bank flotation plant.

### Linear Programming Simulation Model I

<table>
<thead>
<tr>
<th>Equations</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBJ = ( \sum w_j a_j )</td>
<td>Objective function: ( w_1 = 100 ) monetary units/tonne</td>
</tr>
<tr>
<td>( r_{m_{21}} + r_{y_{21}} + u_2 = m_2 + y_2 ) ( r_{y_{12}} + r_{y_{21}} + u_1 = m_1 + y_1 ) ( r_{m_{12}} + r_{2} + b_1 = m_1 ) ( r_{y_{12}} + a_2 = y_2 ) ( r_{y_{21}} + a_1 = y_1 ) ( \text{0.001} \leq m_j \leq 0.440 ) ( \text{0.001} \leq y_j \leq 0.440 ) ( 0 \leq u_k \leq 300 ) ( 0 \leq a_k \leq 0 ) ( 0 \leq a_2 \leq 20 ) ( 0 \leq b_1 \leq 20 ) ( 0 \leq b_2 \leq 0 ) ( r_{y_{12}} = 0 )</td>
<td>Top bound corresponds to maximum pumping capacity ( \text{Feed to rougher (fixed)} ) ( \text{No concentrate from bank 1 (rougher)} ) ( \text{No tailings from cleaner} ) ( \text{No concentrate recycle from cleaner to rougher} )</td>
</tr>
</tbody>
</table>

The following structural parameters are produced by Linear Programming Model I (consult Fig. 3):

\[ t_{12}^* = 1; t_{11}^* = 0; c_{21}^* = 0; c_{22}^* = 1; \sigma_{C1}^* = 0; \]
\[ \sigma_{C2}^* = 1; \beta_{T1}^* = 1; \beta_{T2}^* = 0. \]

The total flow rates, which are also produced by this simulation model, may be obtained from Fig. 3.

### Linear Programming Simulation Model II

<table>
<thead>
<tr>
<th>Equations</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBJ = ( \sum w_{kj} a_{kj} )</td>
<td>Objective function: the value of ( w_{kj} ) depends on current mineral prices ( w_{kj} = 200 ) monetary units/tonne</td>
</tr>
<tr>
<td>( t_{12}^* m_{12} + c_{21}^* y_{11} + u_{11} = m_{22} + y_{12} ) ( t_{12}^* m_{12} + c_{21}^* y_{12} + u_{11} = m_{22} + y_{12} ) ( t_{12}^* m_{12} + \beta_{T1}^* y_{11} = m_{22} )</td>
<td>See structural parameters above for these constraints: also consult Fig. 1</td>
</tr>
<tr>
<td>( t_{12}^* m_{12} + \beta_{T2}^* m_{12} = m_{22} ) ( c_{21}^* y_{11} + \sigma_{C1}^* y_{11} = y_{12} ) ( c_{21}^* y_{12} + \sigma_{C2}^* y_{12} = y_{22} ) ( 6 \leq u_{11} \leq 6 ) ( 6 \leq u_{12} \leq 6 ) ( 3 \leq u_{11} \leq 3 ) ( 191 \leq u_{12} \leq 191 ) ( 293.495 = m_{11} ) ( 90.349 = m_{12} ) ( 96.854 = y_{11} ) ( 6.505 = y_{12} ) ( 293.495 = b_{1}^* ) ( 6.505 = a_{2}^* ) ( \beta_{T1}^* = 1 ) ( \beta_{T2}^* = 0 ) ( \alpha_{C1}^* = 0 ) ( \alpha_{C2}^* = 1 )</td>
<td>Feed rates of each species ( k ) to bank ( j ) (fixed)</td>
</tr>
<tr>
<td>( 300.000 = u_{1}^* ) ( 1m_{11} \leq y_{11} \leq 24m_{11} ) ( 0.01m_{12} \leq y_{11} \leq 1m_{12} ) ( 0.01m_{12} \leq y_{21} \leq 0.3m_{12} ) ( 1.2m_{12} \leq y_{11} \leq 4m_{12} ) ( 0m_{22} \leq y_{12} \leq 0.127m_{22} ) ( 0m_{22} \leq y_{22} \leq 0.006m_{22} )</td>
<td>The separation factors for the different species ( k ) at each bank ( j )</td>
</tr>
</tbody>
</table>

The flow rates of the mineral species, which are produced by this model, are depicted by Fig. 3.

**Process variables.** From the results depicted by Fig. 3, the following calculations regarding the retention time and flotation conditions may be done:

- **(a) Retention time**
  Assume the following data (Lynch et al. 1981):
  - Mass percent solids: 40%
  - Specific gravity of species 1: 4.20
  - Specific gravity of species 2: 2.81
  - Specific gravity of species 3: 2.65.

  From these data the relative pulp density of the rougher and cleaner feeds could be calculated as 1.34.

  **Volume of a rougher cell:** 15 m³
  **Volume of a cleaner cell:** 1.5 m³.

  From eq. (8) the retention time in a rougher cell is calculated as 1.23 min and in a cleaner cell 0.5 min.
Optimal design of mineral separation circuits

All flow rates tonnes/h

<table>
<thead>
<tr>
<th>Feed Species</th>
<th>Concentrate Species</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: 6.0</td>
<td>1: 5.703</td>
</tr>
<tr>
<td>2: 3.0</td>
<td>2: 0.304</td>
</tr>
<tr>
<td>3: 291.0</td>
<td>3: 0.498</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tailings to Rougher Species</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: 0.297</td>
</tr>
<tr>
<td>2: 2.696</td>
</tr>
<tr>
<td>3: 290.502</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Concentrate to Cleaner Species</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: 7.129</td>
</tr>
<tr>
<td>2: 2.697</td>
</tr>
<tr>
<td>3: 87.028</td>
</tr>
</tbody>
</table>

Fig. 3. Circuit structure and flow rates for example 1.

(b) Flotation rate constants in the rougher
By the application of eq. (37) and assuming that the rougher bank consists of 16 cells, the following rate constants can be calculated:

\[ k_{11} = 0.180 \text{ min}^{-1} \]
\[ k_{21} = 0.036 \text{ min}^{-1} \]
\[ k_{31} = 0.013 \text{ min}^{-1} . \]

(c) Flotation rate constants in the cleaner
The following rate constants can be calculated for 10 cleaner cells in a similar way to (b) above:

\[ k_{12} = 0.350 \text{ min}^{-1} \]
\[ k_{22} = 0.020 \text{ min}^{-1} \]
\[ k_{32} = 0.001 \text{ min}^{-1} . \]

Hence, for an optimal flotation circuit and optimal recovery, the chemical environment in a flotation cell must be such that it supports these calculated flotation rates in the rougher and cleaner cells. These flotation rates are of the same order as those given by Lynch et al. (1981).

Example 2
This example illustrates a simulation of a four-bank flotation circuit. The circuit consists of a rougher, a scavenger, a cleaner and a recleaner bank. By the application of eq. (37) the separation factors for species 1–3 at the rougher bank \( (j = 1) \) could be produced:

\[ 1 \leq y_{m11} \leq 24.0, \ 0.01 \leq y_{m21} \leq 1 \text{ and } 0.1 \leq y_{m31} \leq 0.58 . \]

For the scavenger and cleaner banks \( (j = 2–4) \) the separation factors are: \( 3.0 \leq y_{m12} \leq 4.0, \ 0.01 \leq y_{m22} \leq 0.127 \) and \( 0 \leq y_{m32} \leq 0.2 \), respectively. The bounds were produced by taking variations of the number of cells, retention time and flotation rates into account.

From appropriate tailing grades [constraint (15)] and the separation factors above, the following constraints for banks 1–4 may be defined [constraint (22)]:

\[ 0.0832m_1 \leq y_1 \leq 0.600m_1 \]
\[ 0.0310m_2 \leq y_2 \leq 0.200m_2 \]
\[ 0.0310m_3 \leq y_3 \leq 0.100m_3 \]
\[ 0.0310m_4 \leq y_4 \leq 0.450m_4 . \]

From the above data the respective simulation models can be formulated. The results for this example are depicted by Fig. 4. The other process variables may be calculated as was done in example 1.

Example 3
The data used for the simulation of a four-bank flotation plant was taken from Green (1984). In this plant a feed consisting of 3.28 kg/min of species 1 (Cu), 1.72 kg/min of species 2 (Ni) and 1309 kg/min of species 3 (gangue) are separated into concentrate and tailings. It was established that the separation factors for each species produce the following constraints
**Example 4**

The data for this example were taken from Van der Spuy (1982), an example of a gravity separation plant separating a feed of 0.015 t/h of species 1 (uranotherianate), 1.68 t/h of species 2 (ZrO₂) and 418.305 t/h of species 3 (gangue) into concentrate and tailings.

From the data the following separation factors could be identified:

\[ y_{m1j} = 3 \]
\[ y_{m2j} = 2 \]
\[ z_{m1j} = 4 \]
\[ z_{m2j} = 1 \]

Furthermore, from the given material balance (Van der Spuy, 1982), the following bounds can be defined for constraints (22) and (23):

\[ 0.4 m_j \leq y_j \leq 0.6 m_j \]
\[ 0.2 m_j \leq z_j \leq 0.4 m_j \]

The results for this example are depicted by Fig. 6.

**DISCUSSION**

**Example 1**

This example shows a practically feasible circuit configuration.

The Cu grade produced is 30.5% and the recovery 91%, very close to the values given by Lynch et al.
Optimal design of mineral separation circuits

Fig. 5. Circuit structure and flow rates for example 3.

(1981) for the same flotation circuit configuration and feed rates.

Example 2
The simulation produces a four-bank flotation circuit which shows the function of a recleaner flotation bank very clearly. The circuit configuration is typical of flotation plant practice. A concentrate Cu grade of 32.4% and a Cu recovery of 94.3% is achieved by the simulation.

Example 3
The circuit configuration produced by Green (1983) is profoundly different from that produced here for the same data. Furthermore, the circuit shown here corresponds closely with circuit configurations suggested by Lynch et al. (1981).

Example 4
The circuit depicted here shows the recycle of middlings around the rougher and cleaner Reichert cones used for gravity concentration. This is normal practice in gravity concentration.

The solution produced a recovery of 56% of the valuable mineral species 1, which is in close agreement to the recovery of 60% reported by Van der Spuy (1982). Only a section of the plant described by Van der Spuy was simulated, hence the lower recovery. The circuit configuration exhibited by this example is very typical of gravity separation plants.

CONCLUSIONS AND SIGNIFICANCE
From the application examples it can be concluded that:
(a) Circuit configurations produced by the simulation model are similar to those encountered in industry, hence permitting the application of these simulation models in the design of separation plants. The grades and recoveries obtained correspond closely with those of similar separation plants in industry.
(b) The simulation models are very time efficient, producing solutions within seconds, making them superior to time consuming iterative numerical procedures.
(c) The only data required by the simulation models are the separation characteristics, tailings grade constraints and the feed rates.

Note: The model can never produce negative flow rates as may be the case in some iterative procedures.
(**Fig. 6. Circuit structure and flow rates for example 4.)**

(d) Approximate separation data may be used during the formulation of the simulation models. The solutions are not subject to inaccurate separation models.

*Note:* Exact separation data are not a prerequisite. This is meaningful, since in the designing stages of a plant exact separation data may not be available.

(e) The simulation model supports a multiple feed strategy, if required.

These points reflect a simulation model that may be useful in the design of separation plants. Subsequently, once the plant has been commissioned, the model may be used to simulate its operation. If conditions change drastically in the plant, the simulation model may suggest an alternative circuit configuration and new operating conditions quickly.

**NOTATION**

*Variables of Linear Programming Simulation Models I and II*

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_i$</td>
<td>total concentrate recovery from separator $j$</td>
</tr>
<tr>
<td>$a_{iL}$</td>
<td>lower bound on $a_i$</td>
</tr>
<tr>
<td>$a_{iU}$</td>
<td>upper bound on $a_i$</td>
</tr>
<tr>
<td>$a_{iL}$</td>
<td>total tailings recovery from separator $j$</td>
</tr>
<tr>
<td>$a_{iL}$</td>
<td>lower bound on $a_i$</td>
</tr>
<tr>
<td>$a_{iU}$</td>
<td>upper bound on $a_i$</td>
</tr>
<tr>
<td>$b_j$</td>
<td>grade of mineral species $k$ in tailings stream $j$</td>
</tr>
<tr>
<td>$b_{jL}$</td>
<td>lower bound on $b_j$</td>
</tr>
<tr>
<td>$b_{jU}$</td>
<td>upper bound on $b_j$</td>
</tr>
<tr>
<td>$G_{k}^{T}$</td>
<td>Reaction rate of mineral species $k$ at separator bank $j$</td>
</tr>
<tr>
<td>$G_{k}^{T}$</td>
<td>lower bound on tailings grade</td>
</tr>
<tr>
<td>$G_{k}^{T}$</td>
<td>upper bound on tailings grade</td>
</tr>
<tr>
<td>$k_{aj}$</td>
<td>number of subunits in separator bank $j$</td>
</tr>
<tr>
<td>$m_j$</td>
<td>total tailings from separator $j$</td>
</tr>
<tr>
<td>$m_{jL}$</td>
<td>lower bound on $m_j$</td>
</tr>
<tr>
<td>$m_{jU}$</td>
<td>upper bound on $m_j$</td>
</tr>
<tr>
<td>$N_j$</td>
<td>number of subunits in separator bank $j$</td>
</tr>
<tr>
<td>OBJ</td>
<td>total earnings due to the recovery of valuable mineral species $k$</td>
</tr>
<tr>
<td>OBJM</td>
<td>total cost to produce concentrate</td>
</tr>
</tbody>
</table>

**Feed**

<table>
<thead>
<tr>
<th>Species</th>
<th>Flow Rate (tonnes/h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.015</td>
</tr>
<tr>
<td>2</td>
<td>1.68</td>
</tr>
<tr>
<td>3</td>
<td>418.305</td>
</tr>
</tbody>
</table>

**Concentrate to cleaner**

<table>
<thead>
<tr>
<th>Species</th>
<th>Flow Rate (tonnes/h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0056</td>
</tr>
<tr>
<td>2</td>
<td>0.84</td>
</tr>
<tr>
<td>3</td>
<td>105.154</td>
</tr>
</tbody>
</table>

**Tailings**

<table>
<thead>
<tr>
<th>Species</th>
<th>Flow Rate (tonnes/h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0019</td>
</tr>
<tr>
<td>2</td>
<td>0.42</td>
</tr>
<tr>
<td>3</td>
<td>223.5781</td>
</tr>
</tbody>
</table>

**Middlings to scavenger**

<table>
<thead>
<tr>
<th>Species</th>
<th>Flow Rate (tonnes/h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0075</td>
</tr>
<tr>
<td>2</td>
<td>0.42</td>
</tr>
<tr>
<td>3</td>
<td>85.5725</td>
</tr>
</tbody>
</table>

**Concentrate**

<table>
<thead>
<tr>
<th>Species</th>
<th>Flow Rate (tonnes/h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0084</td>
</tr>
<tr>
<td>2</td>
<td>0.7467</td>
</tr>
<tr>
<td>3</td>
<td>39.2469</td>
</tr>
</tbody>
</table>

**Tailings to cleaner**

<table>
<thead>
<tr>
<th>Species</th>
<th>Flow Rate (tonnes/h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0056</td>
</tr>
<tr>
<td>2</td>
<td>0.28</td>
</tr>
<tr>
<td>3</td>
<td>33.7144</td>
</tr>
</tbody>
</table>

**Middlings to recycle**

<table>
<thead>
<tr>
<th>Species</th>
<th>Flow Rate (tonnes/h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0028</td>
</tr>
<tr>
<td>2</td>
<td>0.3733</td>
</tr>
<tr>
<td>3</td>
<td>99.6239</td>
</tr>
</tbody>
</table>

**Middlings**

<table>
<thead>
<tr>
<th>Species</th>
<th>Flow Rate (tonnes/h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0113</td>
</tr>
<tr>
<td>2</td>
<td>0.3733</td>
</tr>
<tr>
<td>3</td>
<td>19.6154</td>
</tr>
</tbody>
</table>

**Cleaner Bank**

<table>
<thead>
<tr>
<th>Concentrate Species</th>
<th>Flow Rate (tonnes/h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0084</td>
</tr>
<tr>
<td>2</td>
<td>0.7467</td>
</tr>
<tr>
<td>3</td>
<td>39.2469</td>
</tr>
</tbody>
</table>

**Scavenger Bank**

<table>
<thead>
<tr>
<th>Concentrate Species</th>
<th>Flow Rate (tonnes/h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0056</td>
</tr>
<tr>
<td>2</td>
<td>0.28</td>
</tr>
<tr>
<td>3</td>
<td>33.7144</td>
</tr>
</tbody>
</table>

**Rougher Bank**

<table>
<thead>
<tr>
<th>Concentrate Species</th>
<th>Flow Rate (tonnes/h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0028</td>
</tr>
<tr>
<td>2</td>
<td>0.3733</td>
</tr>
<tr>
<td>3</td>
<td>99.6239</td>
</tr>
</tbody>
</table>
Optimal design of mineral separation circuits

\[ \begin{align*}
\text{rm}_{ij} & \quad \text{total recycle of tailings from separator bank j to i} \\
\text{ry}_{ij} & \quad \text{total recycle of concentrate from separator bank j to i} \\
\text{rz}_{ij} & \quad \text{total recycle of middlings from separator bank j to i} \\
\rho_j & \quad \text{pulp density at bank j} \\
s_j & \quad \text{mass fraction of solids in pulp at bank j} \\
\tau_j & \quad \text{retention time of a separator cell j} \\
u_j & \quad \text{total feed to separator j} \\
u_j^l & \quad \text{lower bound on } u_j \\
u_j^u & \quad \text{upper bound on } u_j \\
u_{kj} & \quad \text{feed of mineral species k to separator bank j} \\
u_{kj}^l & \quad \text{lower bound on } u_{kj} \\
u_{kj}^u & \quad \text{upper bound on } u_{kj} \\
u^J & \quad \text{bank volume} \\
w_j & \quad \text{price weight of } a_j; \text{depends on current production costs} \\
w_{kj} & \quad \text{price weight of mineral species k from separator bank j; depends on current mineral costs} \\
y_j & \quad \text{total concentrate from separator j} \\
y_{kj} & \quad \text{concentrate flow of mineral species k from separator bank j} \\
y_j^l & \quad \text{lower bound on } y_j \\
y_j^u & \quad \text{upper bound on } y_j \\
y_{kj}^l & \quad \text{lower bound on } y_{kj} \\
y_{kj}^u & \quad \text{upper bound on } y_{kj} \\
(y_{mkj})^{L} & \quad \text{lower bound on } y_{mkj} \\
(y_{mkj})^{U} & \quad \text{upper bound on } y_{mkj} \\
(Y_{Mkj})^{L} & \quad \text{lower bound on } \sum G_{ij}^L y_{mkj} \\
(Y_{Mkj})^{U} & \quad \text{upper bound on } \sum G_{ij}^U y_{mkj} \\
z_j & \quad \text{middlings from separator j} \\
z_{kj} & \quad \text{middlings flow of mineral species k from separator bank j} \\
z_j^l & \quad \text{lower bound on } z_j \\
z_j^u & \quad \text{upper bound on } z_j \\
z_{kj}^l & \quad \text{lower bound on } z_{kj} \\
z_{kj}^u & \quad \text{upper bound on } z_{kj} \\
(z_{mkj})^{L} & \quad \text{lower bound on } z_{mkj} \\
(z_{mkj})^{U} & \quad \text{upper bound on } z_{mkj} \\
(Z_{Mkj})^{L} & \quad \text{lower bound on } \sum G_{ij}^L z_{mkj} \\
(Z_{Mkj})^{U} & \quad \text{upper bound on } \sum G_{ij}^U z_{mkj} \\
\end{align*} \]

**Constants produced by Model 1**

\[ \begin{align*}
a_{ykj} & \quad \text{fraction of } y_j \text{ that is concentrate product} \\
a_j^* & \quad \text{total concentrate flow rate from separator j} \\
b_{mj}^* & \quad \text{fraction of } m_j \text{ that is tailings product} \\
b_j^* & \quad \text{total tailings flow rate from separator j} \\
\end{align*} \]

REFERENCES


The Use of Linear Programming in the Optimal Design of Flotation Circuits Incorporating Regrind Mills

M.A. REUTER and J.S.J. VAN DEVENTER
Institute for Non-Ferrous Process Metallurgy, Technical University Aachen, Intzestrasse 3, D-5100 Aachen (F.R. Germany)
Department of Metallurgical Engineering, University of Stellenbosch, Stellenbosch 7600 (South Africa)

(Received April 12, 1988: accepted after revision July 25, 1989)

ABSTRACT


Two linear models, the second being a subset of the first, are proposed for the simulation of flotation plants by use of linear programming. The first linear model produces the circuit structure, as well as the optimal flow rates of the valuable element between any number of flotation banks incorporating any number of recycle mills. An optimal grade for the valuable element in the concentrate is given by the second model. Operating conditions in the flotation banks and recycle mills are included as bounds in these models, permitting their possible application in expert systems. The simulated circuit structure, concentrate grade and recoveries closely resemble those of similar industrial flotation plants. The only data required by the simulation models are the feed rates of the species of an element, and their separation factors which are estimated from a multi-parameter flotation model.

INTRODUCTION

Froth flotation is one of the most commonly used unit operations for the separation of valuable minerals from associated gangue impurities. The process is based on the property of some mineral particles to adhere stronger than others to air bubbles formed by an air flow in chemically treated water. Flotation plants usually consist of cells which are grouped into banks and interconnected in a predefined manner (Lynch et al., 1981). Although considerable research has been published on the chemically and process-engineering of flotation, it is still extremely difficult to design and control such plants.

Fine grinding is normally required to adequately liberate a fine-grained ore before flotation (Johnson, 1987). Over-grinding of the ore could cause lower
recoveries of the valuable mineral, increased entrainment of hydrophilic gangue and consequently lower grades. Composite particles containing both valuable and gangue minerals cause problems in flotation. Many of the complexities of flotation circuits are due to attempts to find the most efficient way to treat composite particles so that recovery of the valuable minerals is maintained at a maximum while dilution of the concentrate by gangue is minimised (Lynch et al., 1981). In some flotation plants intermediate tailings and/or concentrate must be reground prior to further processing.

When designing flotation plants, it is attempted to produce an optimal structure as well as optimal operating conditions. Few authors (King, 1976; Sutherland, 1977) have considered the effect of regrind mills in their simulation routines, which are normally based on numerical methods.

Mehrotra and Kapur (1974) derived an optimal flotation structure from a generalized circuit by direct search methods. The simulator of King (1976) consisted of mass-balance equations defined for mineral classes in a fixed structure, which were solved iteratively. Niemi et al. (1982) included a recycle conditioning tank in the design of flotation networks. Williams and Meloy (1983) and Williams et al. (1986) used a technique called "circuit analysis" to simulate mineral separation processes. Unit operations were assumed to be linear, so that the ensuing linear algebraic equations could be solved without the use of any numerical methods.

Green (1982, 1984) and Green et al. (1985) developed mass balance equations in terms of split factors and bound on the flow rates in flotation banks. The resulting linear models could be solved by use of linear programming. An optimal structure was based on the valuable element, with the flow rates of the remaining mineral species being calculated by an iterative procedure. It is noteworthy that the flow rates of the valuable mineral after this iterative procedure were not the same as those produced by the linear programming procedure. Furthermore, the circuits produced by this method are difficult to apply in practice owing to their complexity.

In a previous paper, Reuter et al. (1988) optimised mineral separation circuits by solving two sequential linear models, but based the structure on total flow rates. However, regrind mills were not considered and the concentrate grades were not necessarily optimal.

It is the aim of the present paper to propose a general optimisation routine for flotation plants which incorporate regrind mills. Both the recovery and grade of valuable mineral in the concentrate will be optimised by use of two sequential linear programming models. Realistic solutions are yielded by this model, as will be illustrated by three sample problems.

BASIS OF SIMULATION MODEL

Significant progress has been made in recent years on the mathematical modelling of mills (Klimpel and Austin, 1984; Austin et al., 1987a, b). Never-
theless, more work is needed before these models could be used reliably. It is possible, however, to predict bounds on the size distribution caused by a mill.

Various methods for the measurement of mineral liberation and the modelling thereof have been proposed (King, 1979; Malvik, 1982; Holland-Batt, 1983; Finch and Petruk, 1984). It is possible to quantitatively estimate the mineral liberation for different particle sizes from image analyses on polished ore sections (King, 1979). Hence, bounds could be determined for the change in mineralogy within different particle sizes caused by milling.

Flotation is influenced not only by particle size and liberation, but also interactively by surface properties and conditions in the pulp and froth phases. Consequently, it is extremely difficult to predict flotation behaviour solely from milling and liberation models.

In this paper flotation species $s$ are defined by lumping milling, liberation and flotation response in one batch flotation test, where the recovery $R$ of element $k$ is given by a model such as (Lynch et al., 1981; Ross, 1984):

$$R = \Phi \left[ 1 - \{ (1 - \Omega) \cdot \exp(-k_j t) + \Omega \cdot \exp(-k_f t) \} \right]$$  \hspace{1cm} (1)

with $[1 - \Phi]$ being the non-floating, $[\Phi - \Omega]$ the slow-floating and $[\Omega(1 - \Omega)]$ the fast-floating species. Such flotation tests may be conducted on material from different positions in the flotation plant, and before and after a regrind mill. The effect of a mill will be simulated here by a matrix which transforms flotation species $l$ into $s$. Suitable models (Ross, 1988) may be used to convert the values of $k_l$ and $K_s$ to $k_{kj}$ for plant conditions. The element $k$ may also be a lumped entity such as the total sulphur content.

The separation factor $ym_{kj}$ of species $s$ at bank $j$ defined by:

$$ym_{kj} = \frac{y_{kj}^s}{m_{kj}^s}$$  \hspace{1cm} (2)

could be related to the mean retention time $\tau_j$ in a cell, the $N_j$ cells in bank $j$ and the rate constants $k_{kj}$ derived from eq. 1 (Lynch et al., 1981):

$$ym_{kj} = (1 + k_{kj} \cdot \tau_j)^{N_j} - 1$$  \hspace{1cm} (3)

Hence, bounds on $ym_{kj}$ could be estimated if bounds on $k_{kj}$ and $\tau_j$ are known, or derived from plant experience.

In contrast with most other simulators, this simulator includes the separation characteristics of a mineral species at a particular bank as bounds and not as a model included in the mass-balance equations. This is realistic if it is considered that most flotation, milling and liberation models are empirical and suitable for a particular application only. A knowledge-based system could be used to define and modify such bounds.

The mass-balance equations which define the flow rates of flotation species $s$ of elements $k$ and the structural parameters in a flotation plant are non-linear (Reuter et al., 1988), and must be linearised in order to use linear program-
ming methods. This is achieved here by structuring them as two sets of linear equations, the second set being a subset of the first.

An optimal circuit structure and flow rates for the valuable element between flotation banks are produced by the first linear model, which is based on the maximization of the recovery of the valuable element. The second linear model produces the flow rates of the non-valuable elements subject to the flow rates of the valuable element and circuit structure produced by the first linear model. In the second model the grade of the final concentrate is maximized, which is equivalent to the minimization of the recovery of the non-valuable elements. These models are described separately in the next two sections.

MAXIMIZATION OF THE RECOVERY OF THE VALUABLE ELEMENT

Linear Programming Model I optimises the flow rates and the path of the valuable element $h$ between the flotation banks and through the mills. This simulation model will be developed here by formulating all possible constraints imposed on the mass-balance equations.

Mass-balance constraints

The mass-balance equations define the flow rates of the valuable element $h$ between the flotation banks and through the mills. Since a particular element is conserved in a mill at steady state, the flow rate through it remains constant. However, the mill could affect the floatability of an element, including $h$, and hence its separation characteristics in the flotation banks. Therefore, the mill does affect the flow rates of concentrate and tailings of the valuable element $h$ from the flotation banks.

Fig. 1 defines the mass flow rates of the valuable element $h$ between two flotation banks $i$ and $j$. The three steady-state mass-balance equations for the valuable element $h$ are:

$$\sum_{j, i \neq j} rm_{ij}^h + \sum_{j, i \neq j} ry_{ij}^h + u_{hi} = m_{hi} + y_{hi} \quad (4)$$

$$\sum_{i, i \neq j} rm_{ij}^h + b_{hj} = m_{hj} \quad (5)$$

$$\sum_{i, i \neq j} ry_{ij}^h + a_{hj} = y_{hj} \quad (6)$$

External constraints

A number of external constraints should be imposed on the following three mass-balance equation.

1. The production capacity of a flotation bank limits the total flow rate of
Fig. 1. Steady state flow rates of the valuable element h from flotation bank j to i.

the valuable element h in its feed. These constraints also determine which banks produce concentrate and which produce tailings. If desired, fresh feed can be supplied to more than one flotation bank (eq. 7). The bounds on the (multiple) feed rate of the valuable element h, final concentrate flow rate and the final tailings flow rate are defined by the following constraints:

\[ u_{hj}^L \leq u_{hj} \leq u_{hj}^U \]  
(7)

\[ a_{hj}^L \leq a_{hj} \leq a_{hj}^U \]  
(8)

\[ b_{hj}^L \leq b_{hj} \leq b_{hj}^U \]  
(9)

(2) All the flow rates of the valuable element h from a flotation bank have a lower bound of zero to ensure that no flow rate becomes negative.

(3) Certain recycle streams \( ij \) which are not physically meaningful, must be excluded, e.g. recycling of rougher tailings to the cleaner banks (eqs. 10 and 11):

\[ rm_{ij}^h = 0 \]  
(10)

\[ ry_{ij}^h = 0 \]  
(11)
Flotation constraints

The degree of separation of the valuable element \( h \) into concentrate and tailings is determined by the efficiency of the flotation process. In this paper the valuable element \( h \) is characterized by three flotation species \( s \), i.e. a fast floating, a slow floating and a non-floating species. Bounds on separation factors \( ym_{hj}^s \) may be obtained from operational experience, or by fitting a flotation model such as eq. 1 to batch or plant data:

\[
 ym_{hj}^s \leq y_{hj} \leq (ym_{hj}^s)^U m_{hj}^s \quad (12)
\]

By defining the separation factor \( ym_{hj}^s \) as bounds, the optimization model becomes well-suited for inclusion in a knowledge-based simulator that permits the intelligent choice of bounds.

Milling affects the flotation rate constants, and hence the separation factor \( ym_{hj}^s \), by changing amongst others the degree of mineral liberation or the surface chemistry of the different minerals. This is taken into account here by shifting the bounds on constraint 12. The transformation of species by milling will be considered in the second simulation model.

The separation of the valuable element \( h \) in the feed into a concentrate and a tailings stream is a function of the fraction of species \( s \) present in the valuable element \( h \) in the tailings \( F_{hj}^s \), and the separation characteristics of species \( s \):

\[
 y_{hj} = \left( \sum_s F_{hj}^s ym_{hj}^s \right) m_{hj} \quad (13)
\]

where

\[
 F_{hj}^s = \frac{m_{hj}^s}{\left( \sum_s m_{hj}^s \right)} \quad (14)
\]

An upper and a lower bound on eq. 14, together with eq. 12 produce eq. 15:

\[
 (YM_{hj})^L m_{hj} \leq y_{hj} \leq (YM_{hj})^U m_{hj} \quad (15)
\]

where:

\[
 (YM_{hj})^L = \sum_s (ym_{hj}^s)^L (F_{hj}^s)^L
\]

\[
 (YM_{hj})^U = \sum_s (ym_{hj}^s)^U (F_{hj}^s)^U
\]

Objective function

The aim of the optimization is to maximize the recovery of valuable element \( h \) in the concentrate, subject to the constraints 2, 4 to 11, and 15. This can be stated as:
MAXh = \sum_j w_{hj} a_{hj} \tag{16}

where MAXh is the objective function which maximizes the production of valuable element h, \(w_{hj}\) = the price weight for the production of \(a_{hj}\) (e.g. monetary units/tonne), and \(j=\text{bank(s)}\) from which concentrate is vented.

**Linear Programming Model I**

Linear Programming Model I consists of eqs. 2, 4–11, 15 and 16. This model produces the optimal circuit structure, and the optimal flow rates and recovery of the valuable element h.

**MAXIMIZATION OF THE CONCENTRATE GRADE**

Linear Programming Model II simulates the flow rates of all species \(s\) of elements \(k\) in such a way that the concentrate grade is maximized. This model is subject to the optimal flow rates for valuable element \(h\) and the circuit structure produced by Linear Programming Model I, as well as the separation constraints for all species \(s\) and the transformation effects produced by the mill(s).

Fig. 2 defines the steady state flow rates of all species \(s\) in elements \(k\) between banks \(i\) and \(j\), and Fig. 3 presents the transformation of species in the mill(s). Mass-balance equations will be formulated with reference to these diagrams. (Variables denoted by an * are constants produced by Model I).

**Mass-balance equations for species \(s\)**

The general structure of the mass-balance equations for all flotation species \(s\) is very similar to that developed for the valuable element \(h\) in Model I:

\[
\sum_{j,i \neq j} t_{ij}^* m_{kj}^s + \sum_{j,i \neq j} c_{ij}^* y_{kj}^s + \sum_{j,i \neq j} M_{ij}^{sk} + \sum_{j,i \neq j} Y_{ij}^{sk} + u_{ki}^s = m_{ki}^s + y_{ki}^s
\tag{17}
\]

\[
\sum_{i,i \neq j} t_{ij}^* m_{kj}^s + \sum_{i,i \neq j} e_{ij}^* m_{kj}^s + \beta \tau_{ij}^s m_{kj}^s = m_{kj}^s
\tag{18}
\]

\[
\sum_{i,i \neq j} c_{ij}^* y_{kj}^s + \sum_{i,i \neq j} g_{ij}^s y_{kj}^s + \alpha c_{ij}^* y_{kj}^s = y_{kj}^s
\tag{19}
\]

The optimal flow rates of \(h\) produced by Model I are included as:

\[
M_{hj}^* = \sum_s m_{hj}^s
\tag{20}
\]

\[
Y_{hj}^* = \sum_s y_{hj}^s
\tag{21}
\]
Fig. 2. Steady state flow rates of flotation species \( s \) of an element \( k \) between banks \( i \) and \( j \), and through mills in streams \( g_{ij}^* \) and \( e_{ij}^* \).

\[
A_{hj}^* = \sum_s a_{hj}^* \\
B_{hj}^* = \sum_s b_{hj}^* 
\]

External constraints

The bounds on the (multiple) feed rates of the flotation species \( s \) are given by:

\[
(u_{kj}^s)_L < u_{kj}^s < (u_{kj}^s)_U 
\]
e_{ij}^{s}m_{kj}^{l} \rightarrow \text{MILL IN TAILINGS STREAM} \rightarrow m_{kj}^{s} = \sum_{l} e_{ij}^{s}m_{kj}^{l} \\
g_{ij}^{s}y_{kj}^{l} \rightarrow \text{MILL IN CONCENTRATE-STREAM} \rightarrow y_{kj}^{s} = \sum_{l} g_{ij}^{s}y_{kj}^{l} \\

Fig. 3. The transformation of flotation species \( l \) of element \( k \) into flotation species \( s \) in the mills situated in streams \( e_{ij}^{s} \) and \( g_{ij}^{s} \).

**Grade constraints**

The grade constraint for element \( k \) is defined as:
\[
\left( G_{kj}^{U} \right)^{T} \leq \frac{\sum_{s} m_{kj}^{s}}{\sum_{k} \sum_{s} m_{kj}^{s}} \leq \left( G_{kj}^{l} \right)^{U}
\]
(25)

**Flotation bank constraints**

The separation constraints for the non-valuable elements are similar to those for the valuable elements given in eq. 12, so that the following general separation constraint may be defined:
\[
(ym_{kj}^{s})^{l} m_{kj}^{s} \leq y_{kj}^{s} \leq (ym_{kj}^{s})^{U} m_{kj}^{s}
\]
(26)

**Mill constraints**

In Fig. 3 the transformation matrix for the mill is \( b_{kj}^{sl} \) in stream \( e_{ij}^{s} \) and that for the mill in stream \( g_{ij}^{s} \) is \( d_{kj}^{sl} \). Eqs. 27 and 28 define the transformation of flotation species \( l \) of element \( k \) in a mill into flotation species \( s \):
\[
M_{ij}^{sk} = \sum_{l} b_{kj}^{sl} e_{ij}^{s} m_{kj}^{l}
\]
(27)
\[
Y_{ij}^{sk} = \sum_{l} d_{kj}^{sl} g_{ij}^{s} y_{kj}^{l}
\]
(28)
These constraints may be extended into bounded constraints if the transformation matrices are not known exactly:

\[
\sum_{i} (b_{kij}^{ni}) L e_{ij}^{*} m_{kj} \leq M_{ij}^{sk} \leq \sum_{i} (b_{kij}^{ni}) U e_{ij}^{*} m_{kj} \tag{29}
\]

\[
\sum_{i} (d_{kij}^{ni}) L g_{ij}^{*} y_{kj} \leq Y_{ij}^{sk} \leq \sum_{i} (d_{kij}^{ni}) U g_{ij}^{*} y_{kj} \tag{30}
\]

**Objective function**

The objective function of Linear Programming Model II is the maximization of the grade of the valuable element \( h \), subject to eqs. 17 to 30. This is equivalent to the minimization of the recovery of the non-valuable gangue elements.

\[
\text{grade} = \sum_{j} \sum_{s} (a_{hj}^{s} - a_{zj}^{s}) \tag{31}
\]

where \( \text{grade} \) = objective function which maximizes the grade, \( a_{hj}^{s} \) = valuable species in concentrate, \( a_{zj}^{s} \) = non-valuable species in concentrate, and \( j \) = banks (s) from which concentrate is vented.

**Linear Programming Model II**

Linear Programming Model II consists of eqs. 2 and 17–31, and yields the optimal flow rates of the non-valuable elements and water, and the maximum grade of the final concentrate.

**APPLICATION ALGORITHM**

In order to design an optimal flotation circuit, Linear Programming Models I and II should be solved in succession as described below.

(a) Define bounds on the (multiple) feed rate(s) of the valuable element, and bounds on the flow rates of final concentrate and final tailings in eqs. 7–9.

(b) Establish bounds imposed on the recycle streams in eqs. 10 and 11, i.e. to exclude those recycle streams which are not practically feasible.

(c) Bounds on the separation factors for all species of the valuable element in eq. 12 must be established from fundamental models, batch flotation data, plant experience or by an appropriate knowledge-based system.

(d) Bounds on the composition of the valuable element in terms of species \( s \), i.e. the distribution of floatabilities of the valuable element should be selected for the construction of eq. 15.

(e) Any linear programming package may now be used to solve Linear Programming Model I.

(f) The structural parameters and the optimal flow rates for the valuable
element produced by Linear Programming Model I may now be used to construct eqs. 17 to 23.

(g) The feed flow rates of the flotation species \( s \) to a bank \( j \) should be bounded in eq. 24.

(h) Production requirements must be established, which could be related to the constraints on the grade of element \( k \) in the tailings streams, as defined in eq. 25.

(i) Constraints on the separation factors for all species \( s \) of all elements \( k \) in eq. 26 could be derived from batch flotation data, plant experience or an appropriate knowledge-based system.

(j) If a mill is included in the flotation circuit, eqs. 27–30 should be formulated. For this purpose a matrix for the transformation of species \( l \) to species \( s \) must be constructed from milling, liberation and flotation data measured in batch experiments, pilot plants or industrial operations. An appropriate knowledge-based system may also be used to update these bounds on the transformation matrix.

(k) A linear programming package could be used to solve Linear Programming Model II, which yields the flow rates of flotation species \( s \) that maximize the grade of the valuable element in the final concentrate.

(l) The solution of Linear Programming Model II produces a set of exact separation factors for each species \( s \) and a transformation matrix for the mills. Theoretical flotation, liberation and milling models may subsequently be used to specify the operating conditions in the flotation cells and mills which will produce an optimal recovery and grade of the valuable element.

In the sample problems described below a linear programming package FMPS on a UNIVAC 1100 computer was used to solve the models. Less than 5 s of central processor time was required in each case.

PROBLEM 1: A TWO-BANK CIRCUIT FOR THE FLOTATION OF PYRITE

A simple 2-bank rougher–cleaner circuit for the flotation of pyrite \((\text{FeS}_2)\) from a quartzitic ore will be used here to demonstrate the applicability of the simulation model to the design of flotation plants which incorporate recycle mills. The performances of circuits with and without mills will be compared. While the bounds on separation factors are hypothetical, they are similar to those for real ores \((\text{Liddell and Dunne, 1984; Hanekom, 1987})\). Element 1 (pyrite, \(\text{FeS}_2\)), element 2 (gangue) and element 3 (water) are separated into concentrate and tailings streams.

The bounds on the separation factors for the three species of the three elements are given in Tables I and II for a circuit that includes a recycle mill between the rougher and cleaner bank. In this example it is assumed that \( N_j = 5 \) and \( \tau_j = 5 \) (min) for all banks and mineral species, respectively. Although these
TABLE I

Bound on separation factors ($y_m^s_{kj}$) for rougher ($j = 1$)

<table>
<thead>
<tr>
<th>Species</th>
<th>Element:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$k = 1$</td>
</tr>
<tr>
<td>$s = 1$</td>
<td>1060-448401</td>
</tr>
<tr>
<td>$s = 2$</td>
<td>7-156</td>
</tr>
<tr>
<td>$s = 3$</td>
<td>0</td>
</tr>
</tbody>
</table>

TABLE II

Bound on separation factors ($y_m^s_{kj}$) for cleaner ($j = 2$)

<table>
<thead>
<tr>
<th>Species</th>
<th>Element:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$k = 1$</td>
</tr>
<tr>
<td>$s = 1$</td>
<td>1330-640972</td>
</tr>
<tr>
<td>$s = 2$</td>
<td>1-524</td>
</tr>
<tr>
<td>$s = 3$</td>
<td>0</td>
</tr>
</tbody>
</table>

values will differ between banks in practice, identical values are used here for simplicity. All flow rates are given in tonnes/h.

Rougher bank

By using the separation factors for the species of the valuable element, bounds may be established for constraint 15. The upper and lower bounds on $YM_{h1}$ ($h = FeS_2$) for the rougher bank are produced as follows:

$$(YM_{h1})^L = \sum_s (YM_{h1}^s)^L (F_{h1}^s)^L$$

$$= (1060 \times 0) + (7 \times 0) + (0 \times 0)$$

$$= 0$$

$$(YM_{h1})^U = \sum_s (ym_{h1}^s)^U (F_{h1}^s)^U$$

$$= (448401 \times 0.000053) + (156 \times 0.0015) + (0 \times 1)$$

$$= 24$$

The values for $(F_{h1}^s)^L$ and $(F_{h1}^s)^U$ could be obtained from experimental batch flotation data, plant data or set by the designer as a design variable. Hence:

$$0 \leq (y_{FeS_2})_1 \leq 24 (m_{FeS_2})_1$$
Cleaner bank

The same procedure is followed to obtain bounds for the separation of pyrite at the cleaner bank, i.e. \( (YM_{h2})^L \) and \( (YM_{h2})^U \):

\[
(YM_{h2})^L = \sum_s (ym_{h2}^s)^L (F_{h2}^s)^L
\]

\[
= (1330 \times 0) + (16.6 \times 0) + (0 \times 0)
\]

\[
= 0
\]

\[
(YM_{h2})^U = \sum_s (ym_{h2}^s)^U (F_{h2}^s)^U
\]

\[
= (640972 \times 0.00001) + (524 \times 0.0049) + (0 \times 1)
\]

\[
= 9
\]

Hence:

\[
0 \leq (y_{FeS2})_2 \leq 9 (m_{FeS2})_2
\]

In this example milling increases the mineral liberation and the flotation rates in the cleaner bank. If a mill is not included in the circuit, the flotation rates and hence the separation factor will be lower. The bounds \( (YM_{h2})^L \) and \( (YM_{h2})^U \) for the cleaner bank in a circuit that does not include a mill are:

\[
(YM_{h2})^L = \sum_s (ym_{h2}^s)^L (F_{h2}^s)^L
\]

\[
= (1191 \times 0) + (10.6 \times 0) + (0 \times 0)
\]

\[
= 0
\]

\[
(YM_{h2})^U = \sum_s (ym_{h2}^s)^U (F_{h2}^s)^U
\]

\[
= (573824 \times 0.00001) + (243 \times 0.001) + (0 \times 0)
\]

\[
= 6
\]

Hence:

\[
0 \leq (y_{FeS2})_2 \leq 6 (m_{FeS2})_2
\]

From the above data the following simulation models can be constructed for a two-bank flotation plant.

**Linear Programming Model I**

**Objective function**

\[\text{MAX}h = 200 \ (a_{FeS2})_2\]
Mass-balance constraints

\[
(r_m^{FeS})_{12} + (r_y^{FeS})_{12} + (u_{FeS})_{1} = (m_{FeS})_{1} + (y_{FeS})_{1}
\]

\[
(r_m^{FeS})_{21} + (r_y^{FeS})_{21} + (u_{FeS})_{2} = (m_{FeS})_{2} + (y_{FeS})_{2}
\]

\[
(r_m^{FeS})_{12} + (b_{FeS})_{2} = (m_{FeS})_{2}
\]

\[
(r_m^{FeS})_{21} + (b_{FeS})_{1} = (m_{FeS})_{1}
\]

\[
(r_y^{FeS})_{12} + (a_{FeS})_{2} = (y_{FeS})_{2}
\]

\[
(r_y^{FeS})_{21} + (a_{FeS})_{1} = (y_{FeS})_{1}
\]

Feed constraints (fixed)

\[
(u_{FeS})_{1} = 8.4
\]

\[
(u_{FeS})_{2} = 0
\]

Concentrate constraints

\[
(a_{FeS})_{1} = 0
\]

\[
0 \leq (a_{FeS})_{2} \leq 20
\]

Tailings constraints

\[
0 \leq (b_{FeS})_{1} \leq 20
\]

\[
(b_{FeS})_{2} = 0
\]

Recycle constraint

\[
(r_m^{FeS})_{21} = 0
\]

Separation constraints for valuable element (mill included)

\[
0 \leq (y_{FeS})_{1} \leq 24 (m_{FeS})_{1}
\]

\[
0 \leq (y_{FeS})_{2} \leq 9 (m_{FeS})_{2}
\]

Separation constraints for valuable element (no mill included)

\[
0 \leq (y_{FeS})_{1} \leq 24 (m_{FeS})_{1}
\]

\[
0 \leq (y_{FeS})_{2} \leq 6 (m_{FeS})_{2}
\]
Linear Programming Model II

The structural parameters and the flow rates of the valuable element $h$ (FeS$_2$), which are produced by Linear Programming Model I, are shown in Fig. 4 and are also included in this model. The structural parameters are:

$t_{12}^* = 1$  
$c_{12}^* = 0$  
$a_{C1}^* = 0$  
$b_{T1}^* = 1$  

The elements $k$ in this model are FeS$_2$ (valuable element), gangue (G) and water (W).

Objective function

$\text{grade} = \sum_k (a_{FeS_2}^k)^2 - \sum_k a_{G2}^k - \sum_k a_{W2}^k$

Fig. 4. Circuit structure and flow rates of elements for Problem 1.
Objective function

\[
\text{grade} = \sum_s (a_{\text{FeS}_2}^s)_2 \left[ \sum_s a_{\overline{G}}^s - \sum_s a_{\overline{W}}^s \right]
\]

Mass-balance constraints for species \( s \)

\[
t_{21}^s m_{k1}^s + Y_{21}^s u_{k2}^s = m_{k2}^s + y_{k2}^s
\]

\[
t_{12}^s m_{k2}^s + c_{12}^s y_{k2}^s + u_{k1}^s = m_{k1}^s + y_{k1}^s
\]

\[
t_{21}^s b_{T1}^s m_{k1}^s = m_{k1}^s
\]

\[
t_{12}^s b_{T2}^s m_{k2}^s = m_{k2}^s
\]

\[
Y_{21}^s + a_{C1}^s y_{k1}^s = y_{k1}^s
\]

\[
c_{12}^s y_{k2}^s + a_{C2}^s y_{k2}^s = y_{k2}^s
\]

Flow rates of valuable element (FeS\(_2\)) produced by Model I

\[
8.92 = \sum_s (y_{\text{FeS}_2}^s)_1 \quad 8.028 = \sum_s (y_{\text{FeS}_2}^s)_2
\]

\[
0.372 = \sum_s (m_{\text{FeS}_2}^s)_1 \quad 0.892 = \sum_s (m_{\text{FeS}_2}^s)_2
\]

\[
0.372 = \sum_s (b_{\text{FeS}_2}^s)_1 \quad 8.028 = \sum_s (a_{\text{FeS}_2}^s)_2
\]

Feed rates for species \( s \)

FeS\(_2\):

\[
(u_{\text{FeS}_2}^1)_1 = 5.46
\]

\[
(u_{\text{FeS}_2}^2)_1 = 2.772
\]

\[
(u_{\text{FeS}_2}^3)_1 = 0.168
\]

Gangue (G):

\[
u_{G1}^1 = 6.124
\]

\[
u_{G1}^2 = 2.624
\]

\[
u_{G1}^3 = 282.852
\]

Water (W):

\[
u_{W1}^1 = 370
\]

Separation factors of species \( s \)

These values are given in Tables I and II.
TABLE III

Flotation rates \( (k_{k_j} \text{ min}^{-1}) \) in rougher \((j = 1)\)

<table>
<thead>
<tr>
<th>Species</th>
<th>Element: (k = 1)</th>
<th>(k = 2)</th>
<th>(k = 3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(s = 1)</td>
<td>2.5</td>
<td>0.085</td>
<td>0.002</td>
</tr>
<tr>
<td>(s = 2)</td>
<td>0.156</td>
<td>0.014</td>
<td>0</td>
</tr>
<tr>
<td>(s = 3)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

TABLE IV

Flotation rates \( (k_{k_j} \text{ min}^{-1}) \) in cleaner \((j = 2)\)

<table>
<thead>
<tr>
<th>Section</th>
<th>Element: (k = 1)</th>
<th>(k = 2)</th>
<th>(k = 3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(s = 1)</td>
<td>2.7</td>
<td>0.061</td>
<td>0.01</td>
</tr>
<tr>
<td>(s = 2)</td>
<td>0.062</td>
<td>0.032</td>
<td>0</td>
</tr>
<tr>
<td>(s = 3)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Grade constraint**

\[
\frac{\sum_{s} m_{s}^{*}}{\sum_{k_s} m_{k_s}^{*}} > 0.3
\]

This constraint ensures that the fraction of water in the final concentrate is larger than 0.3.

**Transformation matrix for gangue material in the mill**

The transformation of species within the valuable element has been taken into consideration in the first simulation model. Hence, the transformation matrix of only the gangue element is considered in this model:

\[
\begin{bmatrix}
1 & 2 & 3 \\
1 & 0.5 & 0 & 0 \\
2 & 0 & 0.5 & 0 \\
3 & 0.5 & 0.5 & 1
\end{bmatrix}
\]

This matrix implies that 0.5 of the fast floating species of the gangue has been transformed to the non-floating species. The same holds for the slow floating fraction.
Final results

The operating conditions could now be estimated by applying eq. 3 to the optimal separation factors for the different species of the three elements. Tables III and IV summarize the optimal rate constants, from which operating conditions could be estimated by use of phenomenological models (Ross, 1988). The flow rates of the elements produced by Model II are given in Fig. 4, which shows a practically feasible two-bank circuit configuration.

A sulphur grade of 32.2% and a recovery of sulphur of 95.4% in the concentrate are produced by the circuit without a regrind mill. These values could be improved to 41.3% and 95.6%, respectively, by the inclusion of a mill in the concentrate stream from the rougher to the cleaner. These values are typical of those attained in practice.

PROBLEM 2: A FIVE-BANK CIRCUIT FOR THE FLOTATION OF CASSITERITE

The valuable element concentrated in this circuit is cassiterite (SnO₂), with gangue (G) and water (W) constituting the non-valuable elements. Five flotation banks (i.e. a rougher, scavenger, cleaner, recleaner I and recleaner II) will be used here to demonstrate the applicability of the optimization procedure. Again, the effect of a regrind mill will be investigated. Hypothetical bounds on the separation factors are similar to data given by Sutherland (1981) and Frew and Trahar (1982). Whereas the use of Models I and II was explained in more detail under Problem 1, only the more essential information will be summarized here.

The bounds on the separation factors are summarized in Tables V and VI. It is assumed here that \( N_j = 8 \) and \( \tau_j = 4 \) (min) for both the rougher and scavenger banks, whereas \( N_j = 8 \) and \( \tau_j = 2 \) (min) apply to the cleaner banks. A mill is assumed to be included in the concentrate stream from the rougher bank to the recleaner I bank.

From these data the following constraints could be formulated for the valuable element cassiterite (SnO₂) for each flotation bank:

**TABLE V**

Bounds on separation factors \( (y m_{k,s}^*) \) in both the rougher \((j = 1)\) and scavenger \((j = 2)\)

<table>
<thead>
<tr>
<th>Species</th>
<th>Element: ( k = 1 )</th>
<th>( k = 2 )</th>
<th>( k = 3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( s = 1 )</td>
<td>38.3-5764800</td>
<td>12.8-200475</td>
<td>0.097-0.85</td>
</tr>
<tr>
<td>( s = 2 )</td>
<td>1.0-6560</td>
<td>1.7-6560</td>
<td>0</td>
</tr>
<tr>
<td>( s = 3 )</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
TABLE VI

Bounds on separation factors \( (y_m \kappa_j) \) in cleaners \((j=3 \text{ to } 5)\)

<table>
<thead>
<tr>
<th>species</th>
<th>Element:</th>
<th>(k=1)</th>
<th>(k=2)</th>
<th>(k=3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(s=1)</td>
<td></td>
<td>4.8-6560</td>
<td>0.33-7.2</td>
<td>0.024-0.19</td>
</tr>
<tr>
<td>(s=2)</td>
<td></td>
<td>0.1-8.2</td>
<td>0.23-2.76</td>
<td>0</td>
</tr>
<tr>
<td>(s=3)</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Fig. 5. Circuit structure and flow rates of elements for Problem 2.
0 \leq (y_{SnO_2})_1 \leq 11.00 \ (m_{SnO_2})_1
0 \leq (y_{SnO_2})_2 \leq 7.00 \ (m_{SnO_2})_2
0 \leq (y_{SnO_2})_3 \leq 0.70 \ (m_{SnO_2})_3
0 \leq (y_{SnO_2})_4 \leq 0.60 \ (m_{SnO_2})_4
0 \leq (y_{SnO_2})_5 \leq 0.45 \ (m_{SnO_2})_5

Without a mill being included in the concentrate stream from the rougher to recleaner I, the separation factor for recleaner I (bank 4) is:

0 \leq (y_{SnO_2})_4 \leq 0.35 \ (m_{SnO_2})_4

The transformation matrix for the gangue material is:

\[
\begin{bmatrix}
1 & 2 & 3 \\
1 & 0.5 & 0 & 0 \\
2 & 0 & 0.5 & 0 \\
3 & 0.5 & 0.5 & 1
\end{bmatrix}
\]

The circuit structure and the flow rates of the elements for the case where a mill is included, are depicted in Fig. 5. Operating conditions were estimated by the application of eq. 3, and are summarized in Tables VII to XI.

The simulation produces an unusual circuit configuration for a flotation circuit comprising of five flotation banks, although there are certain distinct sim-

**TABLE VII**

Flotation rates \((k^*_s \text{ min}^{-1})\) in rougher \((j=1)\)

<table>
<thead>
<tr>
<th>Species</th>
<th>Element: (k=1)</th>
<th>(k=2)</th>
<th>(k=3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>s = 1</td>
<td>0.146</td>
<td>0.09</td>
<td>0.003</td>
</tr>
<tr>
<td>s = 2</td>
<td>0.091</td>
<td>0.05</td>
<td>0</td>
</tr>
<tr>
<td>s = 3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**TABLE VIII**

Flotation rates \((k^*_s \text{ min}^{-1})\) in scavenger \((j=2)\)

<table>
<thead>
<tr>
<th>Species</th>
<th>Element: (k=1)</th>
<th>(k=2)</th>
<th>(k=3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>s = 1</td>
<td>0.146</td>
<td>0.097</td>
<td>0.017</td>
</tr>
<tr>
<td>s = 2</td>
<td>0.095</td>
<td>0.033</td>
<td>0</td>
</tr>
<tr>
<td>s = 3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
TABLE IX

Flotation rates \( (k_{ij} \text{ min}^{-1}) \) in cleaner \((j=3)\)

<table>
<thead>
<tr>
<th>Species</th>
<th>Element:</th>
<th>( k=1 )</th>
<th>( k=2 )</th>
<th>( k=3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( s=1 )</td>
<td>1.0</td>
<td>0.018</td>
<td>0.001</td>
<td></td>
</tr>
<tr>
<td>( s=2 )</td>
<td>0.033</td>
<td>0.013</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>( s=3 )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

TABLE X

Flotation rates \( (k_{ij} \text{ min}^{-1}) \) in recleaner I \((j=4)\)

<table>
<thead>
<tr>
<th>Species</th>
<th>Element:</th>
<th>( k=1 )</th>
<th>( k=2 )</th>
<th>( k=3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( s=1 )</td>
<td>0.123</td>
<td>0.018</td>
<td>0.011</td>
<td></td>
</tr>
<tr>
<td>( s=2 )</td>
<td>0.028</td>
<td>0.013</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>( s=3 )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

TABLE XI

Flotation rates \( (k_{ij} \text{ min}^{-1}) \) in recleaner II \((j=5)\)

<table>
<thead>
<tr>
<th>Species</th>
<th>Element:</th>
<th>( k=1 )</th>
<th>( k=2 )</th>
<th>( k=3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( s=1 )</td>
<td>1.0</td>
<td>0.018</td>
<td>0.011</td>
<td></td>
</tr>
<tr>
<td>( s=2 )</td>
<td>0.16</td>
<td>0.013</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>( s=3 )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Similarities to existing industrial flotation plants. For example, the tailings are recycled to previous banks. The circuit which does not include a mill, produces an Sn grade of 51.1% and an Sn recovery of 44.0%. When a recycle mill is included in the circuit as depicted in Fig. 5, the Sn grade in the final concentrate increases to 67.3% and the Sn recovery to 75.4%.

PROBLEM 3: A FOUR-BANK CIRCUIT FOR THE FLOTATION OF GALENA

As in Problem 2, the models are not presented in detail, and only the more important data are given here. A fixed circuit structure is selected and the influence of two regrind mills on optimal flotation performance will be determined. The final circuit incorporation the mills is shown in Fig. 6. Mills are placed in the tailings recycle between the cleaner and rougher, and in the concentrate stream between the rougher and the recleaner.
Bounds on the separation factors for the flotation of galena (PbS) from gangue (G) material were chosen so as to approximate the flotation data given by Forssberg et al. (1982) and Sutherland (1977). These bounds are given in Tables XII to XIV. In this problem it was assumed that $N_j = 8$ and $\tau_j = 2$ (min) for the rougher and cleaners, and $N_j = 10$ and $\tau_j = 2$ (min) for the scavenger.

By the application of these data the following constraints could be formulated for the valuable element galena (PbS):

$$0 \leq (y_{PbS})_1 \leq 2.1 \ (m_{PbS})_1$$
TABLE XII

Bounds on separation factors \((y m^*_{ij})\) in rougher \((j = 1)\)

<table>
<thead>
<tr>
<th>Species</th>
<th>Element:</th>
<th>(k = 1)</th>
<th>(k = 2)</th>
<th>(k = 3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(s = 1)</td>
<td>2.84-63</td>
<td>0.46-6.53</td>
<td>0.024-0.087</td>
<td></td>
</tr>
<tr>
<td>(s = 2)</td>
<td>0.02-3.0</td>
<td>0.024-0.087</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>(s = 3)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

TABLE XIII

Bounds on separation factors \((y m^*_{ij})\) in scavenger \((j = 2)\)

<table>
<thead>
<tr>
<th>Species</th>
<th>Element:</th>
<th>(k = 1)</th>
<th>(k = 2)</th>
<th>(k = 3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(s = 1)</td>
<td>3.3-109</td>
<td>0.48-2.7</td>
<td>0.14-0.48</td>
<td></td>
</tr>
<tr>
<td>(s = 2)</td>
<td>0.27-1.59</td>
<td>0.14-0.48</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>(s = 3)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

TABLE XIV

Bounds on separation factors \((y m^*_{ij})\) in cleaner \((j = 3)\) and reclaimer \((j = 4)\)

<table>
<thead>
<tr>
<th>Species</th>
<th>Element:</th>
<th>(k = 1)</th>
<th>(k = 2)</th>
<th>(k = 3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(s = 1)</td>
<td>4.1-112.4</td>
<td>1.07-15.8</td>
<td>0.04-0.48</td>
<td></td>
</tr>
<tr>
<td>(s = 2)</td>
<td>0.082-2.0</td>
<td>0.082-1.19</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>(s = 3)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

0 \(\leq (y_{PbS})_2 \leq 0.7 \ (m_{PbS})_2\)
0 \(\leq (y_{PbS})_3 \leq 0.8 \ (m_{PbS})_3\)
0 \(\leq (y_{PbS})_4 \leq 2.1 \ (m_{PbS})_4\)

Without the two mills in the circuit, as shown in Fig. 6, the separation factors for the valuable element galena in the rougher and reclaimer banks change to:

0 \(\leq (y_{PbS})_1 \leq 0.6 \ (m_{PbS})_1\)
0 \(\leq (y_{PbS})_4 \leq 0.8 \ (m_{PbS})_4\)

The transformation matrix for the gangue material in both mills is taken to be:
The flow rates of the three elements for this circuit are depicted in Fig. 6. The operating conditions were estimated by the application of eq. 3, with the results being summarized in Tables XV to XVIII.

The simulation produces a typical sulphide flotation circuit consisting of four banks. It is clear that the two mills have a marked effect on the performance of this circuit. The recovery of the valuable element Pb in this circuit is 95.6%, while the grade produced is 68.8% Pb. Without these mills included in the circuit, the grade and recovery drop to 30.2 and 87.7% Pb, respectively.

**TABLE XV**

*flotation rates ($k_{ij}$ min$^{-1}$) in rougher ($j=1$)*

<table>
<thead>
<tr>
<th>Species</th>
<th>Element: k=1</th>
<th>k=2</th>
<th>k=3</th>
</tr>
</thead>
<tbody>
<tr>
<td>s=1</td>
<td>0.5</td>
<td>0.033</td>
<td>0.002</td>
</tr>
<tr>
<td>s=2</td>
<td>0.091</td>
<td>0.018</td>
<td>0</td>
</tr>
<tr>
<td>s=3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**TABLE XVI**

*Flotation rates ($k_{ij}$ min$^{-1}$) in scavenger ($j=2$)*

<table>
<thead>
<tr>
<th>Species</th>
<th>Element: k=1</th>
<th>k=2</th>
<th>k=3</th>
</tr>
</thead>
<tbody>
<tr>
<td>s=1</td>
<td>0.1</td>
<td>0.025</td>
<td>0.025</td>
</tr>
<tr>
<td>s=2</td>
<td>0.041</td>
<td>0.008</td>
<td>0</td>
</tr>
<tr>
<td>s=3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**TABLE XVII**

*Flotation rates ($k_{ij}$ min$^{-1}$) in cleaner ($j=3$)*

<table>
<thead>
<tr>
<th>Species</th>
<th>Element: k=1</th>
<th>k=2</th>
<th>k=3</th>
</tr>
</thead>
<tbody>
<tr>
<td>s=1</td>
<td>0.156</td>
<td>0.064</td>
<td>0.034</td>
</tr>
<tr>
<td>s=2</td>
<td>0.051</td>
<td>0.007</td>
<td>0</td>
</tr>
<tr>
<td>s=3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
TABLE XVIII

Flotation rates \( (k^l_j \text{ min}^{-1}) \) in recleaner \( (j = 4) \)

<table>
<thead>
<tr>
<th>Species</th>
<th>Element:</th>
<th>( k=1 )</th>
<th>( k=2 )</th>
<th>( k=3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( s=1 )</td>
<td>0.6</td>
<td>0.064</td>
<td>0.02</td>
<td></td>
</tr>
<tr>
<td>( s=2 )</td>
<td>0.079</td>
<td>0.007</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>( s=3 )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

CONCLUSIONS AND SIGNIFICANCE

The solutions to the sample problems serve to demonstrate a number of properties of the simulation model, as discussed below.

(a) The circuit structures produced by Linear Programming Model I are similar to those encountered in industry. Furthermore, Linear Programming Model II produces grades and recoveries which compare favourably with those of similar flotation plants in industry. Hence, the two-step approach for the optimization of a non-linear simulation problem produces meaningful results. It was shown that the effect of regrind mills on the recovery and grade could be simulated realistically. Although not demonstrated by the sample problems, the simulation model could also consider a multiple feed strategy as described by Williams et al. (1986).

(b) The simulation model produces solutions within a few seconds, which makes it more time-efficient than iterative numerical procedures. This feature makes the model particularly attractive for inclusion in a control algorithm.

(c) Most flotation models yield only approximate predictions of recoveries and grades. Since the separation characteristics are included here as constraints rather than a flotation model, the inaccuracies in any flotation model may be introduced as an upper and a lower bound. This implies that only approximate and not precise separation data are required by the simulation model. The construction of bounds for the separation characteristics is meaningful, since the operating conditions in a flotation plant may vary greatly and hence, may not be defined accurately by a theoretical or an empirical separation model. Therefore, the bounding of the separation characteristics as is done in this simulation model also embraces the dynamics encountered in a flotation circuit.

(d) The data for the estimation of bound may be obtained from a data base, which may be updated by the user or by the simulation model in a control loop. The inclusion of operating conditions as bounds makes this model particularly attractive for use in a knowledge-based system, which could manipulate the available data base to estimate bounds.

(e) The only data required by the simulation models are the separation factors for the fast and slow floating species of a particular element \( k \), and the
feed composition of element $k$ in terms of the fast floating, slow floating and non-floating species.

Once a flotation plant has been designed and commissioned, the model presented here may be used to simulate and, if used in a control algorithm, control the operation of the plant. If, for example, there are drastic changes in the ore body and subsequently in the flotation conditions, this model may be used to suggest an alternative circuit and new operating conditions.

APPENDIX - LIST OF SYMBOLS

- $a_{hij}$ = concentrate recovery of valuable element $h$ from flotation bank $j$
- $a_{sij}$ = concentrate recovery of species $s$ of valuable element $h$ from flotation bank $j$
- $a_{zij}$ = concentrate recovery of species $s$ of non-valuable elements $z$ from flotation bank $j$
- $a_{l}^{L}$ = lower bound on $a_{hij}$
- $a_{hij}^{U}$ = upper bound on $a_{hij}$
- $A_{hij}$ = concentrate flow of valuable element $h$ from flotation bank $j$; optimal value produced by Linear Programming Model I.
- $b_{hij}$ = tailings recovery of valuable element $h$ from flotation bank $j$
- $b_{sij}$ = tailings recovery of species $s$ of valuable element $h$ from flotation bank $j$
- $b_{l}^{L}$ = lower bound on $b_{hij}$
- $b_{hij}^{U}$ = upper bound on $b_{hij}$
- $B_{sij}$ = tailings flow of valuable element $h$ from flotation bank $j$; optimal value produced by Linear Programming Model I.
- $b_{kij}^{a}$ = breakage of species $s$ of element $k$ into species $l$ by use of a mill in tailings recycle $ij$
- $(b_{kij}^{a})^{L}$ = lower bound on $b_{kij}^{a}$
- $(b_{kij}^{a})^{U}$ = upper bound on $b_{kij}^{a}$
- $c_{ij}$ = fraction of $y_{hij}$ that is recycled from flotation bank $j$ to $i$; optimal value produced by Linear Programming Model I.
- $d_{kij}^{a}$ = breakage of species $s$ of element $k$ into species $l$ by use of a mill in the concentrate recycle $ij$
- $(d_{kij}^{a})^{L}$ = lower bound on $d_{kij}^{a}$
- $(d_{kij}^{a})^{U}$ = upper bound on $d_{kij}^{a}$
- $e_{sij}$ = tailings stream that includes a mill; optimal value produced by Linear Programming Model I.
- $F_{hij}$ = $m_{hij} + \sum_{s}m_{sij}$; fraction of species $s$ in element $k$ in the tailings
- $G_{kij}$ = grade of element $k$ in tailings stream $j$
- $(G_{kij})^{L}$ = lower bound on tailings grade
- $(G_{kij})^{U}$ = upper bound on tailings grade
- $g_{sij}$ = concentrate stream that includes a mill; optimal value produced by Linear Programming Model I.
- Grade = objective function that maximizes the grade
- $k_{f}$ = rate constant of the fast floating species
- $k_{s}$ = rate constant of the slow floating species
- $k_{sij}$ = flotation rate of the species $s$ of element $k$ in flotation bank $j$
LINEAR PROGRAMMING IN THE DESIGN OF FLOTATION CIRCUITS  

$m_{bj}$ = tailings flow of valuable element $h$ from bank $j$
$m_{kJ}$ = tailings flow of species $s$ in element $k$ from bank $j$
$M_{bj}$ = tailings flow of the valuable element $h$ from flotation bank $j$; optimal value produced by Linear Programming Model I
$M_{kj}^{u}$ = $\sum_{i} d_{ki}^{u} e_{i}^{u} m_{ji}^{l}$: flow rate of species $s$ of element $k$ from a mill situated in the tailings recycle stream $ij$

$MAXh$ = total earnings due to the optimal recovery of valuable element $h$
$N_j$ = number of cells in flotation bank $j$
$R$ = recovery in batch flotation
$rm_{hj}^{h}$ = recycle of valuable element $h$ in the tailings from flotation bank $j$ to $i$
$ry_{hj}^{h}$ = recycle of valuable element $h$ in the concentrate from flotation bank $j$ to $i$
$t$ = flotation time
$\lambda_{ij}^{u}$ = fraction of $m_{bj}$ that is recycled from flotation bank $j$ to $i$; optimal value produced by Linear Programming Model I
$Y_{bj}$ = concentrate flow of valuable element $h$ from flotation bank $j$; optimal value produced by Linear Programming Model I
$y_{hj}^{k}$ = concentrate flow of valuable element $h$ from flotation bank $j$
$y_{hj}^{s}$ = concentrate flow of species $s$ in element $k$ from flotation bank $j$

$ym_{hj}^{s}$ = separation factor; $y_{hj}^{s} / m_{hj}^{u}$
$(ym_{hj}^{s})^{l}$ = lower bound on $ym_{hj}^{s}$
$(ym_{hj}^{s})^{u}$ = upper bound on $ym_{hj}^{s}$
$YM_{bj}$ = $\sum_{s} F_{hj}^{s} y_{hj}^{s}$
$(YM_{bj})^{l}$ = lower bound on ($\sum_{s} F_{hj}^{s} y_{hj}^{s}$)
$(YM_{bj})^{u}$ = upper bound on ($\sum_{s} F_{hj}^{s} y_{hj}^{s}$)

$Y_{hj}^{ak}$ = $\sum_{i} d_{ki}^{u} g_{ij}^{u} y_{hj}^{l}$: flow rate of species $k$ from a mill situated in concentrate recycle stream $ij$

$u_{hj}$ = feed of valuable element $h$ to flotation bank $i$
$u_{kj}^{l}$ = feed of species $s$ of element $k$ to flotation bank $j$
$u_{hj}^{u}$ = lower bound on $u_{hj}$
$u_{hj}^{U}$ = upper bound on $u_{hj}$
$(u_{hj}^{l})^{l}$ = lower bound on $u_{hj}^{l}$
$(u_{hj}^{l})^{u}$ = upper bound on $u_{hj}^{l}$

$w_{hj}$ = price weight of valuable element $h$ in the objective function $MAXh$

Greek letters

$\alpha_{hj}^{u}$ = fraction of $y_{hj}$ that is concentrate product; optimal value produced by Linear Programming Model I.
$\beta_{hj}^{u}$ = fraction of $m_{hj}$ that is tailings product; optimal value produced by Linear Programming Model I
$\Phi$ = ultimate recovery at infinite time
$\Omega$ = fraction of the ultimately floatable species that is slow floating
$\tau_{hj}$ = retention time of all species $s$ in bank $j$
REFERENCES


Johnson, N.W., 1987. Application of kinetics and liberation data to an industrial flotation process. CIM Bull., 80 (899): 113


STEADY-STATE SIMULATION AND OPTIMIZATION OF GRAVITY SEPARATION CIRCUITS BY USE OF LINEAR PROGRAMMING AND EXPERT SYSTEMS

K.R. ANTHONY, J.S.J. VAN DEVENTER & M.A. REUTER§

Department of Metallurgical Engineering, University of Stellenbosch, Stellenbosch, 7600 South Africa
§ Present address: Institut für Metallhüttenwesen und Elektrometallurgie, R.-W. Technische Hochschule, 5100 Aachen, Germany

ABSTRACT

A knowledge-based system for the steady-state simulation and optimization of gravity separation circuits is presented. The only data required by the simulation models are the number of separation banks, number of species, feed rates, composition of feed and the separation factors of the different species. A knowledge-base which permits the intelligent choice of bounds, is used to generate the bounds on the separation factors. The knowledge-base may be updated by the user or the simulation model itself. In this paper, a gravity concentration plant is described by two linear models, one being the subset of the other. The computer program formulates and solves these linear models sequentially by using any suitable linear programming (LP) package. The first model optimizes the flow rates and the path of the valuable elements between the separator banks. This model produces the optimal flow rates of the valuable elements between the banks, as well as the circuit configuration. The second model simulates the flow rates of all elements in such a way that the concentrate grade is maximized. An optimal grade for the valuable elements in the concentrate is given by the second model. The circuit configurations, grades and recoveries produced by the simulation model are similar to those encountered in industry. This model may also be used to suggest all alternative circuit structure and operating conditions in the case of drastic changes in the operating environment of the plant. The simulation model is very time efficient and produces solutions within minutes.

Keywords
Expert systems; gravity concentration; knowledge-base; linear programming

INTRODUCTION

The use of computers in the mineral processing industry must be increased in order to enhance plant performance. Data base modelling has been part of the minerals industry for some time, with micro-computers being used for process modelling, flowsheet design and optimization, material balancing and plant simulation. The concepts of knowledge-based simulation, expert systems and artificial intelligence, however, are relatively new to most engineering disciplines.

Gravity concentration is not understood sufficiently well, so that fundamental models are not capable of predicting plant behaviour at different operating conditions. If totally empirical models are formulated, it is difficult to decide on model forms that will be valid for a wide variety of conditions. Since gravity concentration is very difficult to model on a theoretical basis, expert systems, which utilize and infer knowledge from a data bank, could supplement or replace theoretical modelling. This wide-ranging heuristic information, which is particularly abundant within the mineral processing industry, can be
incorporated easily into a knowledge-based system. Furthermore, rules of thumb, equipment malfunction or deterioration, and operator experience (especially where there are little known data) can be taken into account in expert system simulation.

When designing mineral separation plants, it is attempted to produce an optimal configuration as well as optimal operating conditions. Reuter et al. (1988, 1990) optimized mineral separation circuits by solving two sequential linear models using linear programming. The mass balance equations which define the flow rates for a system of k mineral elements in a gravity concentrator plant are highly non-linear (Green et al., 1984), and must be linearised in order to apply linear programming methods. Reuter and Van Deventer (1990) accomplished this by rewriting the equations as two sets of linear equations. These two sets form the basis of two linear simulation models, the second set being a subset of the first. In this paper, the methodology proposed by Reuter and Van Deventer (1990) will be applied to gravity separation circuits.

The first linear simulation model produces an optimal configuration and optimal flow rates for the valuable elements between the banks of separators. This solution is a function of the mineral separation in each bank, the plant size and constraints on the grades. The second linear simulation model produces the flow rates of the non-valuable mineral elements in such a way that the concentrate is maximized. This model is subject to the optimal flow rates for the valuable elements and the circuit structure produced by the first model. This second model is subject to the same separation characteristics of the separator and the same constraints on the grades as the first linear simulation model.

This simulator includes the separation characteristics of a mineral element at a particular bank as bounds and not as a model included in the mass balance equations. This approach is realistic if it is considered that most simulation models for mineral separators are not fundamental, but empirical and suitable for a particular application only. Since the separation constraints may be established either by the application of theoretical models, separation curves, empirical models or from plant experience, a knowledge data base can be used to generate the bounds on the separation factors. With the use of an extensive data base, the effect of ore type, mineralogy of the ore and the type of equipment used, can be included in determining the separation of the mineral element.

The objective of this paper is to formulate the structure of the proposed expert system and knowledge data base, and to demonstrate its application in optimizing a gravity concentration circuit. Both the recovery and grade of the valuable mineral elements in the concentrate will be optimized by the use of the two sequential linear models. The model will be applied to three sample problems taken from the literature. A sensitivity analysis will also be carried out in order to determine the effect of structural changes to the circuit on the recovery and grade of the valuable minerals.

STRUCTURE OF KNOWLEDGE-BASED SYSTEM

An expert system is a computer program that behaves like an expert in some, usually narrow, domain of application. The main feature of these systems is the separation of knowledge about the subject from the techniques that are used to think about this knowledge. Because expert systems have to be capable of solving problems that require expert knowledge in a particular domain, they should possess that knowledge in some form. Therefore, they are also called knowledge-based systems. An expert system should also be capable of explaining its behaviour and its decisions to the user. An additional feature that is often required is the ability to deal with uncertainty and incompleteness. Two functions are needed to build an expert system. The problem-solving function must be capable of using the domain-specific knowledge - this may require dealing with uncertainty. The user-interaction function includes the explanation of the system's decisions during and after the problem-solving process (Bratko, 1986).

An expert system can be divided into three main modules: a knowledge base, an inference engine and a user interface.
A knowledge-base comprises the domain knowledge, including simple facts about the domain, rules that describe relations or phenomena in the domain, and possibly also methods, heuristics and ideas for solving problems in this domain. An inference engine contains the techniques and procedures that drive the thought-process. A user interface caters for smooth communication between the user and the system, also providing the user with an insight into the problem-solving process carried out by the inference engine. The inference engine and the user interface can be conveniently viewed as one module, usually called an expert system shell, or simply a shell for brevity.

The foregoing scheme separates knowledge from algorithms that use the knowledge. This division is suitable for the following reason: the knowledge base depends on the application, while the shell is, in principle at least, domain independent. This is a rational way to develop a shell that can be used universally, and then to plug in a new knowledge base for each application.

The development of an expert system can be divided into the following (Bratko, 1986):
1. Select a method for representing and organizing knowledge.
2. Design an inference mechanism that corresponds to this formalism of knowledge.
3. Add user-interaction facilities.
4. Add a facility for handling uncertainty.

Knowledge organization

Conditional statements, also known as if-then rules, will be used to represent knowledge and organize data. Elements of knowledge are contained in rule premises. When these elements become true, the system can "infer" about the truth of other elements within the rule conclusion. As conclusions are made, the system moves to other rules containing these "inferred" elements in the premises. In this way, the search process is driven, not by referencing the linked rules, but rather by the data themselves, which act as connections between rules. Such data are referred to as structural knowledge.

Knowledge located in data base

The first step in developing a knowledge base system is to start the process of knowledge acquisition. The domain knowledge for a gravity separation plant could consist of the following, for example:

<table>
<thead>
<tr>
<th>Ore type</th>
<th>constituents and mineralogy</th>
</tr>
</thead>
<tbody>
<tr>
<td>relative density</td>
<td></td>
</tr>
<tr>
<td>particle shape</td>
<td></td>
</tr>
<tr>
<td>particle size</td>
<td></td>
</tr>
<tr>
<td>water-solids ratio</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Equipment</th>
<th>type and design</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of banks</td>
<td></td>
</tr>
<tr>
<td>size</td>
<td></td>
</tr>
<tr>
<td>arrangement</td>
<td></td>
</tr>
<tr>
<td>malfunction or deterioration</td>
<td></td>
</tr>
<tr>
<td>effect of regrind mills</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Operating conditions</th>
<th>tables (effect of slope and stroke; flow rate and distribution of water)</th>
</tr>
</thead>
<tbody>
<tr>
<td>spirals (effect of water-solids ratio; setting of splitters)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Circuit constraints</th>
<th>Pipe diameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pumping speed limitations</td>
<td></td>
</tr>
<tr>
<td>Excluding certain recycle streams that are not permissible.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>External factors</th>
<th>Production grade requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Production limitations</td>
<td></td>
</tr>
</tbody>
</table>
| Price weights of concentrate or mineral element | }

These elements are transformed into sentences or statements that are incorporated into rules.
Expert program structure

The expert program structure is the shell which consists of the inference engine and the user interface. The user must input certain information about the gravity concentration plant, such as the number of separation banks, equipment used, ore type, ore mineralogy and feed rate of the different mineral elements. The inference engine interacts dynamically with the knowledge-base to determine the bounds on the separation constraints and certain streams, the exclusion of certain recycle streams and the price weights of concentrate and certain mineral elements. The user can optionally change the parameters obtained from the knowledge-base. The first linear simulation model is formulated by the system and solved by using any suitable linear programming (LP) package. This model produces an optimal configuration and optimal flow rates of the valuable elements between the banks of separators. The second linear simulation model is formulated subject to the solution of the first model, and solved by LP. The system can also be used for sensitivity analysis. What-if questions about changes in the circuit structure can easily be solved by the expert system. The structure of the expert system and knowledge base is depicted in Figure 1.

Fig.1 Structure of expert system and knowledge base
Knowledge-base modelling and expert systems in the minerals industry can be applied in various ways. Expert systems could be used to formalize all the existing experience concerning a particular metallurgical plant and utilize it to do fault analysis. The data bank for this expert system could be updated by plant superintendent and operator alike. Ores can be characterized by devising simple tests for a sample of ore, the results of which could be used by an expert system to predict the behaviour of the ore. Hence, the plant superintendent could establish operating conditions by consulting the expert system. An expert system could assist in selecting the correct empirical model for a particular set of operating conditions, hence rendering the empirical approach more general. Ultimately, the above mentioned methods can be applied in adaptive control loops to control various metallurgical plants. This approach focuses not only on equation oriented simulation, but also on reasoning capabilities, which utilize heuristic knowledge and other qualitative information so as to assist the engineer to verify the assumptions made regarding the process during the design stages. The structure of the expert system as part of an integrated methodology for plant optimization is shown in Figure 2.

![Diagram](https://scholar.sun.ac.za)

Fig.2 The expert system as part of an integrated methodology for plant optimization

**SIMULATION ROUTINE**

As discussed previously, the mass balance equations for the separation of \( k \) mineral elements are highly non-linear. These non-linear equations must be linearised in order to apply linear programming methods. The two linear simulation models will be discussed separately below.
The formulation of Linear Programming Model I

Linear Programming Model I optimizes the flow rates and the path of the valuable elements between the separator banks. Density is used to distinguish between different elements in a gravity separation plant. In this paper the valuable element h is characterized by only one species, and not by three species as in Reuter and Van Deventer (1990). This simulation model will be developed here by formulating all possible constraints imposed on the mass balance equations.

Mass balance constraints

The mass balance constraints define the flow rates of the valuable elements h between the banks of mineral separators. Consider the two separator banks i and j in Figure 3. The middlings stream z could recycle around any bank j, but would not constitute a product from the circuit.

From Figure 3 the following four steady-state mass balance equations may be derived:

\[ \sum_{i \neq j} r_{ij} x_{ij} + \sum_{i \neq j} y_{ij} + \sum_{i} r_{i} = z_{ij} \]  
\[ \sum_{i \neq j} r_{ij} x_{ij} + b_{ij} = m_{ij} \]  
\[ \sum_{i} r_{ij} = z_{ij} \]  
\[ b_{ij} = y_{ij} \]

Fig. 3 Steady-state flowrates of the valuable element h from bank j to i
External constraints

A number of external constraints should be imposed on the four mass balance equations. These constraints include the following:

(a) Due to production limitations, the feed rate to the plant cannot exceed certain limits. If desired, the fresh feed can be a multiple feed, i.e. it may be fed to all separator banks, if required.

$$u^L_{hj} \leq u_{hj} \leq u^U_{hj}$$

(b) Certain recycle streams which are not physically meaningful, must be excluded, e.g. recycling of rougher tailings to the cleaner banks.

$$r_{mj}^h = 0$$
$$r_{zj}^h = 0$$
$$r_{yj}^h = 0$$

(c) The designer must establish which separators produce concentrate and which produce tailings.

$$a^L_{hj} \leq a_{hj} \leq a^U_{hj}$$
$$b^L_{hj} \leq b_{hj} \leq b^U_{hj}$$

(d) The total flow rates from the separator banks may not exceed certain values owing to limitations in pipe diameter and pumping rate.

$$m^L_{hj} \leq m_{hj} \leq m^U_{hj}$$
$$z^L_{hj} \leq z_{hj} \leq z^U_{hj}$$
$$y^L_{hj} \leq y_{hj} \leq y^U_{hj}$$

Separator constraints

The separation of the feed into concentrate, middlings and tailings characterizes this constraint, which may be established either by the application of theoretical models, separation curves, empirical models or from plant experience.

The separation characteristics of the separator $j$ for the element $k$ may be defined in terms of the separation factors $y_{mkj}$ and $z_{mkj}$ as follows:

$$y_{mkj} = y_{kj}/m_{kj}$$

$$z_{mkj} = z_{kj}/m_{kj}$$

Due to variations in the conditions in separator bank $j$, the separation factors may vary between an upper and lower bound. By defining the separation factor for the valuable element $y_{mhj}$ as bounds, the optimization model becomes well-suited for inclusion in a knowledge-based simulator that permits the intelligent choice of bounds. Hence, if the separation factors are not known precisely, constraints (16) and (17) may replace equalities (14) and (15):

$$[y_{mhj}]_L \leq y_{hj} \leq [y_{mhj}]_U$$

$$[z_{mhj}]_L \leq z_{hj} \leq [z_{mhj}]_U$$
Objective function

The aim of the optimization is to maximize the recovery of the valuable elements \( h \), subject to the constraints (1)-(17). This can be stated as:

\[
OBJ = \sum_j w_{hj} a_{hj}
\]

(18)

where OBJ is the objective function which maximizes the production of valuable element \( h \); and \( w_{hj} \) is the price weight for the production of \( a_{hj} \), which depends on current operating costs (e.g. 100 monetary units/tonne).

Linear Programming Model I

The linear simulation model will consist of equations (1)-(18). This model produces the optimal circuit structure, and the optimal inter-bank flow rates and recovery of the valuable element \( h \).

The formulation of Linear programming model II

Linear Programming Model II simulates the flow rates of all non-valuable elements \( k \) in such a way that the concentrate grade is maximized. This model is subject to the optimal flow rates and the circuit structure produced by Linear Programming Model I, as well as the separation constraints for all elements \( k \).

Mass balance constraints

\[
\sum e_i^* m_{kj} + \sum c_i^* y_{kj} + \sum e_i^* y_{kj}
\]

From figure 4, the following six mass balance equations may be derived for element \( k \) flowing between separator banks \( i \) and \( j \) (variables denoted by an * have been evaluated by
**Linear Programming Model I:**

\[ \sum_{j, l \neq j} b_{kj} = B_{Tj} m_{kj} + \sum_{j, l \neq j} c_{ij} y_{kj} + \sum_{j} e_{ij} z_{kj} + v_{ki} = m_{ki} + y_{ki} + z_{ki} \]  
(19)

\[ b_{kj} = B_{Tj} m_{kj} \]  
(20)

\[ a_{kj} = \alpha_{cj} y_{kj} \]  
(21)

\[ r_{m_{ij}} = t_{ij} m_{kj} \]  
(22)

\[ r_{y_{ij}} = c_{ij} y_{kj} \]  
(23)

\[ r_{z_{ij}} = e_{ij} z_{kj} \]  
(24)

The bounds on the feed rate are given by

\[ u_{k} \leq u_{kj} \leq u_{ukj} \]  
(25)

The optimal flow rates of the valuable elements \( h \) produced by Linear Programming Model I are: \( m_{kj}, z_{kj}, y_{kj}, b_{kj} \) and \( a_{kj} \)

**Separator constraints**

These constraints are similar to those imposed on Linear Programming Model I [constraints (14) and (15), or (16) and (17)].

**Grade constraints**

The grade constraint for element \( k=1 \) is defined as:

\[ (G_{Tlj})^{L} \leq \frac{m_{lj}}{\sum m_{kj}} \leq (G_{Tlj})^{U} \]  
(26)

**Objective function**

Given the constraints (16) and (17), and (19)-(26), the objective is to maximize the grade of the valuable elements \( h \). This is equivalent to the minimization of the recovery of the non-valuable gangue elements \( (k \neq h) \). This can be stated as:

\[ \text{grade} = \sum_{j} (a_{hj} - \sum_{k, k \neq h} a_{kj}) \]  
(27)

where \( \text{grade} \) is the objective function which maximizes the grade, \( a_{hj} \) is the valuable element in the concentrate, and \( \sum_{k, k \neq h} a_{kj} \) are the non-valuable elements in the concentrate.

**Linear Programming Model II**

This model consists of equations (16) and (17), and (19)-(27).

**SAMPLE PROBLEMS**

The computer program was used to simulate three gravity concentration circuits in order to demonstrate its applicability during the design of separation plants. For the purpose of illustration, the use of Linear Programming Models I and II will be explained in more detail in example 1 than in the case of examples 2 and 3. The bounds on the separation factors were produced by taking 25% variations in the flow rates into account. In the sample
problems described below, a linear programming package, LP88, on an IBM AT personal computer, was used to solve the models. Less than 5 minutes of central processor time was required to solve the models in each case. A sensitivity analysis to determine the effect of multiple feed rates on the recovery of the valuable element was also carried out.

Example 1

The data for this example were taken from Van der Spuy (1982), and represent a rougher-scavenger-cleaner gravity separation circuit. Only the low phosphate circuit of the plant is simulated here. In the circuit, the feed consisting of 0.015 t/h of element 1 (uranotherianate, and depicted here as U$_3$O$_8$), 1.68 t/h of element 2 (ZrO$_2$) and 418.305 t/h of element 3 (gangue) are separated into concentrate and tailings. Figure 5 illustrates the simplified flowsheet.

![Fig.5 Circuit structure and flow rates for example 1](https://scholar.sun.ac.za)
The bounds on the separation factors for the three elements are given in Tables 1 and 2.

**TABLE 1 Bounds on separation factors \((y_{m_{kj}})\) for example 1**

<table>
<thead>
<tr>
<th>Element</th>
<th>Bank:</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(j = 1)</td>
<td>(j = 2)</td>
<td>(j = 3)</td>
</tr>
<tr>
<td>(k = 1)</td>
<td>2.25 - 3.75</td>
<td>2.25 - 3.75</td>
<td>2.25 - 3.75</td>
</tr>
<tr>
<td>(k = 2)</td>
<td>1.5 - 2.5</td>
<td>1.5 - 2.5</td>
<td>1.5 - 2.5</td>
</tr>
<tr>
<td>(k = 3)</td>
<td>0.375 - 0.625</td>
<td>0.45 - 0.75</td>
<td>0.3 - 0.5</td>
</tr>
</tbody>
</table>

**TABLE 2 Bounds on separation factors \((z_{m_{kj}})\) for example 1**

<table>
<thead>
<tr>
<th>Element</th>
<th>Bank:</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(j = 1)</td>
<td>(j = 2)</td>
<td>(j = 3)</td>
</tr>
<tr>
<td>(k = 1)</td>
<td>3 - 5</td>
<td>3 - 5</td>
<td>3 - 5</td>
</tr>
<tr>
<td>(k = 2)</td>
<td>0.75 - 1.25</td>
<td>0.75 - 1.25</td>
<td>0.75 - 1.25</td>
</tr>
<tr>
<td>(k = 3)</td>
<td>0.3 - 0.5</td>
<td>0.15 - 0.25</td>
<td>0.15 - 0.25</td>
</tr>
</tbody>
</table>

The separation constraints for the valuable elements \(\text{U}_3\text{O}_8\) and \(\text{ZrO}_2\) for all banks \(j\) are:

\[
2.25 \; m_{\text{U}_3\text{O}_8,j} \leq y_{\text{U}_3\text{O}_8,j} \leq 3.75 \; m_{\text{U}_3\text{O}_8,j};
\]

\[
1.50 \; m_{\text{ZrO}_2,j} \leq y_{\text{ZrO}_2,j} \leq 2.50 \; m_{\text{ZrO}_2,j};
\]

The bounds on the following streams were chosen arbitrarily:

\[
0 \leq m_{h,j} \leq 0.005; \; 0 \leq y_{h,j} \leq 0.01; \; 0 \leq z_{h,j} \leq 0.01
\]

\[
0 \leq b_{h,j} \leq 0.005; \; 0 \leq a_{h3} \leq 0.01
\]

Fresh feed is only fed to the rougher bank:

\[
u_{\text{U}_3\text{O}_8,1} = 0.015; \; u_{\text{ZrO}_2,1} = 1.68
\]

Only the cleaner bank produces concentrate:

\[
a_{h1} = 0; \; a_{h2} = 0
\]

The following recycle streams are excluded:

\[
rm_{h_{ij}} = 0, \text{ with } i, j = 1, 2, 3
\]

\[
ry_{h_{ij}} = 0, \text{ with } i = 1, 2; \; j = 1, 2, 3 \text{ and } i = j
\]

\[
rz_{h_{ij}} = 0, \text{ with } i = 1 \text{ and } j = 1, 2, 3
\]

\[
rz_{h_{ij}} = 0, \text{ with } i = 2, 3; \; j = 1, 2, 3 \text{ and } i = j
\]
The objective function for Linear Programming Model I is:

\[ \text{OBJ} = 100 a_{13} + 100 a_{23} \]

The optimal flow rates of the valuable elements produced by Linear Programming Model I are schematically shown in Figure 5. The following structural parameters are also produced by Model I:

- \( t^*_{ij} = 0 \), with \( i, j = 1, 2, 3 \)
- \( c^*_{ij} = 0 \), with \( i, j = 1, 2, 3 \); except \( c^*_{31} = c^*_3 = 1 \)
- \( e^*_{ij} = 0 \), with \( i, j = 1, 2, 3 \); except \( e^*_{21} = e^*_{22} = e^*_{33} = 1 \)
- \( \alpha^*_c = 0 \), \( \alpha^*_{c_1} = \alpha^*_{c_2} = 0 \), \( \alpha^*_{c_3} = 1 \)
- \( \beta^*_{11} = \beta^*_{12} = \beta^*_{13} = 1 \)

The following mass balance equations for the non-valuable element \( k \), are derived from the above structural parameters:

- \( u_{k1} = m_{k1} + y_{k1} + z_{k1} \) (routher bank)
- \( z_{k1} + z_{k2} = m_{k2} + y_{k2} + z_{k2} \) (scavenger bank)
- \( y_{k1} + y_{k2} + z_{k3} = m_{k3} + y_{k3} + z_{k3} \) (cleaner bank)

The feed rates of each element \( k \) to the rougher bank are fixed:

\[ u_{11} = 0.015; \quad u_{21} = 1.68; \quad u_{31} = 418.305 \]

\[ u_{kj} = 0, \text{ with } k = 1, 2, 3 \text{ and } j = 2, 3 \]

The grade constraint for the non-valuable element (element 3) for each bank \( j \) is defined as (designer's choice):

\[ m_{3j} / \sum_k m_{kj} > 0.99 \]

The objective function is

\[ \text{grade} = a_{U_{308},3} + a_{ZrO_2,3} - a_{\text{gangue},3} \]

The flow rates of the non-valuable mineral element, which are produced by this model, are depicted in Figure 5. The solution produced a recovery of 62.3% of the valuable element 1 (U3O8), compared with a recovery of 60% reported by Van der Spuy (1982). The U3O8 grade produced is 0.026% compared with 0.017% given by Van der Spuy. The ZrO2 grade produced by the model is 2.43% and the recovery 51.02%. Van der Spuy reports a ZrO2 grade of 4.2% and a recovery of 67%. Only a section of the plant described by Van der Spuy was simulated and the separation factors were characterized by bounds, hence the slightly higher recovery for U3O8. The reason for the difference between the grade and recovery of ZrO2 produced by the model and the actual values is that ZrO2 is recovered as a byproduct, but the same price weight as that of U3O8 was used in the first model.

Example 2

This example illustrates the simulation of a four-bank gravity separation circuit. The data were taken from Holland-Batt et al. (1982), except that the spirals were omitted from the circuit in the simulation as the latter showed that the spirals did not yield a significant
additional upgrading of the concentrate. The circuit consists of a rougher, scavenger, cleaner and recleaner. Reichert spirals and Reichert cone concentrators for the rougher, scavenger, cleaner and recleaners were used. In the circuit depicted in Figure 6, chromite (element 1) is separated from gangue (element 2). The separation factors for elements 1 and 2 are given in Tables 3 and 4.

The chromite grade produced is 97.02% and the recovery 96.93%, which are very close to the values of 95.05% for the grade and 95.05% for the recovery reported by Holland-Batt et al. (1982). The results for this example are depicted in Figure 6.

![Circuit structure and flow rates for example 2](image)

**TABLE 3** Bounds on separation factors ($y_{mkj}$) for example 2

<table>
<thead>
<tr>
<th>Element</th>
<th>Bank: j = 1</th>
<th>j = 2</th>
<th>j = 3</th>
<th>j = 4</th>
<th>j = 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>k = 1</td>
<td>21-22</td>
<td>4-7</td>
<td>20-34</td>
<td>17-29</td>
<td>1-2</td>
</tr>
<tr>
<td>k = 2</td>
<td>0.33-0.56</td>
<td>0.26-0.43</td>
<td>0.1-0.23</td>
<td>0.36-0.77</td>
<td>0.05-0.11</td>
</tr>
</tbody>
</table>
TABLE 4  Bounds on separation factors ($z_{m_kj}$) for example 2

<table>
<thead>
<tr>
<th>Element</th>
<th>Bank:</th>
<th>$j = 1$</th>
<th>$j = 2$</th>
<th>$j = 3$</th>
<th>$j = 4$</th>
<th>$j = 5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k = 1$</td>
<td></td>
<td>0.68-1.15</td>
<td>3.4-5.8</td>
<td>2.7-4.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k = 2$</td>
<td></td>
<td>0.14-0.24</td>
<td>0.12-0.21</td>
<td>0.17-0.29</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example 3

The data used for the simulation of a four-bank gravity separation circuit (shown in Figure 7) were taken from Voges (1982). A very low-grade tin ore is concentrated by three-stage jigging, spiral concentration and tabling to give a high grade concentrate. The circuit consists of a primary spiral (bank 1), secondary spiral (bank 2), primary table (bank 3) and secondary table (bank 4). In this plant a feed consisting of 0.0684 tonnes/hr of element 1 (tin) and 58.9316 tonnes/hr of element 2 (gangue) are separated into concentrate and tailings. The separation factors for each element are given in Table 5.

![Circuit diagram](image)

**All flow rates tonnes/hr**

- **Feed**
  - Element 1: 0.0684
  - Element 2: 58.926

- **Concentrate to bank 2**
  - Element 1: 0.0613
  - Element 2: 12.891

- **Concentrate to bank 3**
  - Element 1: 0.0595
  - Element 2: 3.2709

- **Concentrate to bank 4**
  - Element 1: 0.0541
  - Element 2: 0.9345

- **Tailings Element**
  - Element 1: 0.0071
  - Element 2: 46.0403

- **Concentrate Element**
  - Element 1: 0.0118
  - Element 2: 9.6204

- **Tailings Element**
  - Element 1: 0.0054
  - Element 2: 2.3364

- **Tailings Element**
  - Element 1: 0.0043
  - Element 2: 0.9207

**Fig.7 Circuit structure and flow rates for example 3**

No middlings streams and recycle streams are produced by the circuit. The results for this example are depicted in Figure 7. A tin recovery of 72.84% and grade of 78% are achieved by the simulation, while the reported tin recovery (Voges, 1982) is 67.50% and the tin grade is 66%. This discrepancy could be explained by the fact that the model used optimal values for the separation factors, which are not necessarily those used in the plant. As stated before, the bounds on the separation factors were selected arbitrarily to differ by ±25%, which is probably more than feasible on an industrial plant. As the knowledge-base in the
Simulation and optimization of gravity separation circuits

expert system is enlarged, more accurate bounds could be established, and such
discrepancies will then be reduced.

**TABLE 5** Bounds on separation factors ($y_{mkj}$) for example 3

<table>
<thead>
<tr>
<th>Element</th>
<th>Bank: j = 1</th>
<th>Bank: j = 2</th>
<th>Bank: j = 3</th>
<th>Bank: j = 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k = 1$</td>
<td>5 - 9</td>
<td>19 - 33</td>
<td>6 - 10</td>
<td>7 - 12</td>
</tr>
<tr>
<td>$k = 2$</td>
<td>0.28 - 0.46</td>
<td>0.34 - 0.56</td>
<td>0.40 - 0.67</td>
<td>0.015 - 0.025</td>
</tr>
</tbody>
</table>

Sensitivity analysis on example 2

The data of example 2 were used for the sensitivity analysis. The circuit was simulated for multiple feed rates, as shown in Table 6. Case no. 1 represents the base case as described for example 2.

**TABLE 6** Sensitivity analysis on example 2

<table>
<thead>
<tr>
<th>case</th>
<th>condition</th>
<th>recovery of chromite</th>
<th>grade of chromite</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Feed to bank 1</td>
<td>96.93%</td>
<td>97.02%</td>
</tr>
<tr>
<td>2</td>
<td>Feed to banks 1 and 2</td>
<td>92.05%</td>
<td>97.87%</td>
</tr>
<tr>
<td>3</td>
<td>Feed to banks 1-3</td>
<td>94.57%</td>
<td>96.94%</td>
</tr>
<tr>
<td>4</td>
<td>Feed to banks 1-4</td>
<td>95.93%</td>
<td>96.70%</td>
</tr>
</tbody>
</table>

In the cases no's 2-4, multiple feeds were applied to the same circuit configuration as that of example 2. The feed stream was fed to both banks 1 and 2 for case no. 2, to banks 1-3 in case no. 3, and to banks 1-4 in case no. 4. In the case of multiple feeds, the 1000 tonnes/hr of feed stream was divided equally between the different banks. The recovery of the chromite increases as the number of feed points increases, as can be seen in Table 6. These recoveries, however, are still lower than the case in which the feed point is only at bank 1 (case no. 1). The grade of the chromite, however, decreases as the number of feed points increases. It is interesting to note that the highest grade was obtained when the feed was divided equally between banks 1 and 2.

Although only multiple feed streams were considered in the sensitivity analysis, the program can also be used to investigate other changes in the circuit structure or the operating conditions which will determine the separation factors.

**CONCLUSIONS**

From the solution of the three application problems it can be concluded that:

(a) The circuit configurations, grades and recoveries produced by the simulation model are similar to those found in industry. The model can thus be used to simulate gravity concentration plants in practice.
(b) The simulation model could also consider multiple feeds and bounds on the feed rates.

(c) The simulation model produces solutions within minutes which makes it more efficient than time consuming iterative numerical methods.

(d) Since the separation characteristics are included as bounds rather than a model, approximate separation data could be used to accommodate any inaccuracies in the gravity concentration model.

(e) The inclusion of the operating conditions as bounds makes this model attractive for use in a knowledge-based system, which would manipulate the available data base to estimate bounds. The knowledge in the data base may be updated by the user or by the simulation model.

(f) The only data required by the model are the circuit parameters (number of banks and number of elements), feed rates and composition, and the separation factors.

The model can be used in the design of gravity separation plants, since exact separation data and feed rates may not be available in the designing stages. Subsequently, once the plant has been commissioned, the model may be used to simulate its operation. The simulation model may suggest an alternative circuit configuration and new operating conditions quickly if conditions change drastically in the plant.

ACKNOWLEDGEMENT

The sponsorship provided by the Council for Mineral Technology (Mintek) is greatly appreciated. This paper is published by permission of Mintek.

NOTATION

Variables of Linear Programming Models I and II

- Concentrate recovery of valuable element \( h \) from separator bank \( j \)
- Concentrate recovery of element \( k \) from separator bank \( j \)
- Tailings recovery of valuable element \( h \) from separator bank \( j \)
- Grade of element \( k \) in tailings stream \( j \)
- Objective function that maximizes the grade
- Tailings flow of valuable element \( h \) from separator bank \( j \)
- Tailings flow of mineral element \( k \) from separator bank \( j \)
- Total earnings due to the optimal recovery of valuable element \( h \)
- Recycle of tailings flow of valuable element \( h \) from separator bank \( j \) to \( i \)
- Recycle of concentrate flow of valuable element \( h \) from separator bank \( j \) to \( i \)
- Recycle of middlings flow of valuable element \( h \) from separator bank \( j \) to \( i \)
- Feed of valuable element \( h \) to separator \( j \)
- Feed of mineral element \( k \) to separator bank \( j \)
- Price weight of valuable mineral element \( h \) from separator bank \( j \)
- Concentrate flow of valuable element \( h \) from separator \( j \)
- Concentrate flow of mineral element \( k \) from separator bank \( j \)
- Separation constant between \( Y_{kj} \) and \( m_{kj} \)
- Middlings flow of valuable element \( h \) from separator \( j \)
- Middlings flow of mineral element \( k \) from separator bank \( j \)
- Separation constant between \( Z_{kj} \) and \( m_{kj} \)

Optimal values produced by Model I

- Fraction of \( y_j \) that is concentrate product
- Concentrate flow of valuable element \( h \) from separator \( j \)
- Fraction of \( m_j \) that is tailings product
Simulation and optimization of gravity separation circuits

\[ b_{*hj} \] tailings flow of valuable element h from separator j

\[ c_{*ij} \] fraction of \( y_j \) that is recycled from separator bank j to i

\[ e_{*jj} \] fraction of \( z_j \) that is recycled from separator bank j to i

\[ m_{*hj} \] tailings flow of valuable element h from separator j

\[ t_{*ij} \] fraction of \( m_j \) that is recycled from separator bank j to i

\[ y_{*hj} \] concentrate flow of of valuable element h from separator j

\[ z_{*hj} \] middlings flow of valuable element h from separator j

Superscripts

- \( h \): valuable element
- \( k \): element
- \( L \): lower bound on variable
- \( U \): upper bound on variable

REFERENCES


AN OFF-LINE COMPUTER SIMULATION FOR CONTROLLING THE REPULP SECTION IN A PLANT

J.A.M. RADEMAN, L. LORENZEN and J.S.J. van DEVENTER

Department of Chemical Engineering, University of Stellenbosch, Private Bag X5018, Stellenbosch, 7599, South Africa
(Received 9 September 1994; accepted 12 December 1994)

ABSTRACT

An off-line computer simulation program is proposed to control a repulping section of a plant that has previously been controlled solely by an operator. The off-line computer program can be used as a tool by the operator to improve the control of repulping a certain material before leaching commences. Controlling the repulping section is of special importance, especially to processes employing unit operations after the repulping section with fast reaction kinetics. The fast reaction kinetics make it essential to control the conditions of the incoming pulp to the leaching vessel. The simulation uses dynamic mass balance equations to simulate the physical properties of the system. It then solves the set of differential equations with an iteration procedure that uses the fourth order Runge-Kutta (RK) procedure and the Golden Section Search (GSS) optimisation procedure, thus obtaining optimum settings so that the desired conditions in the process could be maintained or reached as quickly as possible. The simulation was tested on a plant and indicated that considerable improvement in the stability of the operation could be achieved. Two calculation methods were also investigated for the mathematical models, i.e. the Runge-Kutta procedure mentioned above and the Laplace transform solution. Comparing the two methods it was evident that the Laplacian solution was not as accurate as the Runge-Kutta solution due to the inherent linearization step employed. This was especially evident for large deviations from the desired conditions.

Keywords
Repulping; leaching; control; optimisation.

INTRODUCTION

With the development of process control technology, the industry adopted more modern control methods by using computerised control. Therefore, processes with higher capacities, smaller equipment and higher yields could be developed, thus making processes more sensitive to small variations in operating conditions, which in turn necessitates more stringent control. During difficult economic times, it is important that existing operations run efficiently. Some improvements are usually possible by the more productive use of available instrumentation and control hardware. Some additional, simple and cheap control systems may significantly improve production.

The concept of the off-line computer simulation is to help improve the process and the control thereof with an inexpensive tool — a computer. All hydrometallurgical processes involving solids have at least one unit operation in common and that is repulping. This is usually the part of a process that is neglected in the
sense of its control, and depending on the process and the conditions, a poorly controlled repulp section could negatively influence other unit operations further down stream. Because repulping is one of the primary unit operations, especially in leaching processes, more stringent control on this operation could have considerable improvements in the operational stability of the down stream unit operations, as well as possible cost savings depending on the sensitivity of the unit operation following the repulping section. Therefore the case study presented in this paper is to demonstrate the concept and methodology of controlling the repulping section with an off-line computer simulation and the advantages of having better control on this section for any applicable processes.

The main objective of the computer simulation described in the case study, is to restrict perturbations coming from the repulping section to the leaching process, which could cause instabilities in the leaching process. The aim of the simulation program is to prevent perturbations from leaving the repulp section, rather than to rectify for these perturbations in the leaching process.

Two methods were investigated for the calculation of the mathematical model. The obvious method would be to use Laplace transforms for the calculation procedure, or the alternative method is to use a numerical method for the calculation procedure. However, to use the Runge-Kutta (RK) procedure to predict the different settings in the repulp section, an optimisation procedure was necessary. A one-dimensional region elimination method, the Golden Section Search (GSS) method was chosen due to the following: (i) it is an easy method to compile in a program, (ii) calculation and optimisation method (or time) is fast, and (iii) this method’s accuracy is sufficient for this specific problem.

THE NEED FOR PROCESS CONTROL: CASE STUDY

The repulping section which will be discussed in this case study (see Figure 1) consists of a feed stream from the mill into tank 1. From tank 1 the milled pulp stream is passed through a stream splitter into tank 2 and finally to the leaching vessel. Tanks 1 and 2 can also be described as two CSTR’s (continuous stirred-tank reactors) due to the reactions that occur in these tanks, therefore the repulp section can also be called the preleach section. In this case study it is necessary to know the basic leaching reactions occurring in the repulping section (preleach) and leaching process following the repulping section, in order to have a better understanding of the possible improvements induced by the off-line computer simulation.

![Fig. 1 Repulping section of the plant.](https://scholar.sun.ac.za)
The material being leached in this process consists mainly of sulphide minerals (AxSy and BxSy) and some alloys of species A. The approximate compositions of the material and acid solution are:

<table>
<thead>
<tr>
<th></th>
<th>Material</th>
<th>Acid solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Species A</td>
<td>51.2 %</td>
<td>28.3 g/l</td>
</tr>
<tr>
<td>Species B</td>
<td>28.0 %</td>
<td>21.6 g/l</td>
</tr>
<tr>
<td>Acid</td>
<td>-</td>
<td>100 g/l</td>
</tr>
</tbody>
</table>

The main leaching reactions that occur in the pulp are:

\[ \text{AxSy} + y\text{B}^{2+} \rightarrow y\text{A}^{2+} + \text{A}_{(x-y)}\text{S} + y\text{B}_{(y-1)} \]  
\[ \text{AxSy} + 2\text{H}^+ + \frac{1}{2}\text{O}_2 \rightarrow (x-y)\text{A}^{2+} + y\text{AS} + \text{H}_2\text{O} \]  
\[ \text{A} + \text{B}^{2+} \rightarrow \text{B} + \text{A}^{2+} \]

(i) \hspace{1cm} (ii) \hspace{1cm} (iii)

Batch experiments emulating the conditions in tanks 1 and 2 confirmed that substantial reactions occur in these two tanks. The purpose of the simulation is therefore to maintain the optimum reaction conditions in these two tanks. Therefore the optimum reaction conditions for tanks 1 and 2 would be where the maximum leaching for species A, equations (i) and (ii), and maximum cementation of species B, equations (i) and (iii), occur. At the current operating conditions the approximate residence times in tanks 1 and 2 are 300 minutes and 180 minutes, respectively. The leaching efficiencies of species A for tanks 1 and 2 with the indicated residence times are 7.2% (with a pulp density of 1900 kg/m\(^3\)) and 6.5% (with a pulp density of 1600 kg/m\(^3\)), respectively.

The data for Figures 2–5 were obtained from batch experiments emulating the conditions in tanks 1 and 2 and the leaching vessel. Figure 2 indicates the variation in the concentration of species A in the solid.
phase, the concentration of species B in the solution and the pH changes with time. Figure 3 shows the variation in the concentration of species A in the solid phase and the concentrations of species B and acid in the solution phase for tank 2. From Figures 2 and 3 the leaching rate of species A, the cementation rate of species B out of the solution and the pH increase (acid decrease) are evident. The results as presented in Figures 2 and 3 emphasise the need for controlling the residence time (tank levels) in tanks 1 and 2.

![Graph showing concentration changes over time]

**Fig.3** The experimental batch emulation of Tank 2.

The aim of this leaching process is to leach species A and at the same time keeping species B cemented in the solid state. Therefore, the problem that arises from the repulping section, is that the pulp from tank 2 sometimes still contains a high concentration of species B in the solution when it arrives at the leaching vessel. In such a case the leaching process would not be able to cement species B out of the solution in the leaching time available in the actual leaching vessel.

The need to control the repulp section is confirmed by the results of Table I, which gives the assays of samples taken from tanks 1 and 2 of an operating plant. From Table 1, at times 8, 9 and 10 hours, the solution analyses of tank 2 indicate a large increase in species B and acid concentrations which will result in a perturbation to the leaching process, following the repulping of the material. This perturbation could be ascribed to the higher pulp density in tank 1, because less acid was added to this tank and less reaction could thus occur. When the pulp was pumped to tank 2, more acid was needed in this tank to maintain the desired pulp density, which increased the acid concentration, as well as the concentration of species B (acid solution contains approximately 21g/l of species B) in this tank. This inevitably led to a higher concentration of species B in the leaching vessel.

The effect of the variation in the pulp density is most evident in the leaching vessel according to the batch experiments conducted, emulating the leaching vessel with variations in the pulp density. The results from these experiments indicate that the leaching rate of species A (Figure 4) increase with a decrease in the pulp density of the reaction mixture. Figure 5 indicates on the other hand that the cementation of species B is faster and more complete at higher pulp densities. The reason for this could be that, for a lower pulp density more acid solution is added, which means that more of species B is added, which in turn must be cemented in the leaching process. These results confirm the necessity of controlling the density of the pulp going to the leaching vessel in order to obtain the maximum leaching of species A, while keeping the concentration of species B in the solution below a certain concentration.
Fig. 4 The effect of pulp density on the leaching characteristics of species A in the leaching vessel, as emulated in a batch test.

TABLE 1 Solution samples taken from the repulp section every hour

<table>
<thead>
<tr>
<th>Time hours</th>
<th>Level %</th>
<th>Tank 1 Pulp density kg/m³</th>
<th>Species B g/l</th>
<th>pH</th>
<th>Level %</th>
<th>Tank 2 Pulp density kg/m³</th>
<th>Species B g/l</th>
<th>Acid g/l</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>84</td>
<td>2060</td>
<td>0.001</td>
<td>4.7</td>
<td>74</td>
<td>1510</td>
<td>0.012</td>
<td>41.0</td>
</tr>
<tr>
<td>2</td>
<td>90</td>
<td>2050</td>
<td>0.004</td>
<td>3.1</td>
<td>103</td>
<td>1560</td>
<td>0.002</td>
<td>45.6</td>
</tr>
<tr>
<td>3</td>
<td>102</td>
<td>2050</td>
<td>0.007</td>
<td>4.6</td>
<td>72</td>
<td>1570</td>
<td>0.007</td>
<td>32.9</td>
</tr>
<tr>
<td>4</td>
<td>92</td>
<td>2020</td>
<td>0.032</td>
<td>4.0</td>
<td>99</td>
<td>1550</td>
<td>0.003</td>
<td>29.9</td>
</tr>
<tr>
<td>5</td>
<td>62</td>
<td>2020</td>
<td>0.003</td>
<td>5.6</td>
<td>55</td>
<td>1610</td>
<td>0.006</td>
<td>35.1</td>
</tr>
<tr>
<td>6</td>
<td>50</td>
<td>1980</td>
<td>0.003</td>
<td>2.3</td>
<td>96</td>
<td>1560</td>
<td>0.006</td>
<td>58.6</td>
</tr>
<tr>
<td>7</td>
<td>87</td>
<td>2120</td>
<td>0.004</td>
<td>3.2</td>
<td>64</td>
<td>1600</td>
<td>0.005</td>
<td>41.2</td>
</tr>
<tr>
<td>8</td>
<td>90</td>
<td>2180</td>
<td>0.011</td>
<td>5.5</td>
<td>59</td>
<td>1610</td>
<td>2.5</td>
<td>80.2</td>
</tr>
<tr>
<td>9</td>
<td>70</td>
<td>2170</td>
<td>0.007</td>
<td>5.5</td>
<td>84</td>
<td>1600</td>
<td>6.1</td>
<td>85.0</td>
</tr>
<tr>
<td>10</td>
<td>65</td>
<td>2100</td>
<td>0.002</td>
<td>5.5</td>
<td>100</td>
<td>1610</td>
<td>9.6</td>
<td>82.7</td>
</tr>
</tbody>
</table>

Fig. 5 The effect of pulp density on the leaching characteristics of species B in the leaching vessel, as emulated in a batch test.
Control strategy

The simulation will be used as a tool to help the operator to control the tank levels (which in effect will mean the residence time in the tanks) and pulp densities in the repulping section. The tank levels and pulp densities are controlled by calculating the flowrates of the pulp and acid streams to and from tanks 1 and 2 so that the desired conditions could be maintained, or reached. Therefore, the rationale for developing the computer program to control the repulp section is to obtain a pulp stream flowing to the leaching vessel with a constant pulp density and a constant chemical composition.

One function of this model is to maintain the pulp densities in tank 1 and especially tank 2, at the desired pulp density. The desired pulp density in tank 2 is achieved by changing the feedrates of the pulp from tank 1 and the acid feed stream. Maintaining the desired pulp density in tank 1 is difficult because the pulp stream coming from the milling circuit is not continuous, as well as the untimely re-entering of recycled solids to tank 1. The recycle of solids to tank 1 and the acid solution used to wash (and repulp) the solids into tank 1, are taken as a continuous input (although it is actually a step input), in order to simplify the calculating procedure.

Controlling the tanks at specific pulp levels is as important as maintaining the desired pulp densities, because the variations in the tank levels will influence the residence time of the pulp in both tanks 1 and 2, and therefore the extent of the reaction. Because of the non-continuous pulp stream coming from the milling section to tank 1, the level of tank 1 cannot be controlled at a specific level. The milling operation starts when the level is below the minimum level and stops when the level reaches the maximum level. The desired level of tank 2 is obtained by manipulating the pulp and acid feedrates to tank 2.

OUTLINE OF SIMULATION PROGRAM

The attraction of this control method is that it requires only a personal computer (a PC with a 386 or higher processor). An operator familiar with a computer can be used successfully to maintain the system. Furthermore, it is inexpensive to implement, because no physical changes in the process equipment are necessary with only a computer that must be acquired. The trained operator can use the package to determine optimum conditions on the PC with plant data [1]. This enables him to make the necessary changes to the plant process control equipment quickly and correctly, if needed.

Input

The user has different options to enter the input values for calculations in the input section for the simulation program, depending on what information the user wishes to obtain. Therefore, the input options for the program are divided into three sections, i.e.

1) calculating the flowrate settings and predicting the time when milling operation will commence and end,

2) calculating the flowrate settings with the milling time known, and

3) calculating the conditions in this section when the flowrate settings are guessed.

Output

The outputs of the program will include the calculated settings for the flowrates, as well as the predicted conditions in tanks 1 and 2 at the end of the time period. As outputs the user has the following options to view the calculated results:
1) Mill conditions, milling times and conditions for tanks 1 and 2
2) Mill conditions and milling times
3) Conditions in tanks 1 and 2
4) Graph of the pulp density change with time for tanks 1 and 2
5) Graph for the tank level changes with time in tanks 1 and 2

MATHEMATICAL MODEL

Two assumptions were made during the compilation of the mathematical model to control the repulp section:

(i) all the feed streams to and from tanks 1 and 2 are continuous, except for the pulp stream from the mill to tank 1,
(ii) variations of pulp densities in the tanks will not directly affect the variation in the tank levels (the volume is independent of the pulp density)

The values of the different variables for the milling process are calculated with a steady state mass balance over the mill.

The demineralised water feed rate to the mill is:

\[ \frac{MF(1 - \frac{P_3}{P_M})}{(P_3 - P_w)} \]

The pulp flow rate from the mill to tank 1 is:

\[ F_3 = WF + \frac{MF}{P_M} \]  

The values of the different variables for tanks 1 and 2 are calculated with dynamic mass balance equations over tanks 1 and 2, respectively.

The dynamic state equations for tank 1 are as follows:

\[ \frac{d(P_1 V)}{dt} = P_1 F_1 + P_3 F_3 + P_4 F_4 + P_5 F_5 - P_2 F_2 \]  

\[ \frac{dV}{dt} = F_1 + F_3 + F_4 + F_5 - F_2 \]

Equation 4 was derived from equation 3 with the assumption that the change in the pulp volume is independent of a change in pulp density. This is assumed because no extensive leaching occurs in either tank 1 or 2, therefore the leaching would not have a noticeable effect on the pulp density of the pulp in either of the tanks. Therefore by substituting equation 4 into equation 3 an equation for the density changes of the pulp (due to the flowrates of the different streams entering the tank), equation 5, in tank 1 can be obtained.
The dynamic state equations for tank 2 are as follows:

\[
\frac{d(P_2, V_1)}{dt} = \frac{P_2, F_2}{V_1} + \frac{P_6}{V_1} F_6 - P_7 F_7 
\] (6)

\[
\frac{dV_1}{dt} = F_2 + F_6 - F_7 
\] (7)

Equation 7 was derived from equation 6 again with the same reasoning as discussed above for tank 1. Therefore by substituting equation 7 into equation 6 an equation for the density changes of the pulp in tank 2, equation 8, can be obtained.

\[
\frac{dP_2}{dt} = \frac{F_2}{V_1} (P_2 - P_7) + \frac{F_6}{V_1} (P_6 - P_7) 
\] (8)

Equations 4, 5, 7 and 8 could now be solved with the fourth order Runge-Kutta procedure.

**CALCULATION METHOD**

**Runge-Kutta**

The numerical method that was chosen to solve the differential equations was the fourth order Runge-Kutta method. This method is also the most widely used in computer solutions to differential equations [2].

The Runge-Kutta (RK) calculation procedure is used to calculate the pulp volume of tank 1 (V) and tank 2 (V_1), the pulp density in tank 1 (P_2) and tank 2 (P_7). (The tank level has a linear relation to the tank volume and is as follows: tank 1, \( h = \frac{V}{\text{Volume of tank} 1} \) and tank 2, \( h_1 = \frac{V_1}{\text{Volume of tank} 1} \). However, before these variables could be calculated the optimum input settings for the manipulated variables must be determined (acid flowrate to tank 1 (F_1), pulp flowrate from tank 1 to tank 2 (F_2) and the acid flowrate to tank 2 (F_6)).

**Golden Section Search**

The optimum flowrate settings for the manipulated variables (F_1, F_2 and F_6) are determined through a region elimination optimisation method, called the Golden Section Search (GSS) [3]. The strategy employed in the GSS is to locate the two interior points so that the interval eliminated on one iteration will be of the same proportion regardless of the interval length. For any particular interval \( L^k \), \( a^k \) and \( b^k \) are the current bounds of the interval at stage \( k \) in the search and the two interior points are located at \( x_1^k = a^k + X_5 L^k \) and \( x_2^k = b^k - X_5 L^k \). Where the fraction \( X_5 = \frac{3 - \sqrt{5}}{2} = 0.382 \) is known in ancient times as the "golden section".
Determining the function values for \( f(x^k) \) and \( f(x^l) \) and seeing that \( |f(x^k) - P_{T_1}| > |f(x^l) - P_{T_1}| \), hence for minimising, at the next step \( a^{k+1} = x^k \) and \( b^{k+1} - b^k \). The interior point \( x^k \) is replaced by \( x^{k+1} \) and we select the new interior point as \( x^{k+1} = b^{k+1} - x^k \). This procedure is repeated until the difference between the calculated value \( f(x^k) \) and the desired value \( (P_{T_1} \text{ or } P_T) \) is less than the specified margin of accuracy.

**Calculation sequence**

A short description of the calculation sequence in the program and the relevant rationale for the sequences which are presented in Figures 6a–e, are discussed below.

Fig.6a Flow diagram of the main program of the computer simulation to control the repulping section of a plant.
**Input options 1 and 2: (see Figure 6a)**

The whole procedure is iterative and begins where starting values for the manipulated variables are determined with the GSS method (Figure 6b) in that specific range of the variable (minimum and maximum). The starting values for the pulp flow from tank 1 to tank 2 (F2) and the acid flowrate to tank 2 (F6) are calculated first (F2 and F6 are the manipulated variables for tank 2), because F2 is necessary to perform the RK calculation on tank 1.

![Flow diagram of the Golden Section Search procedure 2 (GSS-2) of the computer simulation.](https://scholar.sun.ac.za)
A value for the acid flowrate to tank 1 \( (F_1) \) is determined \( (F_1 \) is the manipulated variable for tank 1) with the GSS method (Figure 6d), before the RK calculation (Figure 6e) for tank 1 is conducted. The RK calculation determines the pulp density \( (P_2) \) and the level \( (h) \) of the pulp in tank 1. The pulp density in tank 1 \( (P_2) \) calculated from the objective function is then used as the value which is compared with the desired value. If the calculated value is not within the margin of accuracy \( (99.95\%) \) which is specified, the GSS method continues to optimise for the acid flowrate to tank 1 \( (F_1) \). The pulp density in tank 1 \( (P_2) \) is used as the measure of accuracy for the GSS optimisation process. The acid flowrate to tank 1 \( (F_1) \) is the variable to be optimised and is dependent on the level of the pulp in tank 1 \( (h) \). The maximum flowrate of acid to tank 1 \( (F_1) \) is determined according to the available volume in tank 1:

\[
\frac{V_{\text{TOT}}-\text{VOL}}{\text{Time}} + F_2 - (F_3 + F_4 + F_5)
\]

This procedure will continue until the set condition is reached or it will terminate after a specified number of iterations \( (C_1) \) was performed.
After the acid flowrate to tank 1 ($F_1$) has been calculated, the RK calculation for tank 2 is conducted (Figure 6c). Due to the fact that the pulp density in tank 1 ($P_2$) is a disturbance to tank 2 the RK calculations for both tanks 2 and 1 are performed simultaneously for each time step, so that the influence of the variation of the pulp density in tank 1 ($P_2$) on tank 2 could be taken into account in the calculations. The objective function in the RK calculation for tank 2 will calculate the pulp density ($P_2$) and the level of the pulp ($h_1$) in tank 2. The calculated value of the pulp density in tank 2 ($P_2$) is used to compare with the desired value to determine if the chosen combination of $F_1$, $F_6$ and $F_2$ satisfies the required margin of...
accuracy. The flowrates of the pulp from tank 1 to tank 2 (\(F_2\)) and the acid to tank 2 (\(F_6\)) are dependent on the available volume in tank 2, as well as the flowrate from the pulp from tank 2 to the leaching vessel (\(\frac{V_{\text{TOT}} - \text{VOL}}{\text{Time}} + F_7\)). If the set condition is not met, the GSS procedure for determining the pulp flowrate from tank 1 to tank 2 (\(F_2\)) and the acid flowrate to tank 2 (\(F_6\)) will continue, therefore, repeating the whole process described thus far.

This whole procedure above will continue until the optimum values for \(F_1\), \(F_2\) and \(F_6\) are obtained (specific conditions are satisfied) or until the specified number of iterations (\(C_2\)) have been completed (measure to stop the program from running ad infinitum if no exact solution is possible).

**Input option 3 (see Figure 6a)**

In this section the input values for the acid flowrate to tank 1 (\(F_1\)), the pulp flowrate from tank 1 to tank 2 (\(F_2\)) and the acid flowrate to tank 2 (\(F_6\)) are given by the user, therefore the program will only perform the RK calculations for tanks 1 and 2 to determine the end conditions (pulp densities in tanks 1 (\(P_2\)) and 2 (\(P_7\)) and the pulp levels in tanks 1 (\(h_1\)) and 2 (\(h_2\))) of the process.
The Laplace transformation of the mathematical model is a linear model that approximates the dynamic behaviour of a non-linear system in the neighbourhood of specified operating conditions. Linearisation is widely used in the design of control systems for the following reasons [4]:

(i) Closed-form, analytical solutions are possible for linear systems. This is not possible for non-linear systems, and computer simulation provides us only with the behaviour of the system at specified values of inputs and parameters.

(ii) All the significant developments toward the design of effective control systems have been limited to linear processes.

The one major disadvantage of the Laplace transform solution of the mathematical model is that the necessary accuracy can not be obtained. Furthermore the Runge–Kutta(RK) solution (including the GSS optimisation method) of the mathematical model is more accurate than the Laplace solution, as can be seen from Table 2, by comparing the results of the two calculation methods for the pulp densities in tank 2 ($P_7$ - highlighted columns). In almost all the cases the RK solution calculated the pulp density in the required margin of accuracy, except for three situations where it was physically impossible to reach the desired pulp density in the specific time period. This in effect will mean that the flowrate settings ($F_2$ and $F_6$) calculated by the RK procedure are constantly more accurate than the Laplace solution.

**TABLE 2 Sensitivity analysis and comparison of the Laplace transform solution and the Runge–Kutta solution on the mathematical model for tank 2.**

<table>
<thead>
<tr>
<th>Initial $P_7$</th>
<th>Initial $h_1$</th>
<th>Laplace solution</th>
<th>Runge-Kutta solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_7$</td>
<td>$F_2$</td>
<td>$F_6$</td>
<td>$P_7$</td>
</tr>
<tr>
<td>1600</td>
<td>75</td>
<td>0.0223</td>
<td>0.0297</td>
</tr>
<tr>
<td>1700</td>
<td>75</td>
<td>0.0335</td>
<td>0.0185</td>
</tr>
<tr>
<td>1500</td>
<td>75</td>
<td>0.0110</td>
<td>0.0410</td>
</tr>
<tr>
<td>1600</td>
<td>73</td>
<td>0.0209</td>
<td>0.0356</td>
</tr>
<tr>
<td>1600</td>
<td>77</td>
<td>0.0237</td>
<td>0.0250</td>
</tr>
<tr>
<td>1700</td>
<td>73</td>
<td>0.0321</td>
<td>0.0232</td>
</tr>
<tr>
<td>1700</td>
<td>77</td>
<td>0.0350</td>
<td>0.0137</td>
</tr>
<tr>
<td>1500</td>
<td>73</td>
<td>0.0096</td>
<td>0.0457</td>
</tr>
<tr>
<td>1500</td>
<td>77</td>
<td>0.0125</td>
<td>0.0362</td>
</tr>
<tr>
<td>1800</td>
<td>75</td>
<td>0.0448</td>
<td>0.0072</td>
</tr>
<tr>
<td>1400</td>
<td>75</td>
<td>0.0520</td>
<td>1569.9</td>
</tr>
<tr>
<td>1600</td>
<td>65</td>
<td>0.0151</td>
<td>0.0535</td>
</tr>
<tr>
<td>1600</td>
<td>85</td>
<td>0.0294</td>
<td>0.0059</td>
</tr>
<tr>
<td>1800</td>
<td>65</td>
<td>0.0377</td>
<td>0.0310</td>
</tr>
<tr>
<td>1800</td>
<td>85</td>
<td>0.0353</td>
<td>0.1578</td>
</tr>
<tr>
<td>1400</td>
<td>65</td>
<td>0.0687</td>
<td>1618.3</td>
</tr>
<tr>
<td>1400</td>
<td>85</td>
<td>0.0069</td>
<td>0.0284</td>
</tr>
</tbody>
</table>
The problem with the inaccurate solution of the mathematical model with Laplace transforms is due to the fact that the values that are used to linearise the model around the steady state conditions of the process are not close enough to the initial values. This happens in any linearisation procedure where the point at which the linearisation occurred is not close enough to the point of calculation. Therefore, the further the initial value is from the point of linearisation the more likely it is that the inaccuracy of the calculation will increase.

PROGRAM IMPLEMENTATION

The simulation program was implemented in a plant for a period of 16 hours to determine the effectiveness of the simulator. Because little or no automatic measuring equipment is in use in this section of the plant, it was quite difficult to perform a test run with the program. The flowrates of all the streams had to be measured (with a bucket and a stopwatch) and the valves had to be calibrated manually. The valves had to be calibrated so that the settings calculated by the program could be made reasonably accurately. The experimental and simulated results for the two tanks are presented in Figures 7-10.

![Graph](https://example.com/graph.png)

Fig. 7 Comparison of the predicted and actual levels in Tank 1.

The program gives a good simulation of the tank level changes in tank 1 (Figure 7) and a reasonable simulation of the tank level in tank 2 (Figure 8). From Figure 7 it is evident that the predicted tank level of tank 1 is for practical purposes very accurate. The predicted tank levels for tank 2 show some variations from the actual levels of the pulp in the tank. The desired pulp level in tank 2, which is being controlled by the simulation program, is 80.0%. The constant variation in the actual level of tank 2 is due to the sensitive nature of the setting on the stream splitter; i.e. the screen box of the pump feeding the stream splitter tends to get choked, therefore decreasing the flowrate to the stream splitter. The pump is designed to pump at approximately 0.30 m³/min. A stream with a flowrate of approximately 0.03 m³/min is splitted from the main stream from the pump, hence a small variation in the pump flowrate will have a significant influence on the level in tank 2. This is confirmed by the results for the predicted level of tank 1 (Figure...
7) and the actual level of tank 2 (Figure 8). At 15 hours the predicted level of tank 1 is 35.5% and the actual level 57.0% and for tank 2 it is the opposite where the predicted level is 80.0% and the actual level is 64.0%. Practically this indicates two possibilities for these variations: (i) the calibrated settings on the valve of the acid solution flowrate to the tank 1 are inaccurate, and (ii) the setting on the stream splitter did not deliver the desired flowrate to tank 2 due to choking of the pump feeding the stream splitter.

Fig. 8 Comparison of the predicted and actual levels in Tank 2.

Fig. 9 Comparison of the predicted and actual pulp densities in Tank 1.
The desired pulp densities for tanks 1 and 2 are 2000 kg/m$^3$ and 1600 kg/m$^3$, respectively. For both tanks 1 and 2 the program gives good simulation results for the pulp density changes in the tanks. The variations in the pulp density of tank 1 (Figure 9) could be ascribed to: (i) the solids recycled to tank 1 are not recycled on a continuous basis, and (ii) the amount of acid solution used to wash the recycled solids into tank 1 is not always the same.

The variations of the actual pulp density values in tank 2 (Figure 10) from the predicted values are mainly due to the ineffective stream splitting of the stream splitter and to the variations of the actual pulp densities from the calculated pulp densities in tank 1.

**CONCLUSIONS AND IMPLICATIONS**

The simulation program gives a reasonable prediction of the actual process, as discussed in the case study, in spite of the influence of the practical problems. More accurate and better control with the simulation program will be obtained if the control instrumentation of the process in question is more advanced.

The calculation of the mathematical model for simulation purposes indicated that the Runge-Kutta calculation method, which incorporates the Golden Section search optimisation procedure, is better than the Laplace transformation calculation for this specific type of control.

In general the implications of the control of the pulp density in a tank are very important for a variety of processes, other than leaching, such as flotation, roasting and filtration. This simulation concept, for example, could also be used in solvent extraction processes where the ratios of the reagents to be added could be calculated and controlled.

Level control in any tank in a plant will help to minimise spillages, prevent stoppages due to empty tanks, simplify the operator's job and therefore ensures that processes run more smoothly.
REFERENCES


NOMENCLATURE

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_1$</td>
<td>Iteration limit (tank 1) for Golden Section Search procedure</td>
<td></td>
</tr>
<tr>
<td>$C_2$</td>
<td>Iteration limit (tank 2) for Golden Section Search procedure</td>
<td></td>
</tr>
<tr>
<td>$F_1$</td>
<td>The flowrate of acid solution to tank 1</td>
<td>m³/min</td>
</tr>
<tr>
<td>$F_2$</td>
<td>The pulp flowrate from tank 1 to tank 2</td>
<td>m³/min</td>
</tr>
<tr>
<td>$F_3$</td>
<td>Pulp flowrate from mill to tank 1</td>
<td>m³/min</td>
</tr>
<tr>
<td>$F_4$</td>
<td>Flowrate of recycled solids to tank 1</td>
<td>m³/min</td>
</tr>
<tr>
<td>$F_5$</td>
<td>Solution flowrate used to wash the recycled solids into tank 1</td>
<td>m³/min</td>
</tr>
<tr>
<td>$F_6$</td>
<td>The acid flowrate to tank 2</td>
<td>m³/min</td>
</tr>
<tr>
<td>$F_7$</td>
<td>Pulp flowrate from tank 2 to the leaching vessel</td>
<td>m³/min</td>
</tr>
<tr>
<td>$F_{1\text{MIN}}$</td>
<td>The MINIMUM flowrate of acid solution to tank 1</td>
<td>m³/min</td>
</tr>
<tr>
<td>$F_{1\text{MAX}}$</td>
<td>The MAXIMUM flowrate of acid solution to tank 1</td>
<td>m³/min</td>
</tr>
<tr>
<td>$F_{2\text{MIN}}$</td>
<td>The MINIMUM pulp flowrate from tank 1 to tank 2</td>
<td>m³/min</td>
</tr>
<tr>
<td>$F_{2\text{MAX}}$</td>
<td>The MAXIMUM pulp flowrate from tank 1 to tank 2</td>
<td>m³/min</td>
</tr>
<tr>
<td>$F_{6\text{MIN}}$</td>
<td>The MINIMUM flowrate of acid solution to tank 2</td>
<td>m³/min</td>
</tr>
<tr>
<td>$F_{6\text{MAX}}$</td>
<td>The MAXIMUM flowrate of acid solution to tank 2</td>
<td>m³/min</td>
</tr>
<tr>
<td>(h)</td>
<td>The level of tank 1</td>
<td>%</td>
</tr>
<tr>
<td>(h_1)</td>
<td>The level of tank 2</td>
<td>%</td>
</tr>
<tr>
<td>MF</td>
<td>Material feedrate to the mill</td>
<td>t/h</td>
</tr>
<tr>
<td>$P_1$, $P_5$, $P_6$</td>
<td>Pulp density of the acid solution</td>
<td>kg/m³</td>
</tr>
<tr>
<td>$P_2$</td>
<td>The actual pulp density in tank 1</td>
<td>kg/m³</td>
</tr>
<tr>
<td>$P_3$</td>
<td>Desired pulp density of the mixture leaving the mill</td>
<td>kg/m³</td>
</tr>
<tr>
<td>$P_4$</td>
<td>The pulp density of the recycled solids to tank 1</td>
<td>kg/m³</td>
</tr>
<tr>
<td>$P_7$</td>
<td>The actual pulp density in tank 2</td>
<td>kg/m³</td>
</tr>
<tr>
<td>$P_M$</td>
<td>Density of the material</td>
<td>kg/m³</td>
</tr>
<tr>
<td>$P_T$</td>
<td>The desired pulp density in tank 1</td>
<td>kg/m³</td>
</tr>
<tr>
<td>$P_{11}$</td>
<td>The desired pulp density in tank 2</td>
<td>kg/m³</td>
</tr>
<tr>
<td>$P_W$</td>
<td>Density of the demineralised water added to the mill</td>
<td>kg/m³</td>
</tr>
<tr>
<td>VOL</td>
<td>The initial volume of the pulp in tank 1</td>
<td>m³</td>
</tr>
<tr>
<td>VOL1</td>
<td>The initial volume of the pulp in tank 2</td>
<td>m³</td>
</tr>
<tr>
<td>$V$</td>
<td>The volume of the pulp in tank 1</td>
<td>m³</td>
</tr>
<tr>
<td>$V_1$</td>
<td>The volume of the pulp in tank 2</td>
<td>m³</td>
</tr>
<tr>
<td>$V_{11}$</td>
<td>The desired volume of the pulp in tank 2</td>
<td>m³</td>
</tr>
<tr>
<td>$V_{\text{MIN}}$</td>
<td>The MINIMUM volume of the pulp in tank 1</td>
<td>m³</td>
</tr>
<tr>
<td>$V_{\text{TOT}}$</td>
<td>The MAXIMUM volume of the pulp in tank 1</td>
<td>m³</td>
</tr>
<tr>
<td>$V_{\text{MIN}}$</td>
<td>The MINIMUM volume of the pulp in tank 2</td>
<td>m³</td>
</tr>
<tr>
<td>$V_{\text{MAX}}$</td>
<td>The MAXIMUM volume of the pulp in tank 2</td>
<td>m³</td>
</tr>
<tr>
<td>WF</td>
<td>Demineralised water flowrate to the mill</td>
<td>m³/min</td>
</tr>
</tbody>
</table>
Section 8  Publications on

Knowledge Based Systems
THE APPLICATION OF KNOWLEDGE-BASED SYSTEMS TO THE SIMULATION OF GOLD EXTRACTION PROCESSES

M.A. REUTER*, J.S.J. VAN DEVENTER & I.W. VAN DER MERWE

Department of Metallurgical Engineering, University of Stellenbosch, 7600 Stellenbosch, South Africa.
(Received 15 August 1990; accepted 17 August 1990)

ABSTRACT

Most unit operations and processes of gold extraction are ill-defined and cannot be simulated adequately by conventional mathematical models. This is caused mainly by the complex chemistry and mineralogy of most ore slurries. An object-oriented knowledge-based system (KBS) is proposed here for the dynamic simulation and fault-diagnosis of various sub-processes in the extraction of gold. Two approaches are used, viz.: (1) where the parameters of simple empirical expressions are related to a data base via an evolution of operating conditions, and (2) where the dynamic behaviour of a system is finger-printed by the gradient of change of a state variable. The first approach is applied to the simulation of gold leaching in batch reactors, cascades of continuous reactors, and countercurrent Kamyr towers. The depletion of cyanide and/or oxygen is also taken into account.

The second approach is applied to CIP and CIL cascades. The centre (deep knowledge) of this model is a database containing concentration-time data and an accompanying generalized kinetic model. The knowledge-base is defined by various facts, objects, rules and functions, which capture both deep as well as shallow knowledge regarding the process. Furthermore, it is explained how KBS simulation can be used in fault-diagnostics and the identification and characterisation of ores, slurries and adsorbents. The proposed KBS produces realistic simulations of experimental and published data. It is evident that the accuracy of prediction is entirely dependent on the accuracy and population density of the data base. In most cases, dynamic simulation by KBS has reduced CPU time drastically in comparison with the numerical solution of more phenomenological models.

Keywords

Expert system, knowledge base system, gold metallurgy, activated carbon, leaching, cyanide, adsorption

INTRODUCTION

Despite intensive efforts to model unit operations and circuits in the mineral processing industries over the last two decades, most sub-processes of gold extraction, such as milling, flotation, CIP adsorption, ion-exchange and leaching are still ill-defined. In most cases, these processes cannot be simulated accurately and in a generalized way by fundamental mathematical equations that usually require a large number of parameters to fit the data.

* Present address: Institut für Metallhüttenwesen und Elektrometallurgie, R.-W. Technische Hochschule, 5100 Aachen, Germany
Even in the more phenomenological models it is sometimes difficult to interpret the value of parameters in terms of specific physical phenomena. Existing models are usually incapable of considering the effects of pH, ionic strength, carbon deactivation and ore mineralogy.

Moreover, the chemistry and mineralogy of most slurries are complex and ill-defined, which means that in most cases the results obtained for one ore cannot be applied to another ore. The liberation pattern of different minerals, and the associated density distribution according to particle size, specific gravity, hydrophobicity, magnetic susceptibility or leachability, are extremely difficult to determine a priori. Consequently, it is very difficult to design gold extraction plants optimally, or to devise a control strategy for such plants. Empirical simulation models can usually be developed only after commissioning of a plant, and use mainly historical data from a specific plant.

Nevertheless, this type of ill-defined problem is not unique to the minerals industry, and has received attention in the literature. Recently, knowledge-based systems (KBS) have found increasing application for treating ill-defined problems, especially in the chemical industry [1-12]. Although considerable attention has been paid in the literature to the application of expert systems and artificial intelligence to chemical engineering problems, little emphasis has been placed yet on similar applications in mineral processing. The expert system of Tucker and Lewis [13] for shaking table diagnostics is an example of such pioneering work.

Most KBS in the chemical industry have been formulated for fault diagnostics, steady-state circuit design and simulation, process selection, operator training, data management and scenario analysis by means of "what if?" questions [1-12]. However, the dynamic simulation of ill-defined processes by use of KBS is a new concept, and as such has received little attention in the literature. Reuter and Van Deventer [14,15] explained the application of KBS to the simulation of CIP and CIL circuits. The concept of qualitative simulation has been explored in this regard [16], but has clearly not advanced to the stage where it can be used for complex systems, or where it can compete with a hybrid of KBS and differential equations.

It is the objective of this paper to broaden the concepts of KBS computing published earlier [14,15], with special reference to the extraction of gold. Two approaches will be explained, viz.: (1) where the parameters of simple empirical expressions are related to a data base via an evolution of operating conditions, and (2) where the dynamic behaviour of a system is finger-printed by the gradient of change of a state variable. The first approach will be applied to the simulation of gold leaching in batch reactors, cascades of continuous reactors, and countercurrent Kamyr towers. The consumption of cyanide and oxygen will also be taken into account. The second approach will be applied to CIP and CIL cascades. Furthermore, it will be explained how KBS simulation can be used in fault-diagnostics and the identification and characterisation of ores, slurries and adsorbents.

**GENERAL CHARACTERISTICS OF KBS**

An *expert system* is a computer program that behaves like an expert in a usually narrow field of application [1-12]. One of the features of an expert system is the separation of knowledge from the techniques that are used to think about this knowledge. In view of the fact that expert systems require expert knowledge in some form, they are also called KBS. An expert system should be able to explain its behaviour and decisions to the user. In addition, a KBS should be capable of dealing with uncertainty and incompleteness of information. An expert system consists usually of a *knowledge base*, an *inference engine* and a *user interface*. The *inference engine* contains the techniques and procedures that drive the thought process. An expert system *shell* is the collective term for a module consisting of the inference engine and the user interface.
The application of expert systems or KBS in process engineering may be grouped into the following areas:

The synthesis of process engineering flow sheets in hybrid type systems, i.e. the integration of KBS with equation based simulation programmes [1–3];

Programmes for the selection of equipment, processes or reagents, e.g. DECADE for the selection of catalysts [4,5];

Synthesis of operating procedures for process plants [1,6–8];

Malfunction diagnosis for complete process plants [9–12];

Control synthesis [1,17–19].

Although different architectures are used to define a KBS, most KBS use both qualitative (experiential and heuristic) as well as quantitative (equation orientated) knowledge, also termed shallow and deep knowledge respectively [1–12]. In these systems, data and knowledge are mostly represented as:

Objects (a chunk of structured knowledge [1]), which are also called frames;

Production rules, which are characterized by if-then type statements, and

Functions, which define the equations for quantitative simulation of unit operations.

These data structures form the basis of the various knowledge-based systems and are embedded in various architectures which include:

Object-oriented [1,12] or frame-oriented [3] systems;

Hybrid systems in a Blackboard (Blackboard is the global database permitting the interaction of all knowledge sources) architecture [4,5];

Hierarchical diagnostic systems in which every node in the structure represents a malfunction hypothesis, the top node containing general malfunction information and vice versa, hence the information is grouped into various levels of abstraction [6,9], or

Goal-tree-success-tree architecture used for the design of control systems [19].

Most of the systems that perform malfunction diagnosis do so on a macroscopic level i.e. analyse flowrates, temperatures etc. and suggest appropriate remedial action. However, few of the systems in the literature attempt to analyse the environment within a particular unit operation and suggest appropriate action on the basis of the intrinsic parameters that characterise the process. This weakness, as well as the inability of most systems to embrace dynamics, are addressed in this paper.

**REPRESENTATION OF KNOWLEDGE IN THE KBS**

An object-oriented approach was applied in the KBS proposed here in order to define the structure of the knowledge-base. Most of these systems implement object-oriented languages such as KEE or CommonLISP [1] for the programming of the qualitative model. Nevertheless, Turbo Pascal was used to program the proposed KBS so as to permit the definition of complex data structures and perform numerical analysis in the same programme. This avoids the time consuming communication between two different programmes, i.e. one to perform the knowledge engineering and the other the numerical simulation. A very large data base will require an AT machine for efficient computation.
Techniques of knowledge (shallow and deep) representation form an important aspect of the proposed KBS system. As discussed by Tzouanas et al. [17], these techniques may include facts, objects (frames), rules and functions:

**Facts** may include the type of unit operation or process under consideration, such as leaching, adsorption, ion-exchange or flotation, or the type of reactor system, such as a cascade of continuous stirred tanks or countercurrent columns.

Each **Object (Frame)** is structured so as to include all relevant information regarding a particular process variable and its effect on the process, or information regarding a particular unit operation e.g.:

**Unit-operation-object for a CIP-plant**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>unit operation:</td>
<td>continuous reactors</td>
</tr>
<tr>
<td>mass of adsorbent in reactor:</td>
<td>23.5g/l</td>
</tr>
<tr>
<td>Feed concentration:</td>
<td>2.26ppm</td>
</tr>
<tr>
<td>Pulp density:</td>
<td>1460kg/m³</td>
</tr>
<tr>
<td>volume of the reactor:</td>
<td>400l</td>
</tr>
<tr>
<td>% adsorbent leaking cocurrent:</td>
<td>0</td>
</tr>
<tr>
<td>number of reactors in series:</td>
<td>5</td>
</tr>
<tr>
<td>time one operating cycle [Ts]:</td>
<td>48h</td>
</tr>
<tr>
<td>number of operating cycles:</td>
<td>5</td>
</tr>
<tr>
<td>initial adsorbent loading:</td>
<td>0ppm</td>
</tr>
<tr>
<td>% of adsorbent transferred:</td>
<td>100%</td>
</tr>
<tr>
<td>feed flowrate of the pulp:</td>
<td>800l/h</td>
</tr>
<tr>
<td>% short circuiting of feed:</td>
<td>0</td>
</tr>
<tr>
<td>% of dead volume of reactor:</td>
<td>0</td>
</tr>
<tr>
<td>% of Ts used for transfer:</td>
<td>0</td>
</tr>
</tbody>
</table>

This object was used to compute the simulation results for the CIP plant discussed below. Although given here for a CIP-plant, the general structure of this object permits the simulation of any similar type of process such as resin-in-pulp. Another possible object is the **cost-object** used for economic evaluation.

The **Rules (production rules)** defined here fall into various categories which include the following:

**Adjustment rules (process specific)**

if pyritic ore type 7 is leached by cyanide in batch
and cyanide level is 150 p.p.m.
and level of oxygen is 6.5 p.p.m.
and temperature is 25°C
and pH = 13.1
and average particle size is 90 μm
and mass % solids in slurry is 54
and rate of agitation is 130 r.p.m.
and time of solid-liquid contact is 2 hours
then concentration gradient factor = 0.95

**Search rules**

For a pyritic ore from a particular ore-body in the mine for example, the chemical environment which produces a particular recovery and concentration - time gradient can be estimated as follows:
Simulation of gold extraction processes

if concentration gradient 2
and concentration gradient 1.8
and Au recovery is 96%
and Au recovery is 98%
then pH=12.4
density=1460 kg/m³
particle size is between 25 and 125 μm
etc.

The user has control over the search accuracy of this type of rule.

Procedural rules

As this model is an off-line system, these rules are addressed manually at present.

if leaching reaction
and train of continuous reactors
then use CSTR procedure

Functions include the kinetic models, differential equations that define the non-ideal flow through the continuous reactors, or equations that describe fundamental aspects of the process such as the Arrhenius equation. Hence, a function could contain the following elements:

- TYPE OF OPERATION (defining label)
  - Kinetic model (differential equation)
  - Unit-operation-object (as above)
  - Non-ideal flow model (differential equation)
  - Regression adjustment-objects which relate kinetic information to process conditions
  - Fundamental relationships such as mass or heat transfer correlations (e.g. $Sh=a(Re)^b(Sc)^c+d$)

TREATMENT OF KINETIC DATA

The basis of the knowledge representation discussed above is the concentration-time data of the various processes, which represents the deep knowledge regarding the process. The remainder of the knowledge is structured in such a way that it has access to these data or transformations thereof in order to perform simulations and interactive fault-diagnosis. A knowledge-base that is general requires a process independent kinetic model. Such a general kinetic model has been formulated for CIP and CIL processes by Reuter and Van Deventer [14,15], which has subsequently been extended to various metallurgical and other mineral processing (e.g. flotation) operations. A similar methodology will be explained in the subsequent sections.

The kinetic models which define the deep knowledge of leaching and adsorption are developed below and then applied to continuous leaching, CIP and CIL processes. As mentioned previously, leaching data are treated by two different approaches, viz.: (1) where the parameters of simple empirical expressions are related to a data base via an evolution of operating conditions [20], and (2) where the dynamic behaviour of a system is finger-printed in terms of the gradient of change of a state variable [14,15].

Leaching Data

The kinetics of the dissolution of gold from cyanided pulps cannot be modelled in a straightforward manner. It has been found experimentally that the following model fits accurately most concentration-time data from a batch test on Witwatersrand ores [20]:
Solid Phase: \[ \frac{dg}{dt} = -k(g-g^*)^n = r_1 \] (1)

Liquid Phase: \[ \epsilon \frac{dC}{dt} = (1-\epsilon) \frac{dC}{dt} + k(g-g^*)^n = r_2 \] (2)

Thus, the empirical parameters \( k, n \) and \( g^* \) may be estimated from batch experimental data by regression analysis, and then related to process conditions according to the first approach mentioned above [20].

As a result of the leaching reaction, as well as the undesirable leaching of base metals and the oxidation of ore constituents, oxygen and cyanide are consumed. This causes the leaching environment in the reactor to be changing continuously. It is therefore necessary to adjust the leaching condition as time proceeds. The depletion of oxygen and cyanide can be simulated by a simple first order rate equation, where the rate "constant" evolves with changes in the operating conditions [20]. This approach is similar to the second approach mentioned above, where the dynamic behaviour of a system is finger-printed in terms of the gradient of change in a state variable [14,15]. It should be obvious that, in a CIP or CIL plant, where the carbon becomes deactivated owing to fouling by organics and fine ore solids, a similar gradual shift in the adsorption conditions can be accommodated easily by the KBS.

In terms of the second approach, the rate variable \( k[g(t)] \) is defined here to be only a function of the grade \( g(t) \), and is determined from a batch leaching curve, which covers the leaching up to the equilibrium leaching point. It is clear that \( k[g(t)] \) is also a function of the knowledge-base via objects.

For the solution phase:
\[ \frac{dC}{dt} = k[g(t)]g(t) = r_3 \] (3)

For the solids phase:
\[ \frac{dg}{dt} = -\beta k[g(t)]g(t) = r_4 \] (4)

where:
\[ k[g(t)] = 2(C_{n+1}-C_n)/\delta t(g_{n+1}+g_n) \]
\[ \beta = (M_1/M_2) \]

Adsorption Data

As in the case of leaching data, the basis of the adsorption model is batch concentration-time data, and the corresponding loading-time curve that covers the adsorption profile up to the equilibrium loading [14,15]. Nicol et al. [21,22] presented an example of such data. From this curve and the corresponding concentration-time data, the rate variable \( k[y(t)] \), which is defined here to be a function of only the loading \( y(t) \), may be determined. This rate variable is subsequently used to predict the change in concentration and carbon loading at any carbon concentration \( M_c \), and at any initial concentration at the prevailing chemical process conditions.

\[ \frac{dC}{dt} = -(\alpha_5/\alpha_6)k[y(t)]C = r_5 \] (5)

\[ \frac{dy}{dt} = \alpha_5 k[y(t)]C = r_6 \] (6)

where:
\[ k[y(t)] = -2(C_{n+1}-C_n)/\delta t(C_{n+1}+C_n) \]
\[ \alpha_5 = (M_1/M_2) \]
\[ \alpha_6 = (M_1/M_2) \]

It must be noted here that these models do not attempt to suggest a mechanism for the reaction under consideration. What they do imply, however, is that a kinetic curve which describes a process up to equilibrium is a finger-print of that reaction and therefore of the reaction mechanism. Consequently, the model can be used as a basis to describe the
reaction under all possible conditions, given that the mechanism does not change. The database will obviously cover all possible mechanisms for a particular system. This basic finger-print curve and its associated shallow and deep level knowledge, defined as objects, are termed the pivot-data, or the standard condition. These pivot-data serve as a reference with which other curves are compared, and which can subsequently be used for fault-diagnosis or process identification.

Any condition that differs in whatever way from this standard condition or the pivot-data, is considered to be non-standard. Consequently, any change in the empirical parameters (such as in Eq. 1), or the profiles of $k(y(t))$ or $k(g(t))$ should be considered as the combined effect of different deviations from the standard condition, and therefore as taking all interactions into account. The changes in the leaching, adsorption or other reactions may be caused by changes in the mineralogy and chemistry of the ore, deactivation of the adsorbent, or changes in the concentration of reagents. It is convenient to express deviations from the pivot-data or standard parameters as percentages, which are estimated through experience, directly from the plant or from experimental data. It is also possible to express these deviations in terms of qualitative statements according to the principles of order-of-magnitude reasoning [23]. The proposed system is totally flexible, and more variables can be added through rules, if required.

OVERALL STRUCTURE OF THE KBS

Figure 1 illustrates the overall structure of the KBS, which consists mainly of: (1) A data-base for entering/editing of all relevant experimental and heuristic information; (2) An inference procedure for utilizing the data in the database to effect the diagnosis of problems, the identification and classification of unknown ores, and the simulation of different reactor systems; (3) A working memory which contains the current input and status of the specific problem being solved. The different modules of the system communicate via files in order to increase the working memory; (4) A user-friendly interface used to communicate with the above three components.

FAULT-DIAGNOSIS AND PROCESS IDENTIFICATION

If the type of process and reactor configuration (for example leaching in pachucas) are known, the KBS may perform fault-diagnosis or process identification of the system under consideration. This is effected by a search through both shallow and deep knowledge subject to if-then rules and a comparison to the pivot-data. The results of such a fault-diagnosis are saved in the system so as to facilitate learning of the KBS and an improvement of the expert capabilities of the system.

For example, a CIP or CIL plant may produce an unexpectedly high concentration of gold in the tailings. If sufficient knowledge about this particular plant and other similar operations has been saved in the KBS, a fault-diagnosis of this problem could have identified reasons such as poisoning of the carbon by organics, the origin of the organics, insufficient regeneration or elution of the carbon, etc. The more general data and heuristic information are saved in the system, and the more information is provided regarding a specific problem, the more exact the diagnosis from the expert system will be.

The KBS can also be used for the identification of ores, slurries, reagents or adsorbents. For example, the leaching characteristics of an unknown ore can be compared with those of known ores in the system at similar operating conditions [20]. The more information is known about the unknown ore, the more accurate the identification will be. This procedure is analogous to the mineralogical characterisation of ores by means of diagnostic leaching [24]. The IDENTIFY option of the proposed KBS enables the user to take advantage of the vast amount of information available on the leaching (for example) of known ores. In view of the tedious and expensive nature of experimental testwork to determine the complete leaching characteristics of an ore, it is advantageous to minimize such work and to rather utilize the existing body of relevant information in the KBS [20].
Any two ores, leached at the standard or pivot condition, will exhibit their own batch leaching curves. When these two curves differ widely, it can be said that the two ores are different. When the curves differ by only a slight margin, it could be said that the ores may be similar. Nevertheless, gold leaching curves are not unique to a specific gold ore, which means that a small difference cannot be considered as sufficient evidence that the two ores are the same. In order to complete the identification of an ore, the effects of various process variables on the leaching curves should be determined [20]. An ore can be characterized completely only when the effects of the major process variables are understood. In the database, the different ores are classified according to their basic mineralogy (i.e. refractory, sulphidic, oxidic, quartzitic, etc.). Therefore, if the type of mineralogy of the unknown ore has been established, the identification procedure becomes easier. The KBS is ideally suited for the inclusion and manipulation of diagnostic leaching data.

It is essential to set a sensitivity level for the comparison of the unknown ores with the known ones in the KBS [20]. The larger the number of known ores, the smaller the level of tolerance and the more accurate the identification process will be. It is clear that this
IDENTIFY option could be applied in a similar fashion to the characterization of activated carbon used in CIP/CIL, the flotation response of ores, or the size reduction and liberation behaviour of ores.

MODELS FOR CONTINUOUS FLOW REACTORS

The kinetic equations (1) to (6) can be incorporated into suitable process models which describe continuous flow systems. The difference between batch and continuous models lies in the material balance terms for flow into and out of the reactor, and the possible effect of non-ideal flow conditions. The simulation of continuous leaching, CIP or CIL processes will be given below as examples. Similar equations may be developed for the dynamic simulation of other mineral processing operations such as milling and flotation. As mentioned earlier, this deep knowledge can also be used in fault-diagnostics on continuous reactors.

The equations (7) to (9) that define the change of concentration in the solution, loading on the carbon and element content (gold in this case) in the ore were published previously [14,15]. Ideal flow is assumed in the formulation of these models, although in the KBS non-ideal flow in the form of short-circuiting \[C_{i-1} = \Phi C_{i-1} + (1-\Phi)C_{i-2}\] and dead volume \[V_{act} = V_{total} - V_{dead}\] is incorporated. The theoretical values of the dead volume and bypass streams can be obtained from tracer tests, or can be estimated by comparing real plant behaviour with that predicted from an ideal flow model.

\[
\frac{dC_i}{dt} = \left[\frac{v_{p,in}}{V_{act}}\right]C_{i-1} - \left[\frac{v_{p,out}}{V_{act}}\right]C_i + r_5 + r_3 \quad (7)
\]

\[
\frac{dy_i}{dt} = \left[\frac{m_{cc}}{M_{cc}}\right][y_{i+1} - y_i] + \left[\frac{m_{co}}{M_{co}}\right][y_{i-1} - y_i] + r_6 \quad (8)
\]

\[
\frac{dg_i}{dt} = \left[\frac{v_{p,in}}{V_{act}}\right]g_{i-1} - \left[\frac{v_{p,out}}{V_{act}}\right]g_i + r_4 \quad (9)
\]

Depending on \(m_{cc}\), \(M_{cc}\), \(m_{co}\) and \(M_{co}\) any type of carbon transfer mode, ranging from a low periodicity to continuous, co-current to counter-current or a combination of these may be simulated. Reuter and Van Deventer [14] showed how the same philosophy of integrating KBS and dynamic material balance equations could be used to simulate the adsorption of gold or silver cyanide on activated carbon in fixed beds or periodic countercurrent columns. It is clear that similar equations may be developed for the depletion of cyanide and oxygen in continuous flow systems [20], or for the simultaneous leaching and/or adsorption of different species such as gold and silver cyanide.

The material balance equations for the countercurrent leaching of gold ore in a Kamyr column were explained in a recent paper by Van der Merwe et al. [20], and will not be repeated here. However, the methodology of simulating Kamyr columns by KBS will be outlined briefly below.

In the Kamyr leaching column, the pulp enters at the top and leaves at the bottom, while the clear leaching solution enters at the bottom and leaves as clear solution at the top [20]. In modelling this operation, two distinct streams were defined, namely: (1) the pulp stream containing a mixture of leaching solution and solids, and (2) the clear leaching solution stream. As the leaching solution leaving the top of the column contains no solids, it is assumed that the solid phase exhibits only a downward flow. However, the liquid phase can move in either direction. It is therefore possible that the entering pulp can have a lower specific gravity than the pulp leaving at the bottom, or vice versa. By viewing the column as a series of non-ideal reactors, each containing a CSTR unit, it is possible to simulate a complex flow pattern. As explained by Van der Merwe et al. [20], a dead volume and bypass could be used on each of these reactor stages to cater for non-ideal flow conditions.

In order to simplify the computational procedure, it could be assumed that any change in the specific gravity follows a linear profile through the length of the reactor. When the number of mixed reactor stages is known, the change in the liquid fraction for each reactor
stage can be calculated. Under normal operating conditions, only the inlet conditions at opposite ends of the column are known. This causes some difficulty in the numerical work, because the material balance equations require values at both sides of the column. An iterative procedure could be followed whereby the outlet conditions for the pulp stream at the bottom are guessed initially. The material balances for gold in both solution and solid phases, as well as those for oxygen and cyanide, are then solved stage-wise from the bottom to the top of the column. The calculated concentrations in the clear stream outlet at the top are then compared with the given values, and the procedure repeated until the material balances are satisfied over the entire column [20].

SIMULATION RESULTS

In this section, the results of only a few case studies on simulations of leaching, CIP and CIL will be shown. In view of the fact that most of these simulations are based on a large body of information regarding the behaviour of the system under consideration, it is not possible to provide here all background information required for reproducing the simulations. Hence, the following figures and tables are merely indicative of the versatility and practical capabilities of the proposed KBS model.

Batch leaching

The Mintek leaching model [22] (Equations 1 and 2) was employed here as a basis for determining the \( k[g(t)] \) values used to predict the values in columns (b) and (d) of Table 1. The values in column (c) were determined by increasing the rate in the Mintek rate equation by a factor of 2.857. Hence, the if–then rule produced from the adjustment-objects for the particular ore under consideration may be:

if Au-leaching
and pyritic ore
and particle size 75% -74 μm
and \([CN]=200 \text{ ppm}\)
and density=1460 kg/m\(^3\)
and temperature is 30°C
then slope factor=2.857

(The calculated results are shown in tabular form so as to facilitate a more accurate comparison of actual values and those simulated by the KBS.)

**TABLE 1** Data in columns (a) and (c) were used to calculate \( k[g(t)] \), which was used to estimate the ore grades [g/ton] in columns (b) and (d) respectively.

<table>
<thead>
<tr>
<th>TIME (h)</th>
<th>(a)</th>
<th>(b)</th>
<th>(c)</th>
<th>(d)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2.3</td>
<td>2.3</td>
<td>2.3</td>
<td>2.3</td>
</tr>
<tr>
<td>1</td>
<td>1.03</td>
<td>1.03</td>
<td>0.594</td>
<td>0.607</td>
</tr>
<tr>
<td>2</td>
<td>0.705</td>
<td>0.702</td>
<td>0.413</td>
<td>0.417</td>
</tr>
<tr>
<td>4</td>
<td>0.476</td>
<td>0.474</td>
<td>0.308</td>
<td>0.309</td>
</tr>
<tr>
<td>8</td>
<td>0.334</td>
<td>0.334</td>
<td>0.251</td>
<td>0.251</td>
</tr>
<tr>
<td>16</td>
<td>0.256</td>
<td>0.256</td>
<td>0.221</td>
<td>0.221</td>
</tr>
<tr>
<td>23</td>
<td>0.23</td>
<td>0.23</td>
<td>0.211</td>
<td>0.211</td>
</tr>
</tbody>
</table>
Leaching in a series of CSTR’s

The KBS was used to compute the leaching behaviour of 10 t/h of a refractory ore in a slurry containing 50 mass % solids fed to a cascade of 11 CSTR’s, each having a volume of 400 m³, a dead volume of 10%, and in which 10% of the slurry is short-circuiting. The sensitivity of this ore to changes in the level of oxygen during batch leaching at 150 g NaCN/t is illustrated in Figure 2. In Figures 2 to 4 the solid lines represent KBS predictions on the basis of the existing data base which takes all the various interactions into account.

![Figure 2](image1.png)

**Fig.2** Effect of oxygen on the batch leaching of an ore containing initially 3.4 g Au/t.

![Figure 3](image2.png)

**Fig.3** Effect of initial level of cyanide on the leaching of a gold ore in a series of CSTR’s.

Figure 3 illustrates the sensitivity of the cascade of CSTR’s to different initial levels of cyanide. The cyanide levels in the leaching or CIL sections are controlled by on-line analytical equipment on a number of gold plants. This causes not only less consumption of
cyanide, but also improved metallurgical results due to the significant effect of cyanide on adsorption in CIP. Hence, it will be most useful if a KBS could be used to predict the behaviour of the leaching or CIL plant at different levels of cyanide. Moreover, if the nature of the ore changes, the consumption of cyanide will be different. If sufficient batch data or plant data for similar ores are available, the KBS could be used to simulate plant behaviour at new conditions. Few conventional simulators are capable of performing such functions.

![Graph](image.png)

Fig.4 The dynamic response of leaching of a gold ore in a series of CSTR's to a step change in the flow rate.

The KBS can be used to simulate the dynamic response of the cascade of leaching reactors to changes in either the operating conditions or the characteristics of the feed. Figure 4 shows as an example the response of the ore grade to a step change in the feed rate from 3 to 10t/h for the first three reactors.

**Leaching in a countercurrent column**

The simulation of a continuous countercurrent column takes more CPU time than that of the simpler systems. In the example used in Figures 5 and 6, the column with a diameter of 5m and a volume of 400m³ was subdivided into 11 theoretical mixed reactor units, as described earlier. The flowrate of the same slurry used in Figures 2 to 4 was maintained at 10 ton of pulp per hour. As this type of reactor configuration is relatively new, little industrial data are available to support the KBS predictions in Figures 5 and 6. However, the trends appear to be realistic. In Figure 5, the ore grade is depleted as it moves down the column, while the concentration of cyanide decreases from the bottom to the top. Figure 6 illustrates the effect of a higher level of cyanide fed to the column.

**Batch adsorption**

Table 2 compares the loading-time adsorption data produced by the Dixon model [25] to those produced by the proposed KBS model. All entries in the (a) columns give the results produced by the given Dixon model (pulp density 1460 kg/m³, 1:1 mass ratio solids:liquid). All the predictions at other carbon masses and starting concentrations in the (b) columns are based on the concentration-time data of the entries in the 1(a) column only. \((\delta t=1\text{h})\) was used to define \(k[y(t)]\). This example illustrates the use of a fact (discussed earlier) during simulation.
**Simulation of gold extraction processes**

**Fig. 5** The steady-state profiles of gold in the ore and cyanide in solution through a countercurrent leaching column.

**Fig. 6** The steady-state profile of gold in the ore at different levels of cyanide through a countercurrent leaching column.

**Adsorption in a countercurrent CIP cascade**

Only facts and functions describing the process were used here to perform the simulation. The evolution of $k[y(t)]$ for this simulation was produced via Equations 5 and 6 from the data given by Nicol et al. [21] (p. 40). The following values were used in calculations: pulp density = 1460 g/l, mass % solids = 50 %, $[Au]_{av}$ = 0.8 ppm in the first stage throughout and 25 g carbon per litre slurry. These data are summarized in Table 3.

The computed values for $k[y(t)]$ could then be used to simulate a 5-stage CIP pilot plant as described by Nicol et al. [22]. The volume of one reactor stage was 400 l, the flow rate of pulp was 800 l/h, and the entire carbon inventory of each stage was transferred every 48 h.
In Table 4, the entries in columns (a) are measured process values given by Nicol et al. [22], the entries in columns (b) are predictions by the model suggested by Nicol et al. [22], and the entries in columns (c) are computed by the KBS model proposed here.

**TABLE 2** Concentration-time data corresponding to the loading data in column 1(a) were used to calculate $k'y(t)$, which are applied to predict the loadings in [ppm] on the carbon in all columns (b). Initial loadings $= 0$.

<table>
<thead>
<tr>
<th>$M_C$</th>
<th>$C_O$</th>
<th>0.5 g/l (30 ppm)</th>
<th>5 g/l (1 ppm)</th>
<th>15 g/l (1 ppm)</th>
<th>15 g/l (15 ppm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t$ (h)</td>
<td>1(a)</td>
<td>1(b)</td>
<td>2(a)</td>
<td>2(b)</td>
<td>3(a)</td>
</tr>
<tr>
<td>1</td>
<td>3628</td>
<td>3600</td>
<td>94</td>
<td>92</td>
<td>46.5</td>
</tr>
<tr>
<td>2</td>
<td>5909</td>
<td>5893</td>
<td>127</td>
<td>126</td>
<td>48.5</td>
</tr>
<tr>
<td>4</td>
<td>8427</td>
<td>8413</td>
<td>143</td>
<td>143</td>
<td>48.6</td>
</tr>
<tr>
<td>9</td>
<td>10400</td>
<td>10390</td>
<td>145</td>
<td>145</td>
<td>48.6</td>
</tr>
<tr>
<td>19</td>
<td>10844</td>
<td>10835</td>
<td>145</td>
<td>145</td>
<td>48.6</td>
</tr>
<tr>
<td>29</td>
<td>10861</td>
<td>10848</td>
<td>145</td>
<td>145</td>
<td>48.6</td>
</tr>
</tbody>
</table>

The amount of carbon predicted by the model proposed here i.e. 23.5 g/l, is much closer to the real process value of 25 g/l than the value of 16 g/l predicted by Nicol et al. [22]. Using the rules of the KBS model, the process conditions could be determined which produce the exact carbon concentration of 25 g/l. This could be expected in view of the fact that the database is an accurate representation of the real data.

**TABLE 3** Summary of the calculated $k'y(t)$] values according to equations 5 & 6, using the data of Nicol et al. [21]

<table>
<thead>
<tr>
<th>Loading Range [g/t]</th>
<th>Slope</th>
<th>Time [h]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00 to 4000</td>
<td>-3.42</td>
<td>0 to 50</td>
</tr>
<tr>
<td>4000 to 6804</td>
<td>-2.40</td>
<td>50 to 100</td>
</tr>
<tr>
<td>6804 to 8176</td>
<td>-1.17</td>
<td>100 to 150</td>
</tr>
<tr>
<td>8176 to 9168</td>
<td>-0.85</td>
<td>150 to 200</td>
</tr>
<tr>
<td>9168 to 9986</td>
<td>-0.70</td>
<td>200 to 250</td>
</tr>
<tr>
<td>9986 to 10307</td>
<td>-0.27</td>
<td>250 to 300</td>
</tr>
<tr>
<td>10307 to 10512</td>
<td>-0.18</td>
<td>300 to 350</td>
</tr>
<tr>
<td>10512 and larger</td>
<td>-0.00</td>
<td>350 and larger</td>
</tr>
</tbody>
</table>

Leaching and adsorption in a countercurrent CIL cascade

This simulation is an extension of the CIP simulation discussed above, with adsorption and leaching taking place simultaneously. As is the case in many CIL plants, the first few stages are used for leaching only, while the subsequent stages are then used for combined leaching and adsorption on activated carbon. The retention time in the three "pre-adsorption" leaching pachucas is 3 h/pachuca, while that in the subsequent five leaching/adsorption tanks amounts to 1 h/pachuca. The corresponding results are summarised in Table 5. It is clear that the simulated values agree closely with those given by Nicol et al. [22], although the KBS model is capable of taking considerably more factors and even heuristic information into account. Although not shown here, the model can
simulate various carbon transfer modes e.g. continuous, or continuous over a small fraction of the total operating time (periodic transfer). A simulation involving a continuous transfer of carbon was discussed by Reuter and Van Deventer [15].

**TABLE 4** Comparison of process data, predictions by Nicol et al. [22] and simulations by the KBS model.

<table>
<thead>
<tr>
<th>Stage</th>
<th>Solution (g/t)</th>
<th>Loading (g/t)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(a)</td>
<td>(b)</td>
</tr>
<tr>
<td>Feed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2.26</td>
<td>2.26</td>
</tr>
<tr>
<td>2</td>
<td>1.12</td>
<td>1.00</td>
</tr>
<tr>
<td>3</td>
<td>0.39</td>
<td>0.44</td>
</tr>
<tr>
<td>4</td>
<td>0.15</td>
<td>0.19</td>
</tr>
<tr>
<td>5</td>
<td>0.08</td>
<td>0.08</td>
</tr>
<tr>
<td>Mc</td>
<td>25g/l</td>
<td>16g/l</td>
</tr>
</tbody>
</table>

**TABLE 5** Comparison of predictions produced by the CIL-model developed by Nicol et al. [22] (column a) and the model proposed in this paper (column b).

<table>
<thead>
<tr>
<th>STAGE</th>
<th>SOLIDS (g/t)</th>
<th>SOLUTION (g/t)</th>
<th>CARBON (g/t)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(a)</td>
<td>(b)</td>
<td>(a)</td>
</tr>
<tr>
<td>Feed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2.3</td>
<td>2.30</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0.6</td>
<td>0.70</td>
<td>1.70</td>
</tr>
<tr>
<td>3</td>
<td>0.4</td>
<td>0.41</td>
<td>1.90</td>
</tr>
<tr>
<td>4</td>
<td>0.38</td>
<td>0.31</td>
<td>2.02</td>
</tr>
<tr>
<td>5</td>
<td>0.25</td>
<td>0.29</td>
<td>0.43</td>
</tr>
<tr>
<td>6</td>
<td>0.25</td>
<td>0.27</td>
<td>0.09</td>
</tr>
<tr>
<td>7</td>
<td>0.24</td>
<td>0.26</td>
<td>0.02</td>
</tr>
<tr>
<td>8</td>
<td>0.23</td>
<td>0.25</td>
<td>0.006</td>
</tr>
</tbody>
</table>

**CONCLUSIONS**

The KBS approach can be applied to ill-defined processes that cannot be simulated adequately by conventional phenomenological models. The flexibility of a KBS was combined with the dynamic simulation properties of a set of simple ordinary differential equations. Although the proposed KBS model has been applied to limited field data, it appears to give realistic simulations of leaching, CIP and CIL plants. It is clear that the KBS is especially useful to perform a sensitivity analysis and to answer "what-if" questions. The KBS can also be used to perform fault-diagnosis and process identification. This can be most valuable when designing new gold plants.

As expected, the system is heavily dependent on the accuracy and population density of the available data. By using sufficient data over a wide range of operating conditions, a
detailed understanding of the various interactions between the process variables is unnecessary. The user has total control over all shallow and deep knowledge presented in the form of facts, objects (frames), rules and functions. These define both physical and chemical conditions in the process units and may be edited via user-friendly menu-interfaces.

**LIST OF SYMBOLS**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>a, b, c, d, n</td>
<td>Empirical parameters:</td>
</tr>
<tr>
<td>C, C_i, C_n</td>
<td>Concentration in a batch reactor, concentration in stage i and data point n respectively [ppm].</td>
</tr>
<tr>
<td>d_l, d_o</td>
<td>Density of liquid and ore respectively [kg/m³].</td>
</tr>
<tr>
<td>g, g_i, g_n</td>
<td>Ore grade in a batch reactor, grade in stage i and grade data point n respectively [g/ton].</td>
</tr>
<tr>
<td>g*</td>
<td>Refractory gold in ore [g/ton].</td>
</tr>
<tr>
<td>k</td>
<td>Rate variable as a function of y(t) and g(t) [h⁻¹].</td>
</tr>
<tr>
<td>M_c, M_t, M_s</td>
<td>Mass concentration of carbon, liquid and solids respectively [g/l].</td>
</tr>
<tr>
<td>m_{cc}</td>
<td>Mass flow rate of the carbon counter-current and m_{co} for co-current flow [kg/h].</td>
</tr>
<tr>
<td>M_{cc,t}, t</td>
<td>Mass of carbon being transferred counter-current and M_{co,t} co-current [kg].</td>
</tr>
<tr>
<td>r</td>
<td>Rate equation [ppm/h].</td>
</tr>
<tr>
<td>t</td>
<td>Time variable [h].</td>
</tr>
<tr>
<td>V_{act}</td>
<td>Active stage volume [m³].</td>
</tr>
<tr>
<td>v_{p,in}, v_{p,out}</td>
<td>Pulp flow rate in and out of a stage [m³/h].</td>
</tr>
<tr>
<td>y, y_i, y_n</td>
<td>Carbon loading in a batch reactor, loading in stage i and loading data point n [ppm].</td>
</tr>
<tr>
<td>δt</td>
<td>Time increment.</td>
</tr>
<tr>
<td>Re, Sc, Sh</td>
<td>Reynolds, Sherwood and Schmidt numbers.</td>
</tr>
<tr>
<td>ε</td>
<td>Volumetric fraction of pulp that is liquid.</td>
</tr>
<tr>
<td>Φ</td>
<td>Fraction of pulp short-circuiting reactor stage.</td>
</tr>
</tbody>
</table>

**REFERENCES**


Knowledge-Based Simulation and Identification of Various Metallurgical Reactors

M.A. REUTER and J.S.J. VAN DEVENTER

A knowledge-based system (KBS) is discussed for the dynamic simulation and fault diagnosis of various metallurgical reactor systems. The fundamental basis (deep knowledge) of the model is a database containing concentration-time data and an accompanying generalized kinetic model. The knowledge base is defined by various facts, objects, rules, and functions, which capture both deep as well as shallow knowledge regarding the process. This approach permits the simulation and fault diagnosis or identification of metallurgical reactors at all reactor conditions catered to by the knowledge base. Various examples, which include adsorption, leaching, and reduction (pyrometallurgical) reactions, illustrate the various facets of the proposed knowledge-based simulation (batch and continuous), process identification, and knowledge representation.

1. INTRODUCTION

Research on the application of expert systems and artificial intelligence techniques in chemical engineering may be grouped into the following areas:

(1) the synthesis of chemical engineering flow sheets in hybrid-type systems (hybrid system; knowledge-based system (KBS) integrated with equation-orientated simulation programs);1,2,3,4
(2) selection programs, e.g., DECADE* for the selection of catalysts,5,6,7
(3) synthesis of operating procedures for chemical plants;8,9,10
(4) malfunction diagnosis for complete chemical plants;11,12
(5) control synthesis.13,14

Although different architectures are used to define the knowledge base, all cited examples apply both qualitative (experimental and heuristic) and quantitative (equation-orientated) knowledge, also termed shallow and deep knowledge, respectively. In these systems, data and knowledge are mostly represented as

(1) objects (a chunk of structured knowledge),11 which are also called frames,
(2) production rules, which are characterized by if-then type statements, and
(3) functions, which define the equations for quantitative simulation of unit operations.

These data structures form the basis of the various knowledge-based systems and are embedded in various architectures which include:

(1) object-orientated11,12 or frame-orientated13 systems,
(2) hybrid systems in a Blackboard (Blackboard is the global database permitting the interaction of all knowledge sources) architecture,14,15
(3) hierarchical diagnostic systems in which every node in the structure represents a malfunction hypothesis, the top node containing general malfunction information and vice versa (hence, the information is grouped into various levels of abstraction),16,17 or
(4) goal tree-success tree architecture used for the design of control systems for distillation columns.18

Of the possibilities listed, an object-oriented approach was applied in the program discussed here to define the structure of the knowledge base for the proposed diagnostic and simulation system for metallurgical reactors. Whereas most of these types of applications implement object-orientated languages (e.g., KEE or CommonLISP)19 for the programming of the qualitative model, Pascal, which permits the definition of complex data structures and performance of numerical analysis in the same program, was applied. This makes the time-consuming communication between two different programs, i.e., one to perform the knowledge engineering and the other the numerical simulation, unnecessary.

Most of the systems that perform malfunction diagnosis attempt to diagnose malfunction on a macroscopic level, i.e., analyze flow rates, temperatures, etc., and suggest appropriate remedial action. None of the discussed systems, however, attempt to analyze the environment within a particular unit operation and suggest appropriate action on the basis of the basic parameters that characterize the process under consideration. This weakness in the cited diagnostic systems, as well as their inability to embrace dynamics, is addressed in this article. There are, however, qualitative modeling approaches, e.g., the QSIM algorithm, which embrace dynamics and can be included in model-based reasoning systems20 but these have not been applied extensively.

In order to be able to define a general knowledge base for the proposed system, a general fundamental basis had to be defined, as was done by Tzouanas et al.,21 who applied basic control theory and accompanying transfer functions to define the deep knowledge for the control of a distillation column. The numerous kinetic models

M.A. REUTER, formerly Senior Lecturer, University of Stellenbosch, is Research Metallurgist with the Institute for Non-Ferrous Process Metallurgy, Aachen University of Technology, 5100 Aachen, Federal Republic of Germany, J.S.J. VAN DEVENTER, Professor, is with the Department of Metallurgical Engineering, University of Stellenbosch, Stellenbosch 7600, Republic of South Africa.

available to describe various metallurgical reactions render the definition of such a generalized system hardly possible, as the number of parameters among these models vary. This necessitated the development of a generalized kinetic model, which permits the definition of a generalized fundamental basis. The subsequent structure of the knowledge base was based on the single parameter of this model (determined directly from concentration-time data of the process under consideration) and the equilibrium recovery, loading, or conversion (depending on the type of reaction). The effects of all possible process parameters on this single parameter were defined as objects (which define both shallow and deep knowledge), rendering the model totally general. This is in contrast to most existing models which lump all effects into a number of parameters, which make these kinetic models totally process and chemical environment specific. Hence, the existing models are usually incapable of modeling the effects of chemical process conditions, e.g., pH, ionic strength, carbon activity, etc., and hence performing fault diagnosis.

The basis of the model (deep knowledge) is formed by the data base which contains concentration-time data, e.g., adsorption data and leaching data. This basis permits (1) the formulation of a process-independent kinetic model, (2) the definition of a process-independent knowledge base which defines objects and rules that cover kinetic, physical, and economic aspects, and (3) dynamic fault diagnosis by the interactive use of the above-mentioned kinetic model, objects, and rules by comparing dynamic simulation results with production goals. Hence, the model that is proposed here may be compared to a hypothesis-and-test type approach, as discussed by Kramer and Moore.

The shallow knowledge (experiential and heuristic) in the proposed model is defined as objects, and the deep knowledge is defined by the proposed kinetic model. How this knowledge is defined with regard to the proposed model will subsequently be discussed with reference to various metallurgical batch and continuous reaction systems.

II. FORMULATION OF MODELS

Techniques of knowledge (shallow and deep) representation form an important aspect of a knowledge-based system. As discussed by Tzouanas et al., these techniques may include facts, objects (frames), rules, and functions. As stated above, the basis of the knowledge representation discussed in this article is the concentration-time data of the metallurgical reaction being considered. This represents the deep knowledge regarding the process; the remainder of knowledge is structured in such a way that it has access to these data or transformations of the data to perform reactor simulation and interactive fault diagnosis. The above-mentioned techniques of knowledge representation are applied as follows in order to define the knowledge base.

Facts may include, for example, the type of reaction under consideration, be it leaching, adsorption or reduction, or the type of reactor system.

Each object (frame) is structured in order to include all relevant information regarding a particular process variable and its effect on the process (see adjustment object in Section III-B-1), or information regarding a particular unit operation, e.g.:

Unit operation object for a carbon-in-pulp (CIP) plant

<table>
<thead>
<tr>
<th>UNIT OPERATION</th>
<th>continuous reactors</th>
</tr>
</thead>
<tbody>
<tr>
<td>MASS OF ADSORBENT IN REACTOR</td>
<td>22 g/l</td>
</tr>
<tr>
<td>FEED CONCENTRATION</td>
<td>2.02 ppm</td>
</tr>
<tr>
<td>PULP DENSITY</td>
<td>1460 kg/m³</td>
</tr>
<tr>
<td>VOLUME OF THE REACTOR</td>
<td>400 l</td>
</tr>
<tr>
<td>PCT ADSORBENT LEAKING COCURRENT</td>
<td>0</td>
</tr>
<tr>
<td>NUMBER OF REACTORS IN SERIES</td>
<td>5</td>
</tr>
<tr>
<td>TIME ONE OPERATING CYCLE [Ts]</td>
<td>48 h</td>
</tr>
<tr>
<td>NUMBER OF OPERATING CYCLES</td>
<td>5</td>
</tr>
<tr>
<td>INITIAL ADSORBENT LOADING</td>
<td>0 ppm</td>
</tr>
<tr>
<td>PCT OF ADSORBENT TRANSFERRED</td>
<td>100 pct</td>
</tr>
<tr>
<td>FEED FLOW RATE OF THE PULP</td>
<td>400 l/h</td>
</tr>
<tr>
<td>PCT SHORT CIRCUITING OF FEED</td>
<td>0</td>
</tr>
<tr>
<td>PCT DEAD VOLUME OF REACTOR</td>
<td>0</td>
</tr>
<tr>
<td>PCT OF Ts USED FOR TRANSFER</td>
<td>0</td>
</tr>
</tbody>
</table>

This object was used to simulate the CIP plant discussed in Section III-D-1. Although given here for CIP plant, its generality permits the simulation of any similar type of adsorption process, such as resin-in-pulp for example. Another possible object is the cost object used for economic evaluation.

The rules (production rules) defined here fall into various categories which include the following:

Adjustment rules (process specific)

if NiO reduction by H₂ and porosity is 0.728 and temperature is 299 °C and pellet radius is ±0.77 cm then slope factor = Φ = 1.46 load factor = Ω = 1 cost factor = 1

Search rules

For a cobalt ore from a particular ore body from the Blackbird mine, for example, the chemical environment which produces a particular recovery and slope factor can be formulated as

if slope factor ≤ 1.12 and slope factor ≥ 1.0 and Co recovery is ≥ 94 pct and Co recovery is ≤ 96 pct
By controlling the tolerances on the slope factor and the CO recovery, the user has control over the search accuracy of this type of rule.

**Procedural rules**

As this model is an off-line system, these rules are addressed manually at this stage:

if leaching reaction
and train of continuous reactors
then use CSTR procedure
*continuous stirred tank reactor

Functions include the kinetic models, differential equations that define the nonideal flow through the continuous reactors, or equations that describe fundamental aspects of the process, if applicable. For example, the Arrhenius equation. A function would hence contain the following elements:

1. **Type of operation (defining label)**
2. **Kinetic model (differential equation)**
3. **Unit operation object (as above)**
4. **Nonideal flow model (differential equation)**
5. **Regression adjustment objects (as in Section B)**
6. **Fundamental correlations**

This knowledge representation is integrated in a user-friendly simulation program which is written in TURBO PASCAL.* Figures 1 and 2 present flow diagrams of the program, which will be discussed in detail below. Since the concentration-time format of the deep knowledge and the accompanying kinetic model form the basis of the two integrated activities, namely, dynamic simulation and dynamic fault diagnosis or process identification, this will be discussed first. The simulation and fault diagnostic (conditions within reactor) or process identification activities will be discussed subsequently.

**A. Kinetic Models**

A general process-independent knowledge base requires a process-independent kinetic model. Such a general kinetic model has been formulated for CIP and carbon-in-leach (CIL) processes by Reuter and van Deventer, which has subsequently been extended to various metallurgical and other mineral processing (e.g., flotation) processes.

The kinetic models which define the deep knowledge of adsorption, reduction, and leaching reactions will be developed below and extended to continuous processes. Although the three models have been defined separately, they are, in fact, identical, the adsorption model being the most general and the other two simplifications thereof. It must be noted here that these models do not attempt to suggest a mechanism for the reaction under consideration. What they do state, however, is that a kinetic curve which describes a process up to equilibrium is a fingerprint of that reaction and, therefore, of the reaction mechanism. Consequently, the model can be used as a basis to describe the reaction under all possible conditions, given that the mechanism does not change. The data base will obviously cater to all possible mechanisms for a particular system. (See Section III-C-3.) This basic fingerprint curve and its associated shallow and deep level knowledge, defined as objects, are termed the pivot data.

These pivot data serve as a reference to which other curves are compared, which can be used for fault diagnosis or process identification, as depicted in Figure 2. See Section III–E for a discussion of these aspects of the model.

---

*TURBO PASCAL is a trademark of Borland International, Scotts Valley, CA.*

---

![Flow diagram for the knowledge-based simulation aspects of the program.](https://scholar.sun.ac.za)
1. Adsorption model

As mentioned above, the basis of the model is an adsorption concentration-time and corresponding loading-time curve that covers the adsorption kinetics up to the equilibrium loading capacity. From this curve and the corresponding concentration-time, the rate variable \( k(y(t)) \), which is defined here to be only a function of the loading \( y(t) \), may be determined. This rate variable is subsequently used to predict the change in concentration and carbon loading at any carbon concentration \( M_C \), any solution density, and any starting concentrations at the prevailing chemical process conditions.

\[
\frac{dC}{dt} = -\Phi \cdot \left( \frac{\alpha_1}{\alpha_2} \right) \cdot k(y(t)) \cdot C = r_1 \tag{1}
\]

\[
\frac{dy}{dt} = \Phi \cdot \alpha_1 \cdot k(y(t)) \cdot C = r_2 \tag{2}
\]

where \( k(y(t)) \) = \(-2(C_{\ast,1} - C_C)/\delta(C_{\ast,1} + C_C)\) \( \alpha_1 = (M_s/M_C) \) and \( \alpha_2 = (M_s/M_C) \), and \( \Omega \cdot y(t) \leq y(\Omega) \leq \Omega \cdot y(\Omega) \).

The rate variable \( k(y(t)) \) is also a function of the knowledge base via objects; i.e., it is adjusted with reference to the pivot data, permitting modeling at chemical and physical process conditions other than the pivot data.

2. Leaching model

As for the adsorption model, \( k(g(t)) \), which is defined here to be only a function of the grade \( g(t) \), is determined from a leaching curve, which covers the leaching up to the equilibrium leaching point or until complete dissolution of the valuable element has occurred. As for the adsorption model, \( k(g(t)) \) is also a function of the knowledge base via objects.

\[
\frac{dC}{dt} = \Phi \cdot k(g(t)) \cdot \delta g(t) = r_3 \tag{3}
\]

\[
\frac{dg}{dt} = -\Phi \cdot \beta \cdot k(g(t)) \cdot r_4 \tag{4}
\]

where \( k(g(t)) = 2(C_{g,1} - C_g)/\delta(g_{g,1} + C_g) \) and \( \beta = (M_s/M_g) \).

For \( g_{g,1} \leq g(t) \leq g_{g,1} \), and

In the case of deviation from the final recovery as given by the pivot conditions, the above equations are adjusted to

\[
\frac{dC}{dt} = \Phi \cdot k(g(t)) \cdot \delta(g(t) - g^\ast_{g,1} (1 - \Omega))
\]
which holds for
\[ g_{\alpha} - \Omega \cdot [g_{\alpha}^* - g_{\alpha}(t)] \leq g(t) \leq g_{\alpha} - \Omega \cdot [g_{\alpha}^* - g_{\alpha}(t)] \]

3. Carbon-in-leach model
As stated by Nicol et al.,[19] the adsorption and leaching reactions are independent of each other. Hence, the CR model proposed here is a linear combination of the above-mentioned two models.[27,28]

4. Reduction reactions
As in the case for leaching reactions, the rate \( k(r(t)) \), which is defined here to be only a function of the element being reduced, \( r(t) \), is determined from a reduction curve, which covers the reduction up to the point at which equilibrium has been reached or complete reduction has taken place.

\[
\frac{dC}{dt} = \Phi \cdot k(r(t)) \cdot r(t) = r_h \quad [5]
\]
\[
\frac{dr}{dt} = -\Phi \cdot k(r(t)) \cdot r(t) = r_n \quad [6]
\]

where \( k(r(t)) \cdot r(t) = 2(C_{n(t)} - C_d)/\delta(r_{x(t)} + r_n) \)

holds for \( r x(t) \leq r(t) \leq r_n(t) \).

A similar final recovery adjustment equation that holds for leaching also holds for reduction.

B. Knowledge-Based Model
The overall integration of the kinetic model and the various objects to perform the simulation and diagnostic activities are depicted by Figures 1 and 2, respectively.

1. Simulation
The data may be divided into both shallow and deep knowledge. The shallow knowledge is given in the form of qualitative (heuristic) data regarding the process. Associated with this qualitative (heuristic) data is a set of properties, kinetic data and an appropriate kinetic model. This, as well as equations which define various physical properties, e.g., the Arrhenius equation, present the deep knowledge of the model, the basis on which fault diagnosis or process identification is performed.

The shallow and deep knowledge for a particular reaction system is addressed initially by giving an appropriate input, as may be seen in Figure 1. The input to the model comprises initially of a primary input regarding the type of reaction under consideration, followed by a secondary input which could be a special characteristic of that reaction or system. Hence, possible input could be (1) NiO reduction with H₂ followed by selecting (2) the pellet porosity. This selection subsequently extracts the following knowledge of the process from the knowledge base.

a. Shallow knowledge
(1) Objects, which are each defined by a describing label, e.g., temperature, etc., are applied to capture this type of knowledge. Associated with each object are: adjustment factors for the rate, final attainable conversion, and costs; and interaction factors which reflect any interaction which may exist between the heuristic data. These interaction factors reflect the effect of the \( n - 1 \) other adjustment objects on the considered object. Hence, the \( n - 1 \) factors below (third item in object below) reflect no interaction with other items (=1). One adjustment object of \( n \) possible objects with \( m \) subdivisions is defined as follows:

<table>
<thead>
<tr>
<th>Adjustment Object Label</th>
<th>Temperature</th>
</tr>
</thead>
<tbody>
<tr>
<td>n - 1 Interaction Factors</td>
<td>( 299 ) °C</td>
</tr>
<tr>
<td>Load Factor</td>
<td>( \Phi = 0.96 )</td>
</tr>
<tr>
<td>Cost Factor</td>
<td>( \Omega = 1 )</td>
</tr>
</tbody>
</table>

(2) Rules, which permit the extrapolation of knowledge from existing heuristic knowledge, if it is not explicitly available.

(3) The \( k \) objects, also defined by a label, but correlating heuristic as well as fundamental aspects on a statistical basis and not by factors as shown above, are given below. Hence, similar to the above object, a regression adjustment object may be defined as:

<table>
<thead>
<tr>
<th>Adjustment Object</th>
<th>Temperature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>temperature</td>
</tr>
<tr>
<td>Valid</td>
<td>( 50 ) °C to ( 70 ) °C</td>
</tr>
<tr>
<td>k-1 Interaction Factors</td>
<td>( a + bt + cT^2 + dT^{1.5} + iT^3 )</td>
</tr>
<tr>
<td>Load Factor</td>
<td>( \Omega = 1 )</td>
</tr>
<tr>
<td>Cost Factor</td>
<td>( a + bt + cT^2 + dT^{1.5} + iT^3 )</td>
</tr>
</tbody>
</table>

The coefficients of the given regression equation are determined by the appropriate statistical methods, which may be accessed by suitable menu options in the program. The user has a choice of which type of the above two objects he wishes to use. The first may be used to define knowledge which is difficult to correlate, e.g., rules of thumb of the considered plant, and the second which permits a semitheoretical approach. By the appropriate selection of a set of adjustment objects which correspond to the operating conditions of the plant, the final adjustment factors are obtained with reference to the pivot data via if-then rules.

b. Deep knowledge
(1) Kinetic data for all important species that have an influence on the kinetics of the process. These data are regarded here to be the key to fault diagnosis and a very simple simulation strategy. Typical kinetic data for the leaching of gold, for example, could include changes in [Au], [Ag], [Cu], [CN⁻], and pH.

(2) Equations which define various fundamental aspects of the process, e.g., the Arrhenius equation (if it is possible to define the temperature dependence of the given reaction by this equation) or by the application of the above-mentioned objects.

(3) Regression equations which correlate fundamental aspects of the process on a statistical or semitheoretical basis, e.g., mass-transfer correlations.

\[
Sh = a(Re)^b(SCe)^c + d
\]
This equation is included as an subobject in the simulation functions, adjusting the rate constant with reference to the Sherwood (Sh) number of the pivot data. It would also be clear that the following holds:

\[ \Phi = f(\text{Sh})/f(\text{Sh})_{\text{prox}} \]

where \( k = f(\text{Sh}) \).

(4) The influence of the secondary components on the leaching of the principal element is defined by additional regression adjustment objects. Consider, for example, the leaching of gold during which, among others, the cyanide concentration changes. The kinetic models for both gold and cyanide are solved simultaneously, and via a regression adjustment object for cyanide, \( k[\text{g}(t)] \) for gold is adjusted continuously.

(5) Differential equations that define the appropriate continuous unit operations, covering both gold and cyanide are solved simultaneously, and for process simulation of a particular example above.

The application of the deep and shallow knowledge for process simulation of a particular metallurgical unit operation would typically proceed through the following steps:

(1) A particular input made to the model selects the type of metallurgical reaction and reactor system being considered.

(2) A set of kinetic data is extracted from the data bank, smoothed by an appropriate numerical method, e.g., a spline, transformed into slope-load data (see above kinetic models) and adjusted according to the rules extracted from the knowledge base.

(3) The corresponding knowledge (shallow and deep) are extracted from the knowledge base.

(4) These adjusted slope-load data, as well as physical process variables, are then used to perform process simulation.

(5) Suitable regression adjustment objects of the data bank continually adjust the kinetic model as the simulation proceeds as a function of changes which may occur during the process simulation, as discussed for Au and CN above.

(6) An overall economic evaluation is finally performed and the respective simulation data retained as shallow knowledge for future application during first-level fault diagnosis. This knowledge may also be termed experiential knowledge.

To summarize the above, a pivot object is given that defines both shallow and deep knowledge regarding the process. The given object refers to the example discussed in Section III-C-3:

**TYPE OF REACTION**

**SYSTEM** : NiO reduction by H₂

**DISTINGUISHING CHARACTERISTIC** : porosity (0.728)

**n FACTOR ADJUSTMENT OBJECTS** : none defined

**k REGRESSION ADJUSTMENT OBJECTS** : temperature (e.g.), Arrhenius equation

**OBJECTS** : pellet size (1/r), etc.

**KINETIC DATA (1 sets possible per reaction system)** :

- Time conversion
  - 0
  - 2.5 0.69

**COST DATA** :

- function of type of operation (none in this case)

All other adjustment objects would refer to this pivot object (all adjustment factors = 1) for process simulation and process identification. The adjustment objects used in this pivot object are labeled pivot adjustment objects. All objects and physical data may be edited by the user.

2. Diagnostic functions

A flow diagram, which depicts the routes followed to analyze the process and perform fault diagnosis (conditions within reactor) is given in Figure 2. This diagnosis and analysis is affected by a search (backward chaining) through both shallow and deep knowledge subject to if-then rules and a comparison to the pivot data. This procedure involves the use of both shallow and deep knowledge, either separately or simultaneously.

- **Option 1**

The application of shallow knowledge for fault diagnosis or process identification is discussed under this heading.

This search is initiated by inputting a set of kinetic data, which is transformed to produce two values, viz., time-weighted sum of slopes and the final recovery (equilibrium loading or final reduction). These are compared to the available shallow knowledge, i.e., time-weighted slopes and final recoveries of all other applicable pivot objects and pivot data to predict, for example, the ore characteristics and hence characterize the ore. The mathematical basis for this may be found in the following integral:

\[ \int_0^t k' \cdot C \cdot \frac{\text{dt}}{} = C_0 \int_0^t k' \cdot e^{-k' \cdot t} \cdot \frac{\text{dt}}{} = C_0/k' \]  

Hence, the sum of time-weighted slopes for a first-order process produces the result \( C_0/k' \). This integral has been generalized here for the discrete \( k(r(t))_n \) values to (Euler integration of Eq. [7]):

\[ C_0/k'' = C_0 \cdot \sum \frac{t_n \cdot k(r(t))_n \cdot r(t)_n \cdot \delta t}{n} \]

This equation forms the basis for fault diagnosis in batch reactors.

- **Option 2**

If the types of reaction, ore, and reaction system, for example, leaching in pachucas, are known, the model may perform fault diagnosis or process identification of the system under consideration. This is affected by the simultaneous application of shallow as well as deep process knowledge on an interactive basis, e.g., the Au and CN example above. It typically involves the following activities:
Kinetic concentration-time data of the process to be diagnosed are entered. This is transformed into time-weighted slope and final recovery data. All shallow and deep knowledge is consulted for the system under consideration to establish which adjustment objects could produce the kinetic data being considered via if-then rules. A comparison to the pivot data for the process under consideration can establish which process variables deviate from the pivot data and hence diagnose faults.

**Continuous systems**

A similar procedure is followed as that given for the fault diagnosis of batch systems. The difference lies in that a continuous process model such as that defined in Section C is applied. Hence, given a particular recovery of the continuous process to be diagnosed, the model consults both shallow as well as deep knowledge to determine which process conditions could produce the said recovery. This is compared to the pivot data for the process from which deviations may be established and hence process identification.

These results of the fault diagnosis are retained, if so required, to perform the shallow-level diagnosis discussed in option 1, hence constituting the learning process of the model.

It is clear from the above that the following diagnostic exercises may be conducted:

1. Ore, slag, or adsorbent identification on the basis of leaching, reduction, and adsorbent kinetic data.
2. Process fault diagnosis for both continuous and batch processes on the basis of kinetic data and/or final recovery by interactive consultation of the shallow and deep process knowledge and comparison to the pivot data.

**C. Continuous Process Models**

The above-mentioned kinetic models may be included into suitable process models which describe continuous systems. The simulation of continuous CIL processes will be given as an example, and hence, only these equations will be discussed. Similar equations may be developed for slag-reduction reactions in continuous electric furnaces, for example. This deep knowledge is used to perform fault diagnosis in continuous reactors, as discussed previously.

**Continuous models for CIL systems**

The equations that define the change of concentration in the solution, loading on the carbon, and element content in the ore (Eqs. [9] through [11], respectively) will be given subsequently (see List of Symbols for the meaning of the symbols). Ideal flow is assumed in the formulation of these models (Eqs. [9] through [11]), although in the KBS, nonideal flow in the form of short-circuiting (\( \text{mass net}_i = \alpha \text{mass}_i + (1 - \alpha) \text{mass}_i \)) and dead volume (\( \text{volume}_i = \text{volume}_i - \text{volume}_i \)) is formulated.\(^{127,28}\)

\[
\frac{dC_i}{dt} = \frac{V_p}{V_{act}} C_{i-1} - \frac{V_p}{V_{act}} C_i + r_i + r_3 \tag{9}
\]

\[
\frac{dy}{dt} = \frac{m_r}{M_{r_j}} (y_{i+1} - y_i) + \frac{m_c}{M_{c_j}} (y_{i-1} - y_i) + r_2 \tag{10}
\]

\[
\frac{dg_i}{dt} = \frac{v_p}{V_{act}} g_{i-1} - \frac{v_p}{V_{act}} g_i + r_i \tag{11}
\]

Depending on \( m_r, M_r, m_c, M_c \), any type of carbon transfer mode, ranging from batch to continuous, countercurrent to cocurrent, or a combination of these, may be simulated.

**III. VALIDATION OF THE MODELS AND DISCUSSION OF RESULTS**

The defined kinetic model forms the basis of the simulation and fault diagnosis or process identification. To validate the model, various reactions and processes have been simulated as a function of the shallow and deep knowledge of the knowledge base. The results have been compared to results produced by well-established models or published industrial data. Subsequently, these kinetic models have been applied to simulate continuous processes. Finally, the application of these models during fault diagnosis will be discussed. The models are solved by Runge-Kutta methods.\(^{130}\)

**A. Adsorption Model**

This model has already been validated in a previous article.\(^{129}\) The performance of the adsorption model was compared in that article to the well-established Dixon Model.\(^{129}\) Table I compares the loading-time data produced by the Dixon Model to that produced by the proposed model. All entries in the (a) column give the results produced by the given Dixon Model (pulp density 1460 kg/m\(^3\), 1 : 1 mass ratio solids : liquid). All the predictions at other carbon masses and starting concentrations of the entries in the (a) column, which covers the adsorption up to the maximum equilibrium loading, i.e., \( \pm 3541 \) ppm was taken to define \( k(l) \). Only a few points are shown. This example illustrates the use of a fact during simulation.

From column 1(a) in Table I, it would be clear that for the first two and last entries, the following values for \( k(y(t)) \) may be calculated if \( \Delta t = 1 \) h:

\[
0 \leq y(t) < 3248 \quad k(y(t)) = 0.844
\]

\[
3248 \leq y(t) < 3449 \quad k(y(t)) = 0.094
\]

\[
y(t) \geq 3541 \quad k(y(t)) = 0
\]

\(*3248 \text{ppm} = (30 - 12.2) \cdot 0.73/4000 \cdot 10^6\)

The results (given in table form in order to illustrate the accuracy) compare favorably with the results of the Dixon Model.\(^{129}\) Similar simulations may be performed for adsorption onto ion exchange resins.

**B. Leaching Models**

A variety of leaching data from the literature will be used to illustrate the applicability of the proposed kinetic
The first and second examples compare the performance of the proposed model to two published shrinking core models.\(^{111,151}\) The third example is an approximation of data given by Nicol et al.\(^ {110}\) The fourth example uses experimental data to predict kinetic data and appropriate adjustment factors under a variety of operating conditions.

1. Leaching of ZnS in aqueous FeCl\(_3\) solution

Rath et al.\(^ {131}\) showed that the kinetics of the reduction of ZnS in FeCl\(_3\) solution follows a diffusion-controlled shrinking core model. Their derived reaction rate constants were used to determine the kinetic data which are given in columns (a), (c), and (e) of Table II. The data in column (a) was used to determine \(k[g(t)]\) and produce the results presented in column (b). The entries in columns (d) and (f) are produced by adjusting \(k[g(t)]\) according to the reaction rates given by Rath et al.\(^ {131}\) From their given rate constants, appropriate adjustment factors could be determined with reference to the data in column (a), viz., 0.328 and 0.0467 for 50 °C and 30 °C, respectively; hence, the rule for the adjustment factor used to predict the values in column (d) would have the following form:

\[ \text{if } \text{ZnS leaching by FeCl}_3 \]
\[ \text{and distinguishing characteristic } \]
\[ \text{and particle size } 100 \text{ pct } - 75 \mu m \]
\[ \text{and temperature is } 50 \degree C \]

then \[ slope \text{ factor } = \Phi = 0.328 \]
\[ load \text{ factor } = \Omega = 1 \]
\[ cost \text{ factor } = 1 \]

This if-then rule could be constructed from the following temperature adjustment-object which holds for the ZnS leaching system characterized by a particles size of 100 pct - 75 \( \mu m \) (pivot data 60 °C):

- **Adjustment Object Label**: temperature
- **Subdivision of Adjustment Object**: \( n - 1 \) interaction factors: 1
- **Slope Factor**: \( \Phi = 0.328 \)
- **Load Factor**: \( \Omega = 1 \)
- **Cost Factor**: 1

2. Leaching of chalcopyrite at elevated temperature

Yu et al.\(^ {131}\) showed that the leaching of chalcopyrite at elevated temperature can be described by a reaction-controlled shrinking core model. They showed that the linear rate constant may be defined by a Langmuir-type isotherm, which is a function of both temperature and oxygen pressure. These data could be used to define rules for both temperature and pressure. If the data in column (a) are used as a basis, inclusive of operating conditions (forming the pivot data), the adjustment factors in the form of pressure and temperature rules were 0.872 and 2.225 for columns (d) and (f), respectively. The data in columns (a), (c), and (e) were determined using the proposed model and kinetic data of Yu et al.\(^ {131}\) (see Table II).\(^ {131}\) A regression equation describing the \((1/r_0)\)
dependence of particle size on the reaction rate could also be defined, although not required.

From Table III, it is clear that the pivot data for this example would include the adjustment objects for pressure and temperature and the concentration-time data for column (a). The regression adjustment object that describes the pressure dependence could be defined by

**ADJUSTMENT OBJECT**

**LABEL:** pressure

**g = 1 INTERACTION FACTORS**: 1

**VALID RANGE**: 75 ≤ \( p_{00} \) ≤ 400 psi

**SLOPE FACTOR**: \( \Phi = A \cdot p_{00} / (1 + B \cdot p_{00}) \)

**LOAD FACTOR**: \( \Omega = 1 \)

**COST FACTOR**: 1

The \( A \) and \( B \) values in the above regression object were taken from the referenced text (Table II), with \( A \) adjusted in order to normalize the object with reference to the pivot object.

3. **Leaching of Gold**

The Mintek leaching model was used here as a basis for determining the \( k [g(t)] \) values used to predict the values in columns (b) and (d) of Table IV. The values in column (a) were determined by the given Mintek rate equation, while the values in column (c) were produced by increasing the rate by a factor 2.85 by an appropriate adjustment rule. Hence, the if-then rule produced from the adjustment objects for the particular ore under consideration could be

- if Au leaching
- and pyritic ore
- and particle size 75 pct - 74 \( \mu m \)
- and \([CN] = 200 \) ppm
- and density = 1460 kg/m\(^3\)
- and temperature = 30 °C

then slope factor = \( \Phi = 2.85 \)

load factor = \( \Omega = 1 \)

cost factor = 1

4. **Leaching of Blackbird cobalt concentrate**

The data given by Harris et al.\(^\text{[33]}\) were used in this example to illustrate the use of a number of rules to predict the kinetic data. The data in column (a) of Table V were used as a basis for subsequent predictions given in columns (d) and (f). The data in column (a) correspond to a leach of the Blackbird cobalt concentrate performed at 1200 kPa, 150 °C, 30 pct solids, and 46 g/l Na\(_2\)SO\(_4\), which are the pivot data with a reference slope factor of 1.

From the given data rules which include \( O_2 \) pressure, temperature and level of Na\(_2\)SO\(_4\) could be defined. The adjustment factor for the rate the decrease in temperature from 150 °C to 140 °C could be determined from the given kinetic data to be 0.666, for the decrease of Na\(_2\)SO\(_4\) from 46 to 20 g/l an increase of 1.1, and for the decrease in operating pressure from 1200 to 860 kPa as 0.1. These adjustment factors were obtained directly from the given data with reference to the pivot adjustment objects, which were applied during the definition of the following if-then rule for this process:

- if leaching of Blackbird concentrate
- and distinguishing characteristic
- and \( O_2 \) pressure = 1200 kPa
- and temperature = 150 °C
- and 30 pct solids
- and 46 g/l Na\(_2\)SO\(_4\)

then slope factor = \( \Phi = 1 \)

load factor = \( \Omega = 1 \)

cost factor = 1

For the case for which the Na\(_2\)SO\(_4\) concentration decreases to 20 g/l, the adjustment rule would be

- if leaching of Blackbird concentrate
- and distinguishing characteristic
- and \( O_2 \) pressure = 1200 kPa
- and temperature = 150 °C
- and 30 pct solids
- and 20 g/l Na\(_2\)O\(_2\)

then slope factor = \( \Phi = 1.1 \)

load factor = \( \Omega = 1 \)

cost factor = 1

This rule will later be applied to demonstrate the fault diagnostic activities.

C. **Pyrometallurgical Reactions**

As an illustration, three reactions are discussed as a function of various data in the knowledge base.

1. **Ilmenite reduction with graphite**

El-Guindy and Davenport\(^\text{[34]}\) proposed a diffusion-controlled shrinking core model to describe the reduction of ilmenite to Fe and TiO\(_2\) by carbon. The activation energy of this reaction was calculated to be 268 kJ/mol between 1075 °C and 1140 °C. From the data given, a reaction rate of 0.00115 min\(^{-1}\) could be estimated for a

---

**Table III. Data in Column (a) Used to Calculate k[g(t)]**

<table>
<thead>
<tr>
<th>Time (min)</th>
<th>325 psia 160 °C</th>
<th>250 psia 160 °C</th>
<th>250 psia 175 °C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(a)</td>
<td>(b)</td>
<td>(c)</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>30</td>
<td>0.388</td>
<td>0.388</td>
<td>0.345</td>
</tr>
<tr>
<td>60</td>
<td>0.660</td>
<td>0.660</td>
<td>0.600</td>
</tr>
<tr>
<td>90</td>
<td>0.836</td>
<td>0.836</td>
<td>0.778</td>
</tr>
<tr>
<td>120</td>
<td>0.938</td>
<td>0.938</td>
<td>0.894</td>
</tr>
<tr>
<td>150</td>
<td>0.995</td>
<td>0.985</td>
<td>0.960</td>
</tr>
<tr>
<td>180</td>
<td>0.999</td>
<td>0.998</td>
<td>0.990</td>
</tr>
</tbody>
</table>

*All entries are fractional recoveries.*
reduction performed at 1075 °C with 9.6 pct C. Using the given activation energy and shrinking core model, the kinetic data in columns (a), (c), and (e) of Table VI could be determined. As with the previously defined models, a regression equation describing the \( \left(1/r_e^2\right) \) dependence of particle size on the reaction rate could also be defined.

The kinetic data of column (c) form the basis for the predictions in columns (b), (d), and (f), respectively. The predictions are produced by adjusting \( k[r(t)] \) according to the temperature rule, which predicts an adjustment factor of 0.64, 1, and 1.98 for the temperatures 1075 °C, 1100 °C, and 1140 °C, respectively.

2. Decarburization of liquid iron by carbon dioxide

Sain and Belton showed that between 1160 °C and 1600 °C, the decarburization of high carbon iron is first order with respect to \( \rho_{CO} \).

The values in Table VII were taken directly from the graphs; hence, only the two-decimal accuracy (data in columns (c) and (d) are for the same iron sample). The data bank consisted of two rules, viz., one for temperature and one for pressure. The respective adjustment factors necessary to produce the values in columns (d) and (f) are 0.656 and 1, respectively. These were calculated by the application of the given kinetic model and the given rate constants given in a \( \ln(A) \) vs \( 1/T \) format for the different types of iron samples. Hence, the pressure factor object, which could be defined for the data in columns (f) with respect to the data in column (b), is pressure of \( CO_2 \).

### Table IV. Data in Column (a) Were Used to Calculate \( k[rg(t)] \), Which Is Used to Estimate the Ore Grades [ppm] in Columns (b) and (d), Respectively*

(a) \( \frac{dg}{dt} = -0.7(g - 0.19)^2 \)
(b), (d) Equations [3] and [4]

<table>
<thead>
<tr>
<th>Time (min)</th>
<th>Results of Equations</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2.3</td>
</tr>
<tr>
<td>1</td>
<td>1.03</td>
</tr>
<tr>
<td>2</td>
<td>0.705</td>
</tr>
<tr>
<td>4</td>
<td>0.476</td>
</tr>
<tr>
<td>8</td>
<td>0.334</td>
</tr>
<tr>
<td>16</td>
<td>0.256</td>
</tr>
<tr>
<td>23</td>
<td>0.23</td>
</tr>
</tbody>
</table>

*Table taken from Reference 27

### Table V. Data in Column (a) Are Used to Calculate \( k[r(t)] \)*

<table>
<thead>
<tr>
<th>Time (min)</th>
<th>150 °C</th>
<th>140 °C</th>
<th>150 °C</th>
<th>150 °C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(a)</td>
<td>(b)</td>
<td>(c)</td>
<td>(d)</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>30</td>
<td>0.75</td>
<td>0.75</td>
<td>0.50</td>
<td>0.54</td>
</tr>
<tr>
<td>45</td>
<td>0.92</td>
<td>0.92</td>
<td>0.72</td>
<td>0.75</td>
</tr>
<tr>
<td>60</td>
<td>0.95</td>
<td>0.95</td>
<td>0.87</td>
<td>0.88</td>
</tr>
<tr>
<td>75</td>
<td>0.95</td>
<td>0.95</td>
<td>0.95</td>
<td>0.94</td>
</tr>
<tr>
<td>90</td>
<td>0.95</td>
<td>0.95</td>
<td>0.95</td>
<td>0.95</td>
</tr>
</tbody>
</table>

*All entries are fractional recoveries.

### Table VI. Data in Column (c) Are Used to Calculate \( k[r(t)] \)*

(a), (c), and (e): \( 1 - (2/3)a - (1 - a)^{2/3} = k_t \)
(b), (d), and (f): Equations [5] and [6]

<table>
<thead>
<tr>
<th>Time (min)</th>
<th>1075 °C 0.00115</th>
<th>1100 °C 0.00179</th>
<th>1140 °C 0.00354</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(a)</td>
<td>(b)</td>
<td>(c)</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>0.408</td>
<td>0.409</td>
<td>0.495</td>
</tr>
<tr>
<td>40</td>
<td>0.550</td>
<td>0.551</td>
<td>0.656</td>
</tr>
<tr>
<td>60</td>
<td>0.646</td>
<td>0.647</td>
<td>0.761</td>
</tr>
<tr>
<td>80</td>
<td>0.720</td>
<td>0.720</td>
<td>0.836</td>
</tr>
<tr>
<td>100</td>
<td>0.779</td>
<td>0.779</td>
<td>0.893</td>
</tr>
<tr>
<td>120</td>
<td>0.827</td>
<td>0.827</td>
<td>0.935</td>
</tr>
<tr>
<td>140</td>
<td>0.866</td>
<td>0.866</td>
<td>0.966</td>
</tr>
<tr>
<td>180</td>
<td>0.927</td>
<td>0.927</td>
<td>0.999</td>
</tr>
</tbody>
</table>

*All entries are fractional recoveries.
3. Nickel oxide reduction with hydrogen

Szekely and Evans\(^{134}\) used a structural model to describe the reduction kinetics of solid and porous NiO pellets with H\(_2\). Since the two porosities of 0.384 and approximately 0.73 correspond to two different diffusional scenarios, they cannot be compared purely by a manipulation of the rate constant. Each porosity has its own characteristic kinetic rate data. Hence, in this example, column (c) of Table VIII was used to calculate \(k(\tau)|_1\) and subsequently predict the values in columns (b) and (d) while the values in column (g) were used to predict the values in columns (f) and (h). This could be achieved by defining two rules for each porosity, one for temperature on the basis of the given activation energy and one based on the inverse of the radius of the pellet.

Hence, the if-then rule for column (a) would have the following form:

if NiO reduction by H\(_2\),
and porosity = 0.728,
and temperature = 299 °C,
and pellet radius = 0.77 cm
then slope factor = \(\Phi = 0.96 \cdot 1.53 = 1.47\)
load factor = \(\Omega = 1\)
cost factor = 1

The adjustment factors produced by the other rules are summarized in Table VIII. The subsequent deep knowledge is used to perform the simulation. The two-decimal accuracy is attributed to the inaccuracies involved while estimating data from the kinetic curves given by Szekely and Evans.\(^{134}\)

### Table VIII. Data in Columns (c) and (g) Are Used to Calculate \(k(\tau)|_1\) for the Two Respective Porosities

<table>
<thead>
<tr>
<th>Time (min)</th>
<th>0.77 cm</th>
<th>1.178 cm</th>
<th>0.80 cm</th>
<th>1.128 cm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.728</td>
<td>0.728</td>
<td>0.734</td>
<td>0.384</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Porosity</th>
<th>0.384</th>
<th>0.728</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.83</td>
<td>0</td>
</tr>
<tr>
<td>2.5</td>
<td>0.89</td>
<td>0.86</td>
</tr>
<tr>
<td>5.0</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>10.0</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>15.0</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>20.0</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>45.0</td>
<td>1.00</td>
<td>1.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Factors</th>
<th>(n_t)</th>
<th>(n_t)</th>
<th>(n_t)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(n_t)</td>
<td>1.53</td>
<td>1</td>
<td>1.41</td>
</tr>
<tr>
<td>(n_t)</td>
<td>0.96</td>
<td>1</td>
<td>1.23</td>
</tr>
<tr>
<td>(n_t)</td>
<td>1.47</td>
<td>1</td>
<td>1.73</td>
</tr>
</tbody>
</table>

*All entries are fractional recoveries.
a five-stage CIP system as described by Nicol et al.\textsuperscript{191} (Volume of reactors 400 l and other data are as in Table X).

In this simulation, the pulp flow rate amounts to 400 l/h and a total transfer of the carbon inventory occurs every 48 hours. The entries in columns (a) of Table X are measured process data, the entries in columns (b) are as simulated by the model suggested by Nicol et al.\textsuperscript{191} and the entries in columns (c) are as simulated by the model proposed here.

The amount of carbon predicted by the model proposed here, \textit{i.e.}, 22 g/l, is much closer to the real process value of 25 g/l than that predicted by Nicol et al.\textsuperscript{191} of 17 g/l. Using the rules of the data base model, the process conditions could be determined which produce the exact carbon concentration of 25 g/l. This could be expected in view of the fact that the data base is an accurate representation of the real data.

2. \textit{Simulation of a continuous CIL cascade}

This simulation is an extension of the CIP simulation discussed above, with adsorption and leaching taking place simultaneously. The retention time in the three leaching pachucas is 3 h/pachua, while that in the following five leaching/adsorption tanks amounts to 1 h/pachua. The reader is referred to Reuter et al.\textsuperscript{179} for the results of this simulation. A simulation involving a continuous transfer of carbon and the effect of carbon poisoning was discussed by Reuter and van Deventer.\textsuperscript{184}

E. \textit{Fault Diagnosis}

To illustrate various aspects of this model, two typical edit windows of the program are given. Figure 3 depicts the editing of the adjustment object oxygen pressure, while Figure 4 illustrates the final result after a search through the shallow knowledge to identify a fault in the Blackbird system.\textsuperscript{130}

1. \textit{Fault diagnosis in a batch reactor (option 2)}

Consider the leaching of the Blackbird cobalt concentrate.\textsuperscript{131} The if-then rule comprised of the appropriate pivot adjustment objects is repeated here:

\begin{itemize}
  \item leaching of blackbird cobalt concentrate
  \item distinguishing characteristic
    \begin{itemize}
      \item pressure = 1200 kPa
      \item temperature = 150 °C
      \item 30 pct solids
      \item 46 g/l Na$_2$SO$_4$
    \end{itemize}
  \item slope factor = $\Phi = 1$
  \item load factor = $\Omega = 1$
  \item cost factor = 1
\end{itemize}

The concentration-time data associated with this pivot data may be found in column (a) of Table V. The objective now is to diagnose the process if the data in column (c) of the same process are given as an input. The appropriate sum of time-weighted slopes is subsequently determined from this data (Eq. [8]), and if possible, the final attainable recovery is produced.

As may be seen from Table X, only a few of the slope-load data points are given. It is possible to calculate the time-weighted slope for each of the complete data sets, which are 20.043 for the pivot data and 18.075 for the data being diagnosed, respectively. As these data are inversely proportional to the rate $k'$ (Eq. [8]) and since the starting concentration of each are identical, a $\Phi$ value of 1.11 (20.043/18.075) could be estimated. It is also obvious from the given data that the final recovery for Co for both sets of data is 95 pct; hence, $\Omega = 1$. These two bits of information are the basis of subsequent diagnoses.

The inference engine now attempts to match these $\Phi$ and $\Omega$ values with the knowledge available within a pivot object which corresponds to the type of ore being considered. As $\Omega = 1$, the search focuses on $\Phi$, and it would be clear that the diagnoses would produce the following operating conditions:

- temperature = 150 °C
- 30 pct solids
- 20 g/l Na$_2$SO$_4$
- 1200 kPa

as these data produce an adjustment rule with a $\Phi$ value of 1.11.

### Table X. Comparison of Process Data, Predictions by Nicol et al.,\textsuperscript{191} and the Model Proposed in this Article

<table>
<thead>
<tr>
<th>Stage</th>
<th>Solution (g/l)</th>
<th>Loading (g/l)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(a)</td>
<td>(b)</td>
</tr>
<tr>
<td>Feed</td>
<td>2.02</td>
<td>2.02</td>
</tr>
<tr>
<td>1</td>
<td>0.76</td>
<td>0.52</td>
</tr>
<tr>
<td>2</td>
<td>0.24</td>
<td>0.13</td>
</tr>
<tr>
<td>3</td>
<td>0.039</td>
<td>0.034</td>
</tr>
<tr>
<td>4</td>
<td>0.015</td>
<td>0.009</td>
</tr>
<tr>
<td>5</td>
<td>0.002</td>
<td>0.002</td>
</tr>
<tr>
<td>$M_c$</td>
<td>25 g/l</td>
<td>17 g/l</td>
</tr>
</tbody>
</table>

552—VOLUME 228, AUGUST 1991

\textsuperscript{130} METALLURGICAL TRANSACTIONS
Fig. 4—Result after a search through the Blackbird knowledge base.

Fig. 3—Editing the oxygen pivot object for the Blackbird cobalt concentrate.

Fig. 4—Result after a search through the Blackbird knowledge base.

Fig. 3—Editing the oxygen pivot object for the Blackbird cobalt concentrate.
of a shallow level search would then point out that the Na$_2$SO$_4$ is 20 g/l, below the 46 g/l of the pivot object.

To summarize, an initial shallow level search would have produced the results given above if they had been present (option 1). If these data were not available, a second deep-level search would have been performed by the application of if-then search rules until a matching final recovery is obtained. If a matching final recovery is found, the search proceeds to a search of the sum of time-weighted slopes.

2. Fault diagnosis in continuous reactors (option 2) The simulation of the continuous CIP is taken as an example here. As was clear from the results of this example discussed previously, the model produced a carbon concentration of 22 g/l, while it is 25 g/l in actual fact. This discrepancy is attributed to process conditions being dissimilar to the pivot data that are valid for the calculated slope-load.

Fault diagnosis of this process would have identified a decrease in the loading capacity of the carbon due to carbon poisoning as a possible reason. This is effected by a deep-level search through the knowledge base, a simultaneous numerical simulation of the CIP process, and a comparison to the pivot data for this process.

IV. CONCLUSIONS AND SIGNIFICANCE

The discussion and results above serve to illustrate the following regarding the knowledge-based system:

1. The shallow and deep knowledge of the knowledge base can be applied to simulate batch and continuous adsorption, leaching, a combination of these, and reduction processes.

2. It is shown how a very simple model and adjustment factors can be used successfully to perform both fault diagnosis or process identification and the above-mentioned process simulation.

3. The application of shallow and deep knowledge in the form of adjustment objects grouped into larger objects (e.g., pivot objects) permits the simulation of the above-mentioned reaction systems at all process conditions catered to by the knowledge base.

4. The user has total control over all shallow and deep knowledge presented in the form of facts, objects (frames), rules, and functions. These define both physical and chemical conditions in the reactors and may be edited via user-friendly menu interfaces.

5. Although not shown, cost objects and associated cost adjustment objects may be applied to perform economic evaluation of the mentioned processes.

6. No example has been given in which the load adjustment factor has been explicitly discussed. The interested reader is referred to an article by Reuter and van Deventer
dealth in which this aspect is discussed for flotation.

LIST OF SYMBOLS

\[ A, B \text{ constants} \]
\[ a, b, c, d \text{ constants} \]
\[ C_n, C_i, C_p \text{ concentration in a batch reactor,} \]
\[ C_n, C_i, C_p \text{ concentration in stage } i \text{ and data point } n, \text{ respectively [ppm] or [-]} \]
\[ C_i, C_p \text{ initial concentration [ppm] or [-]} \]
\[ r, r_i \text{ ore grade in a batch reactor, grade in stage } i, \text{ and grade data point } n, \text{ respectively [g/ton] or [-]} \]
\[ g, g_i \text{ pivot and initial ore grade, respectively [g/ton]} \]
\[ k \text{ rate variable as a function of } y(t), r(t), \text{ and } g(t) \text{ [hl]} \]
\[ k_1, k_2 \text{ time-weighted slopes [hl]} \]
\[ M_i, M_e \text{ mass concentration of carbon, liquid} \]
\[ M_i, M_e \text{ mass concentration of carbon, liquid} \]
\[ m_i, m_e \text{ mass flow rate of the carbon countercurrent and} m_i \text{ for cocurrent flow [kg/h]} \]
\[ M_{o,c} \text{ mass of carbon being transferred countercurrent and } M_{o,c} \text{ cocurrent [kg]} \]
\[ p \text{ pressure [Pa]} \]
\[ r(t) \text{ concentration of unreacted compound in pyrometallurgical reactions [ppm] or [-]} \]
\[ r_{1,0}, r_n \text{ rate equation [ppm/h]} \]
\[ r_n \text{ radius of pellet [cm]} \]
\[ t \text{ time [h]} \]
\[ T \text{ temperature [°C]} \]
\[ V_{e, s, c} \text{ active stage volume [m$^3$]} \]
\[ y, y_i \text{ pulp flow in and } y_i \text{ but of a stage [m$^3$/h]} \]
\[ y_i \text{ carbon loading in a batch reactor, loading in stage } i, \text{ and loading data point } n \text{ [ppm] or [-]} \]
\[ \delta t \text{ time increment} \]
\[ \text{Re, Sc} \text{ Reynolds, Schmidt, and Sherwood numbers} \]
\[ \text{Sh} \text{ recovery [-]} \]
\[ \alpha_{t,2} \text{ for adsorption [-]} \]
\[ \beta \text{ for leaching [-]} \]
\[ \Phi, \Omega \text{ slope and load adjustment factors, respectively [-]} \]

REFERENCES


METALLURGICAL TRANSACTIONS
The simulation and identification of flotation processes by use of a knowledge based model

M.A. Reuter and J.S.J. van Deventer

Department of Metallurgical Engineering, University of Stellenbosch, Stellenbosch, South Africa

(Received 2 November 1990; accepted after revision 13 November 1991)

ABSTRACT


This paper illustrates the application of a knowledge based system to perform fault-diagnosis, dynamic simulation and the optimization of batch and continuous flotation processes. The proposed approach is essentially useful to simulate ill-defined dynamic processes where existing fundamental and empirical models fail owing to their lack of generality.

A generalized kinetic model is defined and linked to a knowledge base via adjustment objects in order to perform flotation process simulation. The kinetic model and the accompanying knowledge base are also applied to produce the bounds to a generalized linear programming model of a generalized flotation plant to assist in its optimal design. It is shown that the proposed approach could be used to simulate flotation data for batch and continuous operations. In contrast with most existing models, no curve-fitting is required, as the kinetic model utilises experimental data directly. Also, adjustment factors are defined in an intelligent way in the knowledge-based system so as to relate the operating conditions to the kinetics of flotation. In this way, a generalised description of kinetic behaviour is effected.

It is shown how the generalized kinetic model is applied to perform fault diagnosis i.e. to identify the process conditions within a flotation cell or bank, and to recommend process conditions which could improve flotation behaviour.

1. INTRODUCTION

The simulation, identification and optimization of flotation circuits is very complicated by virtue of the large number of process variables which are involved during flotation. The numerous factors that influence the flotation of valuable components in a flotation cell are well documented (Lynch et al.,

Correspondence to: M.A. Reuter, Institut für Metallhüttenwesen und Elektrometallurgie, RWTH, Aachen, Germany.

0301-7516/92/$05.00 © 1992 Elsevier Science Publishers B.V. All rights reserved.
Schumann (1942) explained that the kinetic behaviour of minerals during flotation will in many cases supply worthwhile information regarding the process and hence the effect of the various process conditions. Schumann (1942) demonstrated that changes of process conditions within flotation cells have a measurable effect on the flotation response. In view of this observation, Dowling et al. (1986) investigated the reliability of various kinetic flotation models to describe the flotation of selected copper ores, a first step in the direction of the optimization of a flotation process. Of the thirteen models that were considered a two parameter first order model with a rectangular distribution of floatabilities and a two parameter model with constant floatability were found to be superior. However, the claimed superior first order model with a rectangular distribution produced a final recovery of more than 100% for the least squares fitted data. Most other models performed badly owing to a large confidence limit on the fitted parameters of the models. Although the authors could demonstrate that, for example, the final recovery increases with decreasing particle size, the large confidence limits makes any precise characterization of the specific flotation process difficult. The large number of different rate equations with parameters which all have a different meaning (except the final recovery) makes the formalization of a specific identification method even more difficult.

The parameters of the many existing fundamental models lump numerous process conditions, hence, little information regarding the process can be obtained from them other than the final recovery which could increase or decrease. Attempts have, however, been made to define the parameters of the rate equations in terms of specific process conditions. For example, Frew (1982) suggested an exponential relationship to define the variation of the first order rate variable as a function of density in zinc cleaning circuits, while Lynch et al. (1981) related the collector addition rate to the ball mill of a chalcopyrite circuit to the slow floating fraction of this ore as described by a four parameter model. Performing this type of analysis for each specific model (claiming it to be better than another model), does little to place all these investigations on a common base.

Hence, it is the objective of this paper to propose a two parameter knowledge-based kinetic model which should have the following characteristics: (i) it must fit the data exactly, irrespective of the type of process, (ii) variances in fitted parameters produced via least squares methods must not affect the accuracy, (iii) it must be a function of all possible process conditions e.g. frother and collector addition, depressant, pH, aeration rate, density etc., (iv) and the proposed model and accompanying knowledge based structure must be simple to facilitate its solution by the application of a commercial knowledge based tool e.g. a hybrid expert shell such as “Leonardo” (1989).

The manner in which this is accomplished in this paper is by using a hybrid
knowledge based approach, i.e. both qualitative (experiential and heuristic) and quantitative (equation orientated) knowledge is used to describe the kinetics of the flotation process. This is achieved by applying a variety of tools offered by knowledge based technology, as is clear from the various applications. Various ill-defined chemical engineering and mineral processing problems have been modelled and solved by using knowledge based techniques. Most of these applications (many prototype examples), however, have been concerned with steady-state circuit design (Stephanopolous et al., 1987; Kirkwood et al., 1988; Beltramini and Motard, 1988), programmes for the selection of equipment, processes or reagents e.g. DECADE for the selection of catalysts (Banares-Alcantara et al., 1987; 1988), synthesis of operating procedures (Stephanopolous et al., 1987; Lakshmanan et al. 1988; Fusillo and Powers, 1988; Foulkes et al., 1988), malfunction diagnosis (Shum et al., 1988; Rich and Venkatasubramanian, 1987; Ramesh et al., 1988; Venkatasubramanian and Rich, 1988; Hoskins and Himmelblau, 1989) and control synthesis (Stephanopolous et al., 1987; Tzouanas et al., 1988; Lewin and Morari, 1988; Birkey et al., 1988).

Little emphasis has, however, been placed to date on the application of these techniques to the modelling of mineral processing systems. Tucker and Lewis (1988) applied a knowledge based system to simulate gravity separation plants, while a couple of applications for the control of mineral processing plants have been reported (Ynchausti and Hales, 1990; Davis et al., 1990). The dynamic simulation of ill-defined problems by the use of these techniques has, however, found little application in the literature. Dalle Molle et al. (1988) and Kuipers (1984) proposed the use of qualitative simulation for this purpose. Reuter et al. (1991) and Reuter and Van Deventer (1991) demonstrated how a KBS can be combined with a system of dynamic equations to simulate leaching, pyrometallurgical reduction, adsorption onto resin, carbon-in-pulp (CIP) and carbon-in-leach (CIL) systems. No diagnostic system has really attempted to address diagnostics on a basis that includes all possible process conditions within a particular unit operation. Model based diagnosis, as it is proposed here, has been reported for the diagnosis of electronic circuits (Davis, 1984).

In the above-mentioned systems various architectures have been applied to formalize the knowledge e.g. object orientated (Stephanopolous et al., 1987; Venkatasubramanian and Rich, 1988), frame orientated (Beltramini and Motard, 1988), hybrid systems in blackboard (Banares-Alcantara et al., 1987, 1988), hierarchical (Lakshmanan and Stephanopoulos, 1988; Shum et al., 1988) and goal-tree–success-tree (Birkey et al., 1988) systems. A variety of problem solving strategies exist to solve these problems which may be divided into (i) AI-programming languages e.g. PROLOG (Clocksin and Mellish, 1981) and LISP and (ii) general shells e.g. "Leonardo" for knowledge representation, which include forward and backward chaining, inheritance frames, con-
straints, and probabilistic, non-monotonic and temporal reasoning; some systems only support rules e.g. OPS5 (Forgy, 1981) and EMYCIN (Shortliffe and Buchanan, 1975; Van Melle, 1981) while others are hybrid systems combining a number of knowledge representation techniques e.g. KEE (1986).

The methodology selected to solve the problem discussed in this paper is based on an object orientated approach. The various objects are situated in an hierarchical network architecture permitting various rules to be applied so as to extract knowledge from the knowledge base following a backward chaining algorithm as applied in the MYCIN expert system, for example. In this paper, this methodology will be applied to illustrate the following regarding the proposed model:
- its ability to simulate batch cells and continuous flotation banks as a function of the physical and chemical environment within the reactors, and
- perform fault diagnosis of the process conditions within batch and continuous flotation cells and banks respectively by following a rule and object orientated approach.

Diagnosis is implied here to encompass the identification of the process conditions within the flotation cell, and not the fault analysis of actuators, sensors, pumps etc., which constitute the macroscopic aspects of a plant. Hence, the results of a fault diagnosis will produce information regarding the conditions within the cell and hence how the faults can be alleviated to produce an optimal recovery once again.

It will also be shown how a steady state optimization approach can be integrated into the proposed knowledge based (KBS) model discussed in this paper. The basic theory that has been applied to perform this optimization task on the basis of linear programming has already been developed by Reuter and Van Deventer (1990).

2. DEVELOPMENT OF THE KNOWLEDGE BASED MODEL

The importance of an accurate and generalized kinetic model was emphasized in the introduction. As the kinetic model proposed here forms the basis of the knowledge based model this will be discussed first. The mathematical basis for characterizing an ore and to perform fault diagnosis will be discussed followed by the development of the knowledge base in the form of rules and objects. Finally, the integration of the linear programming model with the knowledge based model to perform circuit optimization will be discussed.

2.1. Kinetic model

As has been defined for leaching, adsorption (activated carbon and resin) and pyrometallurgical reactions (Reuter et al., 1991; Reuter and Van Deventer, 1991), the rate of depletion of the valuable element in the slurry of a
batch cell has been defined by a non-linear first order rate equation, in which the “rate variable” \( k[C^e(t)]_n \) is only a function of \( C^e(t) \), the concentration of element \( e \) (e.g. \( e= \) galena, water, gangue). What this definition of \( k[C^e(t)]_n \) postulates is that the concentration of a species remaining in the pulp during a batch flotation test for example, is an indication of its floatability. This relationship between concentration of the valuable species remaining in the pulp and its rate is postulated here to be an identification number (ID) of the flotation process being considered. Flotation species (e.g. fast and slow floating species within \( e \)) are implicitly catered for in this model. The value of the slope \( k[C^e(t)]_n \) and valid concentration ranges are determined as given below from a flotation curve, which covers the flotation process up to the final recovery \( R \) of the process i.e. until the slope approaches zero.

\[
\frac{dC^e}{dt} = -k[C^e(t)]_n C^e(t)
\]

where

\[
k[C^e(t)]_n = -2(C_{n+1}^e - C_n^e)/\delta t(C_{n+1}^e + C_n^e)
\]

holds for

\[
C_{n+1}^e(t) \leq C^e(t) \leq C_n^e(t)
\]

It is clear that \( k[C^e(t)]_n \) is the average slope between concentrations \( C_n^e \) and \( C_{n+1}^e \). The value of \( k[C^e(t)]_n \) corresponds to the pivot process variables i.e. the standard conditions (discussed in more detail below). Any deviation from the pivot-data is reflected by the \( \alpha \) and \( \beta \) values of the adjustment objects, which will also be defined below. If these \( \alpha \) and \( \beta \) values deviate from 1, the following equation may be applied to predict the kinetics of the flotation process, were \( k[C^e(t)]_n \) is determined by eq. 1.

\[
\frac{dC^e}{dt} = -\alpha \cdot k[C^e(t)]_n [C^e(t) - C_0^e \cdot (1 - \beta)] = r
\]

The adjustment of the validity range for each slope variable \( k[C^e(t)]_n \) is performed as follows:

\[
C_0^e - \beta \cdot [C_0^e - C_{n+1}^e(t)] \leq C^e(t) \leq C_0^e - \beta \cdot [C_0^e - C_n^e(t)]
\]

From the definition of this model, it is clear that each flotation process within a given environment is defined by a set of discrete slopes and corresponding concentrations and can, within bounds which depend on the flotation process, be manipulated by the \( \alpha \) and \( \beta \) values via objects and rules. As this model follows the kinetic curve, it gives an accurate description of the rate as well as the final recovery, which are determined directly by simple algebra from the kinetic data (the number of intervals for \( n \) may be adjusted to suit the required accuracy). This would hence imply that no curve fitting is performed.
and that any type of flotation process could be described accurately. Refraining from the use of least squares curve fitting has the advantage that parameter variances arising during fitting are not included in the $\alpha$ and $\beta$ values. The normal kinetic models for flotation (Dowling et al., 1986) would hence not permit such an accurate manipulation with $\alpha$ and $\beta$ values, unless, however, the applied kinetic model is a complete description of the kinetic data, i.e. when the variances for the estimated parameters are zero.

The simplicity of the model permits the following theoretical "first" order equation to be formulated for batch processes:

$$C^\circ(t) = C^\circ_n \cdot \exp\{-k[C^\circ(t)]_n \cdot (t-t_n)\} \tag{3}$$

which holds for the interval $n$ to $n+1$.

The generality of this model will not become clear by considering flotation only (see Reuter et al., 1991 and Reuter and Van Deventer, 1991, for applications other than flotation). However, in order to demonstrate its generality in this paper, its performance will be compared to three other well established flotation models.

The kinetics of all other components (secondary) are also described by the same kinetic model (eq. 1). This includes all other mineral species as well as for example, the depletion of collector in a flotation cell or bank.

2.2. Equations for fault diagnosis

Not only does the kinetic equation (1) permit a precise description of any kinetic data but it also creates a basis on which a flotation process may be characterized. This also permits the diagnosis of the chemical and physical process conditions within the cell.

Consider the classical first order reaction rate equation and the following time-weighed slope integral for this rate law:

$$\int_0^\infty \int_0^\infty t \cdot k C^\circ \cdot dt = \int_0^\infty t \cdot k_0 C_0^\circ \cdot e^{-k t} \cdot dt = C_0^\circ / k' \tag{4}$$

If the value for $1/k$ (normalized with respect to $C_0^\circ$) is compared to that of another curve with a normalized rate constant $k'$, the ratio $k'/k$ gives a value for the adjustment factor $\alpha$. For this equation, $\beta$ is equal to 1.0 as the final recovery for both rate equations are 100%, irrespective of the rate constants. For $\beta \neq 1.0$, $k'/k$ must be divided by $\beta$. For a process described by the rate equation (1), eq. 4 is generalized by an Euler integration of eq. 4 to:

$$C_0^\circ / k'' = \sum_n t_n \cdot k[C^\circ(t_n)] \cdot C^\circ(t_n) \cdot \delta t \tag{5}$$
By comparing two different sets of kinetic data normalized with respect to $C_0$, the ratio of the inverse of the resulting time-weighed average $k''$-values gives an estimate for $\alpha$. Furthermore, the ratios of the final recoveries, i.e. where $k[C^*(t)]_n$ become zero for the two data sets, are an estimate for $\beta$. As already pointed out earlier, the condition for which $\alpha$ and $\beta$ both equal 1 is considered to be the pivot condition.

The value for $k''$ and the final recovery serve as a fingerprint of the kinetic data on which they are based. These two values form the basis of all ore characterization as well as diagnostic functions. The algorithm applied to perform the diagnostic functions will be discussed below.

2.3. Knowledge representation

Knowledge representation (KR) can be considered to be the central problem of modern artificial intelligence. Hence, the representation of shallow (experiential and heuristic) and deep (equation orientated) knowledge for the application in knowledge based systems, is probably one of the most important aspects of the modelling approach discussed here. One of the first AI researchers to realize that knowledge representation had to be addressed as a problem in itself was Quillian (1969). He introduced semantic networks for representing knowledge, which contain information about objects, concepts and the relationship between them. This has been extended to frame systems, which can be thought of as hierarchical relational databases with inheritance properties (Chabris, 1988).

In the approach discussed here the batch flotation concentration–time recovery data of the valuable element, as well as all other species involved, form the basis of the knowledge base. These data are transformed into a useful form by the discussed generalized kinetic model, the “rate constant” also being a function of all the effects of the flotation process variables via $\alpha$ and $\beta$ adjustment objects. The organization of the data in this manner permits the interaction between the batch flotation concentration–time data, process variables, kinetic model and mathematical modelling. This is achieved here by knowledge representation techniques such as objects (frames), rules and functions (Dalle Molle et al. 1988).

2.3.1. Pivot object

Each pivot-object is structured so as to include all relevant information regarding a particular process class and sub-class and its effect on the process. Values which are not contained by the slots of this object are inherited from other objects. An example here is the pivot-object, that defines both shallow and deep knowledge regarding a sphalerite flotation process. The pivot conditions which are captured by this object are those for cells 1–4 of experiment 1 (Frew and Davey, 1988).
Class
Sphalerite Ore A

Sub-class
Stope X

Kinetic data (sets of kinetic data e.g. change of reagents)
<table>
<thead>
<tr>
<th>Time</th>
<th>Ce</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0.5</td>
<td>0.946</td>
</tr>
</tbody>
</table>

4 Factor adjustment-objects
Particle size
Total % Zn in feed
Depressant addition
Collector addition

k Regression-objects
None

Unit operation
Continuous

Function/procedural
See below

Cost object
None

As explained above, each of this object's class and sub-class are characterized by the \( k'' \) and final recovery values for the given kinetic data. This implies that the sub-class of a class inherits the values of \( k'' \) and \( R \) as a function of the contents of slots within this object. The knowledge base would consist of a variety of these pivot objects to create a meta-class, i.e. a knowledge base for a specific flotation plant.

This pivot object consists of a variety of other types of objects which will be discussed briefly below. Due to the importance of the adjustment objects, a separate section is devoted to discuss them.

Unit operation object. This object describes the physical operating conditions of a set of reactors (data from Frew and Davey, 1988).

<table>
<thead>
<tr>
<th>Unit operation</th>
<th>Continuous reactors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mineral content of feed</td>
<td>11.8%</td>
</tr>
<tr>
<td>% Solids in feed</td>
<td>43.3%</td>
</tr>
<tr>
<td>Retention time of cells 1–4</td>
<td>11.2 min</td>
</tr>
<tr>
<td>Number of reactors in series</td>
<td>4</td>
</tr>
<tr>
<td>Feed flowrate of the pulp</td>
<td>43.3 t/h</td>
</tr>
<tr>
<td>% Short circuiting of feed</td>
<td>0</td>
</tr>
<tr>
<td>% Dead volume of reactor</td>
<td>0</td>
</tr>
</tbody>
</table>

The user may edit these values if it is required.

Function/procedural object. These objects include the kinetic models, differential equations that define the non-ideal flow through the continuous reactors or equations that describe fundamental aspects of the process. Each function/procedural object would hence contain one of the following elements:
Kinetic model (differential equation as above)
Unit-operation-object (as above)
Non-ideal flow model (differential equation)
Linear programming optimization model
Fundamental correlations (e.g. \( Sh = a(Re)^b(Sc)^c + d \))

The last three items in this object will be discussed here briefly.

Non-ideal flow model. Ideal flow is assumed in the formulation of the continuous flow model for the depletion of an element \( e \) in cell \( i \) in the example below, although in the KBS non-ideal flow in the form of short-circuiting
\[
C_{i-1}^e = \left\{ \Phi C_{i-1}^e + (1 - \Phi) C_{i-2}^e \right\}_{\text{old}}
\]
and dead volume \([V_{\text{act}} = V_{\text{total}} - V_{\text{dead}}]\) is incorporated. The theoretical values of the dead volume and bypass streams can be obtained from tracer tests, or can be estimated by comparing real plant behaviour with that predicted from an ideal flow model.

\[
\frac{dC_i^e}{dt} = \frac{F_{\text{p.in}}}{V_{\text{act}}} C_{i-1}^e - \frac{F_{\text{p.out}}}{V_{\text{act}}} C_i^e + r
\]  

Linear programming model. Although the basic linear programming model has already been developed by Reuter and Van Deventer (1990), the more important equations will be repeated here, as they are a simplification of the former formulation. This simplification arises due to the already mentioned fact that only elements \( e \) are considered in this model, whereas the generalized kinetic model intrinsically caters for all flotation species of an element \( e \). Consequently, in view of this simplification, the linear programming model may be formulated as follows for the valuable element \( e \): (Consult Reuter and Van Deventer (1990) for details, as the redevelopment of this linear programming model falls outside the scope of this paper.)

\[
OBJ = \sum_{j \neq i} w_e a_{ej}
\]  

\[
\sum_{j \neq i} r m_{ij}^e + \sum_{j \neq i} r y_{ij}^e + u_{ej} = m_{ej} + y_{ej}
\]  

\[
\sum_{j \neq i} r m_{ij}^e + b_{ej} = m_{ej}
\]  

\[
\sum_{j \neq i} r y_{ij}^e + a_{ej} = y_{ej}
\]  

\[
(y m_{ej})^L m_{ej} \leq y_{ej} \leq (y m_{ej})^U m_{ej}
\]  

\[
(u_{ej})^L \leq u_{ej} \leq (u_{ej})^U
\]  

or instead of eq. 12, eqs. 13 and 14:

\[
(a_{ej})^L \leq a_{ej} \leq (a_{ej})^U
\]  

\[
(b_{ej})^L \leq b_{ej} \leq (b_{ej})^U
\]
Subject to the structure and flow rates produced for the valuable element $e$, a similar model may be formulated for the other elements. The objective function attempts to maximize the recovery of the valuable element $e$ based on the mass balance equations only and the weight gives the cost for the valuable element. This objective function would become more complex for linear programming models that include a number of economic equations in addition to the mass balance equations.

**Fundamental correlation.** A semi-empirical correlation may also be applied to adjust the pivot rate via the $Sh$-number, which is a function of the rate:

$$\alpha = \frac{f(Sh)}{f(Sh_{\text{pivot}})}$$

(15)

This approach permits the adjustment of $\alpha$ over a range of operating conditions dictated by the validity range of the specific correlation being considered.
Cost object. These have not been defined here, but contain all cost data for the pivot object which can also be adjusted by appropriate cost adjustment factors in the adjustment objects.

2.3.2 Adjustment objects

The adjustment objects form the link between the kinetic model and the knowledge base. Two types of adjustment objects have been defined here viz. the factor and regression adjustment objects respectively. Their function is to define deviations from the pivot conditions and the effect these deviations have on the \( \alpha \) and \( \beta \) values. In this capacity each object must be independent of the other, otherwise their combined effect cannot be determined by multiplication. If the independence of each adjustment object cannot be ensured for too large deviations from the pivot values, a new sub-class or pivot object should be defined. Another approach is also possible in which a number of dependent adjustment objects are grouped together to produce a particular adjustment object group, with the additional filling of the interaction slots with suitable values.

The position of a factor adjustment object in the knowledge base is depicted by Fig. 1.

Factor adjustment object. Each factor adjustment-object is characterized by a property \((m \text{ possible})\) and \(n\) sub-properties. The following four adjustment objects define the pivot conditions of the pivot-object given above for the sphalerite flotation described by Frew and Davey (1988):

<table>
<thead>
<tr>
<th>Property</th>
<th>Sub-property</th>
<th>Particles: (d_{50} = 80 \mu m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sub-property</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(n-1) interaction factors</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Slope factor</td>
<td>(\alpha_1 = 1)</td>
<td></td>
</tr>
<tr>
<td>Final recovery factor</td>
<td>(\beta_1 = 1)</td>
<td></td>
</tr>
<tr>
<td>Cost factor</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>%Zn in feed</td>
<td></td>
</tr>
<tr>
<td>Sub-property</td>
<td>11.8%</td>
<td></td>
</tr>
<tr>
<td>(n-1) interaction factors</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Slope factor</td>
<td>(\alpha_2 = 1)</td>
<td></td>
</tr>
<tr>
<td>Final recovery factor</td>
<td>(\beta_2 = 1)</td>
<td></td>
</tr>
<tr>
<td>Cost factor</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>Collector</td>
<td></td>
</tr>
<tr>
<td>Sub-property</td>
<td>14 g/t</td>
<td></td>
</tr>
<tr>
<td>(n-1) interaction equations</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Slope factor</td>
<td>(\alpha_3 = 1)</td>
<td></td>
</tr>
<tr>
<td>Final recovery factor</td>
<td>(\beta_3 = 1)</td>
<td></td>
</tr>
<tr>
<td>Cost factor</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
Note that the bottom three objects are dependent on one another. If they are to be included in the same pivot-object, they must be defined in such a way that they include interaction effects. They must thus always be considered to form an adjustment object group.

The following two objects reflect the effect of the depressant ZnSO$_4$ on the flotation kinetics of galena and marmetite (Lynch et al., 1981, p. 144):

<table>
<thead>
<tr>
<th>Property</th>
<th>ZnSO$_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sub-property</td>
<td>790 g/tonne</td>
</tr>
<tr>
<td>$n-1$ interaction factors</td>
<td>1</td>
</tr>
<tr>
<td>Slope factor</td>
<td>$\alpha_5 = 1$</td>
</tr>
<tr>
<td>Final recovery factor</td>
<td>$\beta_5 = 1$</td>
</tr>
<tr>
<td>Cost factor</td>
<td>1</td>
</tr>
</tbody>
</table>

These objects assume that all other pivot conditions remain constant. The $\beta_6$ value originates from the ratio of the final recoveries for the two sets of kinetic data defined by these adjustment objects viz. 96/93.9. The $\alpha_6$ value could be defined by considering the ratio of the two time-weighted slopes for the two sets of kinetic data which are 0.5744 and 0.4235 respectively. The calculation of time-weighted slopes will be illustrated by an example in the fault-diagnosis section below.

The freshness of sodium isopropyl xanthate and its effectiveness is best defined by a factor adjustment object such as (from the data of Ackerman et al. 1986):

<table>
<thead>
<tr>
<th>Property</th>
<th>Collector freshness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sub-property</td>
<td>Commercial sample</td>
</tr>
<tr>
<td>$n-1$ interaction factors</td>
<td>1</td>
</tr>
<tr>
<td>Slope factor</td>
<td>$\alpha_7 = 1$</td>
</tr>
</tbody>
</table>
Final recovery factor \( \beta_7 = 1 \)
Cost factor \( 1 \)

Two other objects that refer to the same collector freshness are given below. They also assume that all other pivot conditions [concentration of collector \( 5 \times 10^{-3} \) mol/l; pH = 10.5; particle size -48 to +65 mesh (Tyler)] remain constant i.e. only collector freshness affects the process.

These objects are characterized by discrete sub-properties \( (n) \) with reference to the pivot value for each property. This definition permits the inclusion of knowledge such as *if* (property) *then* (response changes by a certain percentage or factor).

Regression adjustment objects. This type of object \( (k \) possible) differs from those given above in that the discrete sub-properties of the factor objects are replaced by a continuous regression equation (logarithmic, polynomial and power), as has already been done by various authors, albeit for various different models. It must, however, be noted that each object must still be independent of the other.

An example which implies independence is the definition of a functional relationship between the slow floating fraction \( \phi \) for the flotation of chalcopyrite at Mount Isa (Lynch et al., 1981, p. 137) and the collector addition \( z \) to the secondary ball mill \( (\phi = 1.069 - 0.00024 \cdot z \) for \( 1000 < z < 3000 \) ml/min). A relationship for a hypothetical set of data given later in Table 4 produces a relationship of \( \alpha = 1.8 - 4 \cdot \phi \), hence permitting the definition of the hypothetical regression adjustment object:

<table>
<thead>
<tr>
<th>Adjustment-object label</th>
<th>Collector ( (z ) ml/min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( k-1 ) interaction factors</td>
<td>1</td>
</tr>
<tr>
<td>Valid range ( (z) )</td>
<td>2600–3000 ml/min</td>
</tr>
<tr>
<td>Slope regression equation</td>
<td>( \alpha = 0.00096 \cdot z - 2.48 )</td>
</tr>
<tr>
<td>Final recovery regression equation</td>
<td>( \beta_8 = 1 )</td>
</tr>
<tr>
<td>Cost regression equation</td>
<td>1</td>
</tr>
</tbody>
</table>

### 2.3.3 Rules

Rules (Davis et al., 1977) have the general form: *if* (statement) *then* (action). These rules are used either to create an overall adjustment factor, i.e an adjustment rule, or to create a rule to perform diagnosis, i.e. the search rule.

**Adjustment rule.** This rule creates an overall adjustment factor from the individual objects of a specific pivot object. If each of the objects are independent from one another these may simply be multiplied:
if Sphalerite Ore A (class)
and Stope X (sub-class)
and $d_{50} = 80 \mu m: 11.5\% < 10 \mu m$
and Zn recovery = 11.8%
and Sodium ethyl xanthate = 14 g/t
and Copper sulphate = 140 g/t
then $\alpha = \alpha_1 \cdot \alpha_2 \cdot \alpha_3 \cdot \alpha_4 = 1 \cdot 1 \cdot 1 \cdot 1 = 1$
\[ \beta = 1 \]
\[ \text{Cost factor} = 1 \]

These rules are constructed after an interactive dialogue with the knowledge base, hence a batch-wise interviewing methodology is followed here (Puppe, 1988). This interactive dialogue is effected via user friendly interfaces.

**Search rule.** This rule forms part of process diagnosis. In the rule below, the Zn recovery, which has a value between 62.8 and 63.2% for Cells 1 to 4, is associated with a particular $\alpha$- and $\beta$-factor set. By a backward chaining algorithm the action part of the rule is determined.

if Zn recovery is $\geq 62.8\%$
and Zn recovery is $\leq 63.2\%$
and Cells 1-4 of rougher bank
then $d_{50} = 80 \mu m: 11.5\% < 10 \mu m$
\[ \text{Zn content of feed} = 11.8\% \]
\[ \text{Sodium ethyl xanthate} = 14 \text{ g/t} \]
\[ \text{Copper sulphate} = 140 \text{ g/t} \]

The user has control over the search accuracy of this search rule i.e. the tolerance on the percentages.

3. ALGORITHMIC DETAILS

The overall system structure is depicted by Fig. 2. The main algorithms involved in each of the three main sections of the proposed prototype system will be discussed under separate headings below.

3.1. Knowledge base

User friendly interfaces facilitate the dialogue with the knowledge base as well as the editing of the knowledge, as is usual for expert systems. An initial selection of a meta-class i.e. type of operation, confines subsequent queries
into the knowledge base to a particular area. A session would typically follow the following path:
— Selection of a class and sub-class and hence an appropriate pivot object.
— Subsequently the primary kinetic data are obtained.
— Calculation of slope–concentration data applying eq. 1.
— Calculation of $k''$ via eq. 5 and saving this value together with the final recovery in a record which can be applied to perform a shallow level search for fault diagnosis.
Selection of operating data via an interactive consultation with the knowledge base and subsequent calculation of adjustment factors.

- Set pointers to correct physical process and economic data, as well as data for fundamental correlations used to adjust process data on a fundamental basis.
- Selection of economic data as pertaining to the various adjustment objects.
- Adjustment of the slope–concentration data as a function of the selected adjustment objects for subsequent use in the simulation and diagnostic aspects of the programme via eq. 2.

Figure 3 depicts the selection of adjusted slope–load data which are applied during simulation and diagnosis.

### 3.2. Simulation of flotation cells and banks

The simulation of a batch flotation cell and a continuous flotation cell/bank is discussed below under two separate headings. Figure 4 illustrates the continuous interaction between the numerical procedure and the knowledge base during simulation.

**Batch flotation cell**

Appropriate adjusted kinetic data are obtained for a specific class and sub-class as discussed above. Subsequently the following activities are performed:

- Smoothing of the kinetic data i.e. for primary and secondary elements.
- Calculation of the relevant $k[C^e(t)]_n$ and validity ranges $C^e_n(t)$ to $C^e_{n+1}(t)$ for the given kinetic data via eq. 1.
- Numerical solution using the 4th order Runge-Kutta method (Press et al., 1989). This involves the selection of the relevant $k[C^e(t)]_n$ for a particular $C^e(t)$ value during each iteration step.
- Equation 3 may also be applied to solve this problem analytically.

**Continuous flotation cell/bank**

The steps which are followed to perform this activity after having selected an appropriate class and sub-class, include:

- Relevant pivot kinetic data (primary and secondary) are obtained and adjusted by the appropriate adjustment objects.
- Editing of process and operating cost data, as well as the additional adjustment of slope–concentration data via fundamental correlations.
- This knowledge is passed to the relevant simulation objects where it is used to simulate a continuous flotation cell or bank by the application of eq. 6. Both primary and secondary data are simulated.
Fig. 3. Flow diagram illustrating the selection of adjusted slope-concentration data which are applied during simulation and diagnosis.

— Changing concentrations of the secondary components may also be used to continuously generate new adjustment factors by continuously consulting appropriate regression adjustment objects, which are defined for this purpose.
— Final results are produced which are either reported to the printer, graphic screen or to the knowledge base.

3.3. Diagnostic capabilities

As was pointed out earlier, the diagnosis is limited here to the identification of the process conditions within flotation cells or banks. The elements of these diagnostic activities may be summarized under the following two headings:
3.3.1. Identification of a class and sub-class of a meta-class

This aspect of the diagnostic activity enables for instance the determination of the characteristics of an unknown ore. A file containing $k''$ and final
recovery values as well as the appropriate pointers to the knowledge base form the basis of this diagnostic method. The elements of this algorithm are:

- Enter the primary kinetic data of a particular process into the system.
- Establish the final recovery of the process and the calculation of \( k'' \) (eq. 5) for these data.
- Search through an index file, which contains all the \( k'' \) values for all the pivot objects, to determine which recoveries and subsequently which \( k'' \) values correspond within a predefined search accuracy.
- Presentation of three most likely pivot object candidates.
- Manual selection of any one of the three pivot object candidates as a final solution.

This search procedure covers all pivot-objects and can hence base its diagnosis on all available classes and sub-classes. The above procedure also enables the identification of a pivot object for application during simulation, as discussed earlier.

3.3.2. Identification of the process conditions within a flotation cell for a specific class and sub-class of a meta-class

The identification of the process conditions within batch and continuous cells/banks is discussed under two headings below:

**Batch cells.** The steps followed to identify the process conditions within a batch flotation cell are:

- Enter the pivot-object and associated adjustment objects for the class and sub-class of the system to be diagnosed and other associated knowledge.
- Subsequently, the kinetic data for the system to be diagnosed are entered into the system.
- These kinetic data are then compared to the pivot-object and deviations are quantified as \( \alpha \) and \( \beta \) values according to eq. 5.
- Calculated \( \alpha \) and \( \beta \) values are either compared to an index file which contains past simulation data (shallow level search), or a backward chaining algorithm is applied to determine suitable adjustment objects and their associated properties.
- The final results are reported to the monitor interface or, if required, to the printer.

These aspects are depicted by Fig. 5, which shows the flow diagram for the inference engine applied during the diagnosis of operating conditions within a batch flotation cell.

**Continuous cells/banks.** Two possible methods exist for the diagnosis of continuous flotation banks.

The first method is identical to that used for batch cells and involves rewriting concentration-cell data for a particular bank as concentration-reten-
tion time data and applying eqs. 1, 2 and 5. Dowling et al. (1986) applied this approach to compare different flotation models and flotation processes. These authors also used the approach of equating cell number with time and performing the same analysis as for concentration–retention time. If the method of these authors is followed then the kinetic data of the pivot objects of the knowledge base must be replaced by concentration–retention time (or concentration–cell number) data and the knowledge base must be constructed around these pivot data. In doing so, pivot objects are constructed for every flotation bank and all possible ore types.
A computationally more involved and lengthy second method, which applies batch kinetic data as a basis, would contain the following steps:
— The usual steps as applied above to obtain suitable knowledge and batch kinetic data for a particular class and sub-class.
— An iterative procedure is then initiated in which $\alpha$ and $\beta$ are adjusted until the final recovery which is attained on a particular flotation bank is produced. Meta-rules (Davis, 1980) assist in this search procedure. One meta-rule is based on the equation that defines the recovery from $N$ ideal CSTR's i.e.

$$\frac{C_e/C_0}{C_{e, \text{pivot}}/C_0} = (1 - R) + R(1 + k \cdot \tau)^{-N}$$

(16)

From this equation an average $k$ value may be estimated from the given value for $\frac{C_e/C_0}{C_{e, \text{pivot}}/C_0}$ and $\frac{C_e/C_{e, \text{observed}}}{C_0}$ respectively (quasi forward chaining).
— As for the batch process, once $\alpha$ and $\beta$ values have been determined, the backward chaining inference engine determines the appropriate operating conditions from the appropriate objects.

3.4. Optimization model

The link between the knowledge base and the optimization model is effected by use of eq. 11. The values for $(ym_{ej})^L$ and $(ym_{ej})^U$ are determined as follows:
— the expected operating conditions in a bank of cells are selected, which produce an adjusted kinetic model (eq. 3),
— this model is then used to simulate a bank of continuously operated flotation cells (eq. 6),
— from this simulation result a separation factor for each element $e$ at the bank can be determined,
— the above is repeated for all banks in the flotation plant,
— formulate a linear programming model for the valuable element $e$ (eqs. 7 to 14), and
— formulate a linear programming model for all other element(s), gangue, water etc. flow rates, subject to the structure produced by the linear programming model for the valuable element $e$.

Details will not be given here as this procedure has already been discussed earlier by Reuter and Van Deventer (1990).

4. MODEL VALIDATION

Various examples will be given in the subsequent section to illustrate the application of the developed models. The results will be presented in tabular form to illustrate numerical accuracy.
4.1. Batch flotation

The generality of the proposed kinetic model (eqs. 1 and 2) is illustrated by comparing its performance to that of three well known kinetic models viz. the classical first order rate equation, a first order rate equation with a rectangular distribution of floatabilities and a four parameter model, as well as practical kinetic flotation data.

4.1.1. Theoretical comparison

Two first order rate equations. A number of values for $k[C^e(t)]_n$ for the concentration–time data in Tables 2 and 3 are summarized in Table 1. These values are subsequently used to produce the results in the (b) columns of Tables 2 and 3.

**Table 1**

Summary of the calculated $k[C^e(t)]_n$ values according to eq. 1, using the data in column 1(a) in Tables 2 and 3

<table>
<thead>
<tr>
<th>$C^e$ Range</th>
<th>$k[C^e(t)]_n$</th>
<th>$t$ (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00 to 0.95</td>
<td>0.496</td>
<td>0-0.102</td>
</tr>
<tr>
<td>0.95 to 0.90</td>
<td>0.487</td>
<td>0-0.204</td>
</tr>
<tr>
<td>0.90 to 0.86</td>
<td>0.478</td>
<td>0-0.306</td>
</tr>
<tr>
<td>etc.</td>
<td>etc.</td>
<td>etc.</td>
</tr>
<tr>
<td>etc.</td>
<td>0.00</td>
<td>etc.</td>
</tr>
</tbody>
</table>

**Table 2**

Columns (a): Concentration–time data from the 1st order model with rectangular distribution of floatabilities. Data in column 1(a) were used to determine $k[C^e(t)]$ by eq. 1, which were subsequently used to predict values in columns (b) according to eq. 2. (a): $C^e/C^0 = 1 - R [1 - (1 - e^{-kt})/kt]$; (b): eqs. 1 and 2

<table>
<thead>
<tr>
<th>$R$ and $k$</th>
<th>1.0 and 1.0/min</th>
<th>0.5 and 0.1/min</th>
<th>0.1 and 4.0/min</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\beta$ and $\alpha$</td>
<td>1.0 and 1.0</td>
<td>0.5 and 0.1</td>
<td>0.1 and 4.0</td>
</tr>
<tr>
<td>$t$ (min)</td>
<td>1(a)</td>
<td>1(b)</td>
<td>2(a)</td>
</tr>
<tr>
<td>0.0</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>0.5</td>
<td>0.787</td>
<td>0.787</td>
<td>0.988</td>
</tr>
<tr>
<td>1.0</td>
<td>0.632</td>
<td>0.633</td>
<td>0.976</td>
</tr>
<tr>
<td>2.0</td>
<td>0.432</td>
<td>0.433</td>
<td>0.953</td>
</tr>
<tr>
<td>4.0</td>
<td>0.245</td>
<td>0.246</td>
<td>0.912</td>
</tr>
<tr>
<td>6.0</td>
<td>0.166</td>
<td>0.166</td>
<td>0.876</td>
</tr>
</tbody>
</table>
TABLE 3

Columns (a): Concentration-time data from the 1st order model. Data in column 1 (a) were used to determine \( k[C'(t)] \) by eq. 1, which were subsequently used to predict values in columns (b) according to eq. 2. (a): \( C'/C_0 = 1 - R[1 - e^{-kt}] \); (b): eqs. 1 and 2

<table>
<thead>
<tr>
<th>( R ) and ( k )</th>
<th>( 1.0 ) and ( 1.0/\text{min} )</th>
<th>( 0.5 ) and ( 0.1/\text{min} )</th>
<th>( 0.1 ) and ( 4.0/\text{min} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \beta ) and ( \alpha )</td>
<td>1.0 and 1.0</td>
<td>0.5 and 0.1</td>
<td>0.1 and 4.0</td>
</tr>
<tr>
<td>( t ) (min)</td>
<td>1 (a)</td>
<td>1 (b)</td>
<td>2 (a)</td>
</tr>
<tr>
<td>0.0</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>0.5</td>
<td>0.607</td>
<td>0.607</td>
<td>0.976</td>
</tr>
<tr>
<td>1.0</td>
<td>0.368</td>
<td>0.368</td>
<td>0.952</td>
</tr>
<tr>
<td>2.0</td>
<td>0.135</td>
<td>0.136</td>
<td>0.909</td>
</tr>
<tr>
<td>4.0</td>
<td>0.018</td>
<td>0.018</td>
<td>0.835</td>
</tr>
</tbody>
</table>

TABLE 4

Columns (a): Concentration-time data generated by a 4-parameter model with \( R=1, k_1=1 \text{ min}^{-1} \) and \( k_2=0.1 \text{ min}^{-1} \). Data in column 1 (a) were used to determine \( k[C'(t)] \) by eq. 1, which were subsequently used to predict values in columns (b) according to eq. 2. (a): \( C'/C_0 = (1-\phi)e^{-1.0t} + \phi e^{-0.1t} \); (b): eqs. 1 and 2

<table>
<thead>
<tr>
<th>( \phi )</th>
<th>0.2</th>
<th>0.1</th>
<th>0.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \beta ) and ( \alpha )</td>
<td>1.0 and 1.0</td>
<td>1.0 and 1.4</td>
<td>1.0 and 0.6</td>
</tr>
<tr>
<td>( t ) (min)</td>
<td>1 (a)</td>
<td>1 (b)</td>
<td>2 (a)</td>
</tr>
<tr>
<td>0.0</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>1.0</td>
<td>0.475</td>
<td>0.479</td>
<td>0.422</td>
</tr>
<tr>
<td>2.0</td>
<td>0.272</td>
<td>0.274</td>
<td>0.204</td>
</tr>
<tr>
<td>4.0</td>
<td>0.149</td>
<td>0.149</td>
<td>0.084</td>
</tr>
<tr>
<td>6.0</td>
<td>0.112</td>
<td>0.112</td>
<td>0.057</td>
</tr>
<tr>
<td>8.0</td>
<td>0.090</td>
<td>0.090</td>
<td>0.045</td>
</tr>
<tr>
<td>10.0</td>
<td>0.074</td>
<td>0.074</td>
<td>0.037</td>
</tr>
<tr>
<td>12.0</td>
<td>0.060</td>
<td>0.060</td>
<td>0.030</td>
</tr>
<tr>
<td>14.0</td>
<td>0.049</td>
<td>0.049</td>
<td>0.025</td>
</tr>
</tbody>
</table>

It is clear from the results summarized in Tables 2 and 3 that the model proposed here can predict results identical to those of the two different theoretical models.

Four parameter rate equation. A comparison of the proposed two parameter model with a typical four parameter model e.g. \( C/C_0 = 1 - R[1 - (1-\phi)\exp(-k_1t) - \phi \exp(-k_2t)] \) (Lynch et al., 1981), shows that the correspondence is not exact, unless if \( \phi \) remains constant and the \( k \)'s vary by a constant factor. The values in the (b) columns of Table 4 reflect a rea-
sonable correspondence with the theoretical values in the (a) columns for different $\phi$ values.

The relationship between $\alpha$ and $\phi$ in Table 4 is $\alpha = 1.8 - 4 \cdot \phi$ (derived from only the three values in Table 4 for illustrative purposes). This relationship was used to construct the hypothetical collector adjustment-object for Mount Isa mentioned earlier in the section on regression adjustment objects. Note that this example is merely illustrative, and shows how previously determined relationships i.e. for $\phi$ in this case, can be rewritten for $\alpha$.

4.1.2. Practical comparison

Two sets of data will be discussed here.

**Effect of a depressant.** The data applied for this example involves the recovery of galena in favour of the gangue sulphide mineral marmatite from a Broken Hill ore (Lynch et al., 1981, p. 144). The data in column 1 (a) of Table 5 represents the pivot data with a ZnSO$_4$ addition of 790 g/tonne.

The adjustment rule that has been created from the relevant pivot object to predict the entries in column 2 (b) of Table 5 is:

if Broken Hill Galena Ore A (class)
and Stope X (sub-class)
and (other process conditions)
and etc.
and ZnSO$_4$ = 250 g/t

then $\alpha = \alpha_1 \cdot \ldots \alpha_n = 1.33$
$\beta = \beta_1 \cdot \ldots \beta_n = 1.022$
Cost factor = 1

Table 5 displays a reasonable correspondence between practical and simulated data.

**TABLE 5**

Columns (a) give the practical data obtained from p. 144, Lynch et al. (1981). Data in column 1 (a) were used to determine $k(C^t(t))$ by eq. 1, which were subsequently used to predict values in columns (b) according to eq. 2 and the given adjustment objects.

<table>
<thead>
<tr>
<th>$t$ (min)</th>
<th>ZnSO$_4$ 790 g/t</th>
<th>ZnSO$_4$ 250 g/t</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1(a)</td>
<td>1(b)</td>
</tr>
<tr>
<td>0.00</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>0.50</td>
<td>0.380</td>
<td>0.380</td>
</tr>
<tr>
<td>1.50</td>
<td>0.180</td>
<td>0.180</td>
</tr>
<tr>
<td>2.50</td>
<td>0.100</td>
<td>0.100</td>
</tr>
<tr>
<td>4.00</td>
<td>0.080</td>
<td>0.080</td>
</tr>
<tr>
<td>7.00</td>
<td>0.061</td>
<td>0.061</td>
</tr>
</tbody>
</table>
TABLE 6

Columns (a) give the practical data obtained from Ackerman et al. (1988). Data in column 1(a) were used to determine \(k[C'(t)]\) by eq. 1, which were subsequently used to predict values in columns (b) according to eq. 2 and the given adjustment objects.

<table>
<thead>
<tr>
<th>(t) (min)</th>
<th>Commercial 1(a)</th>
<th>Fresh sample 2(a)</th>
<th>Commercial 105 3(a)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>0.10</td>
<td>0.780</td>
<td>0.560</td>
<td>0.920</td>
</tr>
<tr>
<td>0.50</td>
<td>0.350</td>
<td>0.140</td>
<td>0.630</td>
</tr>
<tr>
<td>1.00</td>
<td>0.190</td>
<td>0.050</td>
<td>0.350</td>
</tr>
<tr>
<td>1.50</td>
<td>0.100</td>
<td>0.020</td>
<td>0.250</td>
</tr>
<tr>
<td>2.00</td>
<td>0.050</td>
<td>0.010</td>
<td>0.210</td>
</tr>
</tbody>
</table>

**Freshness of collector.** The effect of freshness of sodium isopropyl xanthate on the recovery of chalcocite by batch flotation is demonstrated in the example below (Ackerman et al., 1986). The freshly synthesized sample of collector forms the pivot data. The adjustment objects for two commercial samples of different freshness are:

<table>
<thead>
<tr>
<th>Property</th>
<th>Collector freshness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sub-property</td>
<td>Fresh sample</td>
</tr>
<tr>
<td>(n-1) interaction factors</td>
<td>1</td>
</tr>
<tr>
<td>Slope factor</td>
<td>(\alpha_{10}=1.77)</td>
</tr>
<tr>
<td>Final recovery factor</td>
<td>(\beta_{10}=1.042)</td>
</tr>
<tr>
<td>Cost factor</td>
<td>1</td>
</tr>
</tbody>
</table>

and

<table>
<thead>
<tr>
<th>Property</th>
<th>Collector freshness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sub-property</td>
<td>Commercial sample (105 days)</td>
</tr>
<tr>
<td>(n-1) interaction factors</td>
<td>1</td>
</tr>
<tr>
<td>Slope factor</td>
<td>(\alpha_{11}=0.47)</td>
</tr>
<tr>
<td>Final recovery factor</td>
<td>(\beta_{11}=1)</td>
</tr>
<tr>
<td>Cost factor</td>
<td>1</td>
</tr>
</tbody>
</table>

The value of \(\beta_{10}\) is simply obtained from the ratio of 99/95 (the final recoveries given later in Table 8) and \(\alpha_{10}\) from the ratios of the time-weighted slopes 0.4274/0.2298 \(\cdot\) \(\beta_{10}\) (given later in Table 8). The value of \(\beta_{11}\) being 1.0 indicates that the final recovery is also equal to 95% and not 79% as pointed out by Ackerman et al. (1986). As the kinetic data for this 105 days old collector sample have obviously not reached its final recovery, it was not possible to determine the time-weighed slope. Hence \(\alpha_{11}=0.47\) was determined by trial-and-error.
The adjustment rule that has been created from the relevant pivot object to predict the entries in column 2(b) of Table 6 is:

if Chalcocite Ore A (class) and Stope X (sub-class) and Particle size: -45 to +65 mesh (Tyler) and pH = 10.5 and Collector concentration = 5 \times 10^{-5} \text{ mol/l} and Collector freshness = Fresh sample
then \begin{align*}
\alpha &= \alpha_1 \cdot \alpha_2 \cdot \alpha_3 \cdot \alpha_4 = 1 \cdot 1 \cdot 1 \cdot 1.849 = 1.849 \\
\beta &= \beta_1 \cdot \beta_2 \cdot \beta_3 \cdot \beta_4 = 1 \cdot 1 \cdot 1 \cdot 1.042 = 1.042 \\
\text{Cost factor} &= 1
\end{align*}

4.2. Continuous flotation

In order to illustrate the application of this aspect of the proposed KBS to a practical system, the flotation of sphalerite is considered (Frew and Davey, 1988).

In their example, Frew and Davey (1988) could describe the flotation by simple first order kinetics. The pivot-object for cells 1-4 was defined by means of the factor adjustment objects as explained earlier (experiment 1, as defined by Frew and Davey, 1988). The kinetic data slot contains first order kinetic data with a rate constant of 0.11 min$^{-1}$ as given by Frew and Davey, 1988). This implies that the final recovery is zero ($\beta$ always equal to 1 for this example) and that the computed rate constant $k[C^*(t)]_n$ has a constant value of 0.11 min$^{-1}$.

Since the adjustment objects which are considered for this example are dependent on one another, adjustment object groups must be defined if their values deviate from the pivot conditions. These are for cells 1-4 and experiment 2 as defined by Frew and Davey (1988):

<table>
<thead>
<tr>
<th>Property</th>
<th>%Zn in feed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sub-property</td>
<td>15.0%</td>
</tr>
<tr>
<td>Interaction: collector</td>
<td>0.884</td>
</tr>
<tr>
<td>Interaction: activator</td>
<td>0.884</td>
</tr>
<tr>
<td>$n-3$ interaction factors</td>
<td>1</td>
</tr>
<tr>
<td>Slope factor</td>
<td>$\alpha_{12} = 1.45$</td>
</tr>
<tr>
<td>Final recovery factor</td>
<td>$\beta_{12} = 1$</td>
</tr>
<tr>
<td>Cost factor</td>
<td>1</td>
</tr>
<tr>
<td>Property</td>
<td>Collector</td>
</tr>
<tr>
<td>Sub-property</td>
<td>32 g/t</td>
</tr>
<tr>
<td>Interaction: %Zn in feed</td>
<td>0.884</td>
</tr>
</tbody>
</table>
Interaction: activator 0.884
n - 3 interaction equations 1
Slope factor $\alpha_{13} = 1.45$
Final recovery factor $\beta_{13} = 1$
Cost factor 1

Property Activator
Sub-property 180 g/t
Interaction: %Zn in feed 0.884
Interaction: collector 0.884
n - 3 interaction equations 1
Slope factor $\alpha_{14} = 1.45$
Final recovery factor $\beta_{14} = 1$
Cost factor 1

The group characteristics of these adjustment objects are indicated by their identical adjustment and interaction slots. These interaction slots only serve to indicate that there is an interaction and their values only hold for the group as a whole. The magnitude of these slots are such that when the three adjustment objects are multiplied, their total value equals the slope factor of one adjustment object of the group. From the given pivot object for this example the following adjustment rule could be defined for cells 1–4 of experiment 2 as defined by Frew and Davey, 1988):

**Cells 1–4**

if Sphalerite Ore A (class)
and Stope X (sub-class)
and $d_{50}=80 \mu m: 11.5\% < 10 \mu m$
and Zn content of feed = 15%
and Sodium ethyl xanthate = 32 g/t
and Copper sulphate = 180 g/t
then slope factor $= \alpha = \alpha_1 \cdot \alpha_{12} \cdot \alpha_{13} \cdot \alpha_{14} = 1.45$
final recovery factor $= \beta = \beta_1 \cdot \beta_{12} \cdot \beta_{13} \cdot \beta_{14} = 1$
cost factor = 1

Instead of defining the above adjustment object group, it is also possible to define a new pivot-object, with the above three objects of the group forming the pivot objects for cells 1–4 for experiment 2, as defined by Frew and Davey, 1988). The kinetic data, which correspond to experiment 2, must be included in such a pivot object.

By defining similar pivot objects (any of the two methods) for cell groups
5–7 and 8–10 suitable adjustment factors could be defined. In the example described here, experiment 1 was always taken as the pivot condition.

By assuming ideal flow, by using the appropriate unit-operation-object, and by applying eqs. 1–3, the given retention times (Frew and Davey, 1988) and the adjustment objects for the respective cell groups, the results summarized in Table 7 could be produced for experiment 1 of Frew and Davey (1988).

A similar approach must be followed for all other relevant components, such as galena, pyrrhotite, non-sulphide gangue and water in the above example. For each of these components, an associated adjustment-object must be defined. For the gangue under the prevailing conditions, for example, the rate of water recovery is a factor \( a = 0.055 \) of that of the sphalerite in cells 1–4, 0.027 in cells 5–10, and for the gangue \( a = 0.012 \) for all cells. This information is captured by appropriate adjustment-objects. The results for these simulations are also presented in Table 7.

4.3. Optimization of a flotation circuit

The circuit optimization is conducted using the data of Frew and Davey (1988). This example serves only to illustrate how the link between the knowledge base and optimization model is created, but is not a detailed example as this has been discussed by Reuter and Van Deventer (1990).

The simple objective function for this example attempts to maximize the recovery of the valuable ZnS in the circuit and its weight is merely a price $/mass for the valuable ZnS. From the simulation results in Table 7 for the sphalerite flotation, the following linear programming model could be defined for rougher, scavenger and cleaner banks \((i = 3 \text{ and } j = 3)\):

\[
OBJ = 100a_{ZnS}L
\]

\[
\sum_{j, i \neq j} rm_{ij}^{ZnS} + \sum_{j, i \neq j} ry_{ij}^{ZnS} + u_{ZnS} = m_{ZnS} + y_{ZnS}
\]
\[ \sum_{j \neq i} r_m ZnS^+_ij + b_{ZnSj} = m_{ZnSj} \]
\[ \sum_{j \neq i} r_y ZnS^-ij + a_{ZnSj} = y_{ZnSj} \]
\[ 0 \leq a_{ZnS3} \leq 20 \]
\[ 0 \leq b_{ZnS2} \leq 20 \]
\[ 2.04 \leq u_{ZnS1} \leq 2.04 \quad (\text{assay data of Frew and Davey, 1988}) \]
\[ 0m_{ZnS1} \leq y_{ZnS1} \leq 11.98m_{ZnS1} \quad (\text{Table 7}) \]
\[ 0m_{ZnS2} \leq y_{ZnS2} \leq 3.8m_{ZnS2} \quad (\text{Table 7}) \]
\[ 0m_{ZnS3} \leq y_{ZnS3} \leq 90m_{ZnS3} \quad (\text{assay data of Frew and Davey, 1988}) \]
\[ a_{ZnS1} = 0; \quad a_{ZnS2} = 0; \quad \text{(only concentrate from bank 3)} \]
\[ b_{ZnS1} = 0; \quad b_{ZnS3} = 0; \quad \text{(only tailings from bank 2)} \]
\[ u_{ZnS2} = 0; \quad u_{ZnS3} = 0 \quad \text{(only feed to bank 1)} \]
\[ r_m ZnS^-_{32} = 0; \quad r_m ZnS^-_{12} = 0; \quad \text{(no recycle of final tailings)} \]
\[ r_y ZnS^-_{23} = 0; \quad r_y ZnS^-_{13} = 0; \quad \text{(no recycle of final concentrate)} \]
\[ r_m ZnS^-_{31} = 0; \quad \text{(no recycle of rougher tailings to cleaner)} \]

The constraint for the two cleaners, which were grouped here into one bank, could be estimated from the given assay data of Frew and Davey (1988), as kinetic data for these banks were not supplied. Note that of the twelve recycle possibilities five have been eliminated due to process constraints.

The result of this simulation is the depicted circuit structure (Fig. 6) for a maximization in the sphalerite recovery. The recovery for this structure is 98.4%. However, if the concentrate recycle of the scavenger is forced to report to the rougher, i.e. the structure as given by Frew and Davey (1988), the recovery drops to 98.2%. To determine the optimal grade, a similar model

![Fig. 6. Circuit structure and sphalerite flow rates in t/h for the optimal circuit produced by the KBS.](https://scholar.sun.ac.za)
can be developed for the other elements to minimize their flow for the given optimal structure determined for the sphalerite (Reuter and Van Deventer, 1990).

From the above flow rates it is clear that they do not differ much from those given by Frew and Davey (1988) for their experiment 1. However, the structure differs in that the concentrate reports to the cleaner and is not recycled to the rougher.

4.4. Fault-diagnosis

Fault-diagnosis is discussed under three headings viz. the identification of a class and sub-class of a meta-class, fault-diagnosis for batch flotation cells and fault-diagnosis of continuous flotation banks. As was already pointed out, diagnosis implies an identification of the process conditions within the flotation unit operations. Each of these aspects will be discussed under separate headings.

4.4.1. Identification of a class and sub-class of a meta-class

The data of Ackerman et al. (1986) referring to collector freshness will be used to illustrate this feature of the proposed system. As indicated by the adjustment rule for this example (given earlier), the class and sub-class are Chalcocite Ore A and Stope X respectively. The pivot kinetic data for this example are given in column 1 (a) of Table 6 and the resultant slope-concentration and time-weighted slope data for these data are summarized in columns 2 to 4 of Table 8.

Consider a meta-class that contains various chalcocite flotation classes and sub-classes. For argument’s sake, consider the kinetic data of column 1 (a) of Table 6 to be of a class and sub-class yet to be determined. As explained under

<table>
<thead>
<tr>
<th>TABLE 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slope-concentration data for the kinetic flotation pivot data (commercial collector) and fresh collector sample respectively (Ackerman et al., 1986)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(\Sigma=0.4274)</th>
<th>(\Sigma=0.2298)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\Sigma=0.4274)</td>
<td>(\Sigma=0.2298)</td>
</tr>
<tr>
<td>(n)</td>
<td>Time (min)</td>
</tr>
<tr>
<td>1</td>
<td>0.000</td>
</tr>
<tr>
<td>2</td>
<td>0.020</td>
</tr>
<tr>
<td>3</td>
<td>0.040</td>
</tr>
<tr>
<td>4</td>
<td>0.060</td>
</tr>
<tr>
<td>5</td>
<td>0.080</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>100</td>
<td>2.000</td>
</tr>
</tbody>
</table>
the heading "Diagnostic capabilities", the first step to determine what class and sub-class the kinetic data represents is to determine the time-weighted slope and final recovery. These are respectively 0.4274 and 95%. The search procedure would subsequently search through an index file that contains the time-weighted slope and final recovery data for all available classes and subclasses of the meta-class. A search rule would then obviously have pointed to the class and sub-class Chalcocite Ore A and Stope X respectively (Tables 6 and 8). This search would also have produced the process conditions under which the pivot kinetic data were determined via the pivot adjustment objects, all with a value of 1.

The above process may also be applied to perform process identification. If the method of defining a pivot-object for every data/knowledge set is opted for, rather than defining adjustment objects which contain values higher or lower than 1.0, the above method would also produce process conditions via the pivot adjustment objects.

4.4.2. Identification of the process conditions within a batch flotation cell

Also consider the flotation data summarized in Tables 6 and 8 for this example. For argument's sake, the aim in this example is to determine why the recovery of chalcocite increased (data columns 2, Table 6) relative to the pivot data (data columns 1, Table 6).

The algorithm for this was discussed earlier. It commences by determining the time-weighted slope and final recovery of the kinetic data to be identified. These values are respectively 0.2298 and 99% respectively. From these values \( \alpha \) and \( \beta \) values are determined which are 1.77 and \( 99/95 = 1.042 \) respectively. Subsequently, a search through all available adjustment objects would have pointed out that most probably a fresh collector was used (see \( \alpha_{10} \) and \( \beta_{10} \) in the section with heading "Freshness of collector").

4.4.3. Identification of the process conditions within a continuous flotation bank

The two methods outlined in the section with heading "Diagnostic capabilities" will be discussed separately below.

Method 1. This method does not use eq. 6 to define the dynamics of a flotation bank, but applies eqs. 1 and 2 (as for batch flotation) to describe the kinetics of the flotation bank as a whole. This approach implies that the pivot objects are defined accordingly, i.e. they do not include batch kinetic data but contain concentration-retention time (or cell number) data. All adjustment objects are defined accordingly with reference to these data. One would obviously not be able to perform process identification in batch cells with these data.

Table 9 summarizes the effect of various \( \alpha \) and \( \beta \) values on the recovery-
Retention time data given for the Tennessee copper rougher circuit described by Dowling et al. (1986). Note that the induction period reflected by these data is easily modelled by eq. 1. This would of course not have been possible when using the other standard kinetic models, except if these could be modified accordingly. It is exactly this induction period which could be decisive during process identification (eq. 5), and which would be neglected by the other models.

If this approach is to be followed for diagnosis, the same algorithm could be used for batch reactors. If for argument’s sake, the mentioned rougher copper flotation bank would produce the kinetic data in the last column of Table 9, then via eq. 5 and the final recovery with respect to the pivot data, \( \alpha = 2 \) and \( \beta = 0.9 \) would have been generated. These values are linked via adjustment objects to the relevant knowledge, from which the process conditions could be identified. Subsequently, remedial action could be undertaken on a plant.

**Method 2.** Whereas the batch fault-diagnosis procedure as applied in method 1 uses eq. 5 to determine an estimate of \( \alpha \), eqs. 1–3, 6 and 16 are applied here as the basis of diagnostic procedures. This implies that batch kinetic data form the basis of this method.

The adjustment object defined for the continuous flotation example is also used in this example. For argument’s sake consider the grade of Zn in the first four cells to be 27%. The question to be answered is what operating conditions in the flotation cells would produce this recovery?

Initially a search tolerance is given as an input, as reflected by the statement.

**TABLE 9**

<table>
<thead>
<tr>
<th>Time (min)</th>
<th>Pivot data</th>
<th>( \alpha = 1.0 )</th>
<th>( \alpha = 0.5 )</th>
<th>( \alpha = 1.0 )</th>
<th>( \alpha = 2.0 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>( \beta = 1.0 )</td>
<td>( \beta = 1.0 )</td>
<td>( \beta = 0.8 )</td>
<td>( \beta = 0.9 )</td>
</tr>
<tr>
<td>0.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>1.600</td>
<td>0.987</td>
<td>0.987</td>
<td>0.997</td>
<td>0.989</td>
<td>0.976</td>
</tr>
<tr>
<td>2.400</td>
<td>0.980</td>
<td>0.980</td>
<td>0.990</td>
<td>0.984</td>
<td>0.964</td>
</tr>
<tr>
<td>4.000</td>
<td>0.967</td>
<td>0.967</td>
<td>0.983</td>
<td>0.974</td>
<td>0.929</td>
</tr>
<tr>
<td>8.000</td>
<td>0.921</td>
<td>0.921</td>
<td>0.967</td>
<td>0.937</td>
<td>0.703</td>
</tr>
<tr>
<td>12.000</td>
<td>0.804</td>
<td>0.804</td>
<td>0.949</td>
<td>0.843</td>
<td>0.459</td>
</tr>
<tr>
<td>16.000</td>
<td>0.670</td>
<td>0.670</td>
<td>0.922</td>
<td>0.736</td>
<td>0.309</td>
</tr>
<tr>
<td>20.000</td>
<td>0.517</td>
<td>0.517</td>
<td>0.876</td>
<td>0.614</td>
<td>0.221</td>
</tr>
<tr>
<td>36.000</td>
<td>0.190</td>
<td>0.190</td>
<td>0.599</td>
<td>0.352</td>
<td>0.127</td>
</tr>
<tr>
<td>52.000</td>
<td>0.055</td>
<td>0.055</td>
<td>0.363</td>
<td>0.244</td>
<td>0.127</td>
</tr>
<tr>
<td>64.000</td>
<td>0.030</td>
<td>0.030</td>
<td>0.233</td>
<td>0.224</td>
<td>0.127</td>
</tr>
<tr>
<td>72.000</td>
<td>0.030</td>
<td>0.030</td>
<td>0.191</td>
<td>0.224</td>
<td>0.127</td>
</tr>
</tbody>
</table>
part of the search rule below. Subsequently the diagnostic procedure would have applied eq. 16 to estimate an $\alpha$ value relative to the pivot object. This equation produces a value of $k=0.1$ from the given retention time (2.8 min) and recovery data ($R=1$), and $k=0.145$ for the process to be diagnosed (retention time = 2.68 min). This implies that the pivot data must be adjusted by a factor $\alpha=1.45$. If it is assumed that $\beta=1$, eq. 2 in association with eq. 6 and other relevant data could be used to produce a final Zn grade of 27% for the first four cells of the rougher bank. This result would then be tested against the statement part of the search rule below, which is true in this case. At this point a backward chaining algorithm determines which adjustment objects correspond to the $\alpha=1.45$ and $\beta=1$ values. In this case the adjustment object group given earlier would be applicable here, and is given in the action part of the rule below.

if Zn-grade $\geq 26.8\%$
and Zn-grade $\leq 27.2\%$
and Cells 1–4 of rougher bank
then $d_{50}=80$ µm: $11.5\% < 10$ µm
Zn content of feed = 15%
Sodium ethyl xanthate = 32 g/t
Copper sulphate 180 g/t

If the final recovery grade lies outside the range given by the statement part of the above rule, a new estimate for $\alpha$ and $\beta$ is calculated, but this time it is increased or decreased (dictated by rules) by a small factor. This is repeated until the final recovery lies within the defined range.

Note that the same method discussed for method 1 above could also have been applied here, with the difference that the concentration–retention time data are contained in the pivot object.

5. CONCLUSIONS AND SIGNIFICANCE

The simulation, optimization and identification of batch and continuous flotation processes by using an object-orientated knowledge based system and linear programming was discussed in this paper. The central theme is the development of a kinetic model that can be adjusted by two adjustment factors $\alpha$ and $\beta$ respectively. These should not be affected by variances that occur during least squares parameter fitting, hence a direct calculation of appropriate rate variables and final recoveries from the kinetic data was opted for. Therefore, in the form that this model and its “parameters” are presented they are very useful for process identification (or diagnosis).

The simulation, identification and optimization results serve to illustrate the following regarding this modelling approach:
— It was shown how a generalized non-linear first order rate equation (two parameter model) can be used to approximate three different well-known theoretical flotation models and a variety of practical flotation data. Although not shown here, it has been demonstrated (Reuter and Van Deventer, 1991) that this same approach could be applied to approximate leaching, adsorption and pyrometallurgical reduction reactions. In all cases the approximation was almost exact.

— This general form of the rate equation permits the definition of a generalized knowledge base which can interact with the rate variable and the final recovery via objects. This permits the definition of a well-defined kinetic model for any type of flotation process under consideration.

— The application of this kinetic model was used to simulate the recovery of sphalerite, gangue and water in continuous rougher and scavenger banks by taking the process conditions within a cell into consideration. The results in this case compare well with industrial data.

— The interaction of the data produced by the simulation and process identification activities and a linear programming optimization model to produce optimal circuits, was demonstrated. A circuit was produced giving approximately the same results as the industrial flotation plant being simulated, the difference being that the concentrate of the scavenger reports to the cleaner and not to the rougher. If this stream is forced to report to the rougher by a suitable constraint (as in the real plant), the recovery drops by 0.2%.

— Fault-diagnosis/process identification can also be performed by utilizing the adjustment information $\alpha$ and $\beta$ of the generalized kinetic model. By comparing this information to the pivot-data, the model can suggest the possible conditions within batch or continuous reactors under consideration.

From the results of the knowledge based system, it is evident that the KBS could render an ill-defined flotation process well-defined, and that this approach is able to perform simulations and process identification that standard simulation programmes are not capable of doing.

6. SYMBOLS USED

$\alpha, \beta, c, d$ Constants (–)
$\alpha_{ej}, b_{ej}$ Mass flow rates of final concentrate and final tailings of element $e$ from bank $j$ respectively (t h$^{-1}$)
$(\alpha_{ej})^U$ Upper bound on $\alpha_{ej}$ [Lower bound $(\alpha_{ej})^L$] (t h$^{-1}$)
$(b_{ej})^U$ Upper bound on $b_{ej}$ [Lower bound $(b_{ej})^L$ = ] (t h$^{-1}$)
$C^e_i, C^e_i, C^n_i$ Concentration in a batch reactor, concentration in stage $i$ and at data point $n$ of element $e$ respectively (ppm)
SIMULATION AND IDENTIFICATION OF FLOTATION PROCESSES WITH A KNOWLEDGE BASED MODEL

$C_0^e$ Concentration of element $e$ at start of batch test or in feed to continuous plant (ppm)

$F_{p,in}$ Pulp flowrate in and $F_{p,out}$ out of a stage (m$^3$ min$^{-1}$)

$k[C^e(t)]_n$ Rate variable as a function of $C^e(t)$ at data point $n$ (min$^{-1}$)

$k_f, k_s$ Flotation rate constants for fast and slow floating fractions respectively (min$^{-1}$)

$k'$ rate constant normalised with respect to $C_0^e$ (min$^{-1}$)

$m_{ej}, y_{ej}$ Mass flow rates of tailings and concentrate of element $e$ at bank $j$ respectively (t h$^{-1}$)

$N$ Number of ideal CSTR’s in a flotation bank

OBJ Objective function used to maximise the revenue generated by a flotation plant

$rm_{ij}^e, ry_{ij}^e$ Mass flow rates in the tailings and concentrate recycle streams of element $e$ from bank $j$ to $i$ respectively (t h$^{-1}$)

$r$ Rate equation (min$^{-1}$)

$R$ Final recovery (—)

$(rm_{ij}^e)^L$ Lower bound on the tailings recycle flow of element $e$ between bank $j$ and $i$ [(rm$_{ij}^e)^U$ upper bound]

$Sh, Re, Sc$ Sherwood, Reynolds and Schmidt numbers (—)

$t$ time variable (min)

$u_{ej}$ Mass flow rates of feed of element $e$ to bank $j$ (t h$^{-1}$)

$(u_{ej})^L$ Lower bound on $u_{ej}$ [(u$_{ej})^U$ upper bound] (t h$^{-1}$)

$V_{act}$ Active stage volume (m$^3$)

$w_e$ Price weight of element $e$ recovered in the concentrate.

$(ym_{ej})^L$ Lower bound on the separation factor of element $e$ at bank $j$ [(ym$_{ej})^U$ is upper bound]

$z$ Collector addition rate (ml/min)

$\delta t$ Time increment (min)

$\Phi$ Fraction of pulp short-circuiting reactor stage (—)

$\phi$ Slow floating fraction in flotation

$\alpha, \beta$ Adjustment factors/equations (—)

REFERENCES


SIMULATION AND IDENTIFICATION OF FLOTATION PROCESSES WITH A KNOWLEDGE BASED MODEL


A knowledge based system for the simulation and optimization of metallurgical plants

by M.A. REUTER and J.S.J. VAN DEVENTER

INTRODUCTION

Numerous factors influence the flotation of valuable components in a flotation cell [1-2]. Although a number of fundamental models have been proposed for flotation, the parameters of these models usually lump together numerous process variables [1]. Hence, little information regarding the process can be extracted from these parameters. Therefore, it is the objective of this paper to propose a knowledge-based system (KBS) model which would enable the inclusion of all possible flotation variables, e.g. frother & collector addition, pH, aeration rate, etc. to render the as yet ill-defined flotation process well-defined. This is accomplished here by using a hybrid KBS approach, i.e. both qualitative (experiential and heuristic) knowledge and quantitative (equation oriented) knowledge are used to describe the kinetics of flotation.

The modelling of various ill-defined chemical engineering problems have been performed by using KBS techniques. Most of these applications are, however, confined to steady-state circuit design [3-5], the selection of equipment, processes or reagents e.g. DECADE for the selection of catalysts [6-7], synthesis of operating procedures [3-8-10], malfunction diagnosis [11-14] and control synthesis [3-15-17]. Little emphasis has been placed to date on the application of these techniques to the modelling of mineral processing systems [18]. Although the dynamic simulation of ill-defined problems by the use of these techniques has found little application in the literature, qualitative simulation has been explored in this regard [19]. It has already been shown how a KBS and a dynamic equation oriented system can be combined to simulate CIP and CIL systems [20-21].

In the above-mentioned systems various architectures have been applied to formalize the knowledge e.g. object oriented [3-14], frame oriented [5], hybrid systems in a blackboard [6-7], hierarchical [8-11] and goal-tree-success-tree [17] systems. Of these systems the object oriented approach was used to define a KBS that can embrace both the dynamics of a process as well as do process malfunction diagnosis on the basis of the chemical and physical environment within the reactor. Although this model could be applied to leaching, adsorption or pyrometallurgical processes, it is applied here to simulate the rougher and scavenger banks of a sphalerite flotation process. It is also demonstrated how the KBS system can suggest an optimal flotation circuit by coupling the dynamic equations and the knowledge base to a linear programming model [22].

KNOWLEDGE REPRESENTATION

The representation of shallow (experiential and heuristic) and deep (equation oriented) knowledge for application in knowledge based systems is probably one of the most important aspects of this modelling approach. In the approach discussed here, the batch flotation concentration-time recovery data of the valuable element, as well as all other species involved, form the basis of the knowledge base. These data are transformed into a useful form by a generalized kinetic model, the rate constant also being a function of all the effects of the flotation process variables. The organization of the data in this manner permits the interaction between the batch flotation concentration-time data, process variables, kinetic model and mathematical modelling. This may be realized by knowledge representation techniques such as facts, objects (frames), rules and functions [19]. These techniques are used here in the following manner:

- Facts may include, for example, the type of reaction under consideration, be it leaching, adsorption or reduction, flotation or the type of reactor system (batch, CSTR, counter-current etc.).
- Each Object (Frame) is structured so as to include all relevant information regarding a particular process variable and its effect on the process. An example here is the pivot-object, that defines both shallow and deep knowledge regarding a process under consideration (it refers here to the said sphalerite flotation process with the conditions of Experiment 1 [2] forming the pivot-conditions).
- Type of flotation system
- Distinguishing characteristic
- n Factor adjustment-objects
- k Regression-objects
- Kinetic data [sets of kinetic data e.g. change of reagents]:
  - Time T
  - C
- Type of operation:
  - Batch/Continuous
  - Function-objects & anti-operation-object
- Cost data

This object comprises of various other objects which include amongst others the following unit-operation-object:

Unit operation:

Reactors

Mineral content of feed
% solids in feed
Retention time of cells 1-4
Number of reactors in series
Feed flowrate of the pulp
% short circuiting of feed
% dead volume of reactor

1. Institute für Metallurgie und Elektrochemie, RW-Technische Hochschule, Aachen, Germany.
2. Department of Metallurgical Engineering, University of Stellenbosch, Stellenbosch, 7600, South Africa.

DECEMBRE 1992
or the adjustment-object comprising of \( n \) possible objects with \( m \) sub-divisions which define the effects of the deviation of process variables on the pivot process conditions:

- **Adjustment-object label**: Particle size
- **Sub-division of adjustment-object**: \( d_{50} = 80 \text{ mm} \)
- **Interaction factors**: \( I \)
- **Slope factor**: \( l = \alpha \)
- **Final recovery factor**: \( l = \beta \)
- **Cost factor**: \( c \)

or the \( k \) regression-objects (similar to the object above) that define the effects of process variables via various types of regression equations (logarithmic, polynomial & power) e.g.:

- **Adjustment-object label**: Total \( \% \)Zn in feed
- **Interaction factors**: \( I \)
- **Slope regression equation**: \( 0.058n(\% \text{Zn})+0.33 = \alpha \)
- **Final recovery regression equation**: \( l = \beta \)
- **Cost regression equation**: \( c \)

This regression rule (taken from data of Frew & Davey [2] would imply that for a feed grade of 11.5 \%, a would be unity (no deviation from pivot), hence corresponding to the pivot-rate which is 0.11 min\(^{-1} \) or the corresponding pivot \( k[C(0)] \) values (see below).

The Rules (production rules) defined here fall into various categories which include amongst others the following adjustment and search rules respectively:

- If \( \text{Sphalerite flotation} \) and \( \text{Type of ore boby} \) and \( \text{Zn content of feed} = 17.9 \% \) and \( \phi_{d} = 80 \text{ \mu m} \) and etc.

The user has control over the search accuracy of this search rule.

*Functions* include the kinetic models, differential equations that define the non-ideal flow through the continuous reactors or equations that describe fundamental aspects of the process. A function would hence contain the following elements:

- **Type of operation (defining label)**
  - Kinetic model (differential equation : see below)
  - Unit-operation-object (as above)
  - Non-ideal flow model (differential equation : see below)
  - Regression adjustment-objects (as adjustment-object above with factor replaced by a regression equation)

*Linear programming optimization model*

This knowledge representation is integrated in a user-friendly simulation programme which is written in Turbo Pascal. Figure 1 depicts the integration of these data.

---

**Fig. 1.** Overall structure of KBS.
DEVELOPMENT OF THE KNOWLEDGE BASED MODEL

As has been mentioned above, a generalized kinetic model has been developed which may be used to describe the kinetics of leaching, adsorption and pyrometallurgical reduction reactions. In this section it will be shown how this model can be applied in conjunction with a knowledge base to describe batch flotation kinetics and the flotation in continuous cells. Furthermore, it will be shown how optimal flotation circuits can be determined and fault-diagnosis can be performed by using this kinetic model.

Knowledge based kinetic model

As has been defined for leaching, adsorption and pyrometallurgical reactions [20-21], the rate of depletion of the valuable element in the slurry of a batch cell has been defined by a non-linear first order rate equation, in which the rate variable $k[C(t)]$ is only a function of $C(t)$, the concentration of element e ($e = galena, water, gangue, etc.$). No flotation species (e.g. fast and slow floating species within element e) are catered for in this model, as these are implicitly included in the rate variable $k[C(t)]$ and the accompanying knowledge base, which cover all possible particle size distributions, cell operating conditions and mineral compositions. The value of $k[C(t)]$ is determined as given below from a flotation curve, which covers the float up to the final recovery $R$.

$$\frac{dC_t}{dt} = - k_t[C(t)] [C(t)]$$

(1)

where: $k_t[C(t)] = - 2(C_{t,n} + 1-C_{t,n})/ δ_t[C_{t,n} + 1 + C_{t,n}]$.

The value of $k[C(t)]$ corresponds to the pivot process variables. Any deviation from these pivot-data is reflected by the $a$ and $b$ values of the adjustment objects. If these values deviate from 1, the following equation may be applied to predict the kinetics of the flotation process, were $k[C(t)]$ is as determined by equation 1 and the discrete ranges of $C(t)_n$ scaled linearly to fit between 1 and (1-8R) [R = final recovery for eq. 1]:

$$\frac{dC}{dt} = - \alpha \cdot k[C(t)] [C(t)] - (1 - b) = r.$$  

(2)

Continuous flow model

Ideal flow is assumed in the formulation of the continuous flow model for the depletion of an element e in cell i, although in the KBS non-ideal flow in the form of short-circuiting [C_{f,1} = φ(C_{f,1} + (1 - φ)C_{f,2})] and dead volume [$V_{act} = V_{react} - V_{lead}$] is incorporated. The theoretical values of the dead volume and bypass streams can be obtained from tracer tests, or can be estimated by comparing real plant behaviour with that predicted from an ideal flow model.

$$\frac{dC_i}{dt} = \frac{V_{p,in}}{V_{act}} C_i + \frac{V_{p,out}}{V_{act}} C_i + r.$$  

(3)

Linear programming optimization model

The constraints which define this model have been developed by Reuter and Vand Deventer [22] and the reader is hence referred to this reference for further details. Note, however, that the linear programming model that is defined here is a simplification of that defined in [22], as no flotation species are defined. Of all the constraints in the linear programming model that are of importance, the one that defines the separation of element e at bank j and links the KBS to the optimization routine, is repeated here:

$$\sum_{j} \lambda_{ji} P_{ji} \leq \eta_{ji} \leq \sum_{j} \lambda_{ji} P_{ji}$$

(4)

Values for ($\eta_{ji}$1- and ($\eta_{ji}$)10 are determined as follows: (i) the expected operating conditions in a bank of cells are selected, which produce an adjusted kinetic model (eq.2), (ii) this can subsequently be used to simulate a bank of continuously operated flotation cells (eq.3) and finally from this a separation factor for each element e can be determined.

Process identification and Fault-Diagnosis

The definition of the rate variable $k[C(t)]$ in the manner as described by eq.1 is very useful to perform fault diagnosis using kinetic data only. Consider a first order reaction and the following time-weighted slope integral:

$$\int_{0}^{\infty} \frac{1}{k} C(t) dt = \int_{0}^{\infty} \frac{1}{k} C(t) dt = 1/k.$$  

(5)

If this value for $1/k$ is compared to that of another curve with rate constant $k'$, the ratio $k/k'$ gives a value for the adjustment factor $a$, which can subsequently be used in conjunction with $b$ to search through all adjustment-rules to determine what the chemical environment in a batch flotation cell is. This has been generalized for the rate variable $k[C(t)]$ in the following manner:

$$1/k = \sum_{j} \lambda_{ji} k[C(t)] C(t) \eta_{ji} dt.$$  

(6)

For the process identification in a continuous bank of cells, the $a$ and $b$ values are adjusted continuously, until the recovery corresponds to that obtained from the plant. The $a$ and $b$ values are an indication of the deviation from the data in the pivot-object which subsequently points to the appropriate adjustment-objects, which then suggest the appropriate operating conditions.

SIMULATION RESULTS

Various examples will be given in the subsequent section to illustrate the application of the developed models.

Batch flotation

The generality of the proposed kinetic model (eqs.1 & 2) is illustrated by comparing its performance to that of two well known kinetic models. A number of values for $k[C(t)]$ for the concentration-time data in tables 2 and 3 are summarized in table 1. These values are subsequently used to produce the results in the (b) columns of tables 2 and 3.

It is clear from the results summarized in tables 2 and 3 that one model can predict the same results as two different well known theoretical models. It may be mentioned here that the same model has also been applied to adsorption, leaching and pyrometallurgical reactions, with results comparable to flotation above [20-21].
TABLE 1
Summary of the calculated k[C(C(t))<sup>0</sup>] values according to equation 1, using the data in column (a) in Table 1

<table>
<thead>
<tr>
<th>Column 1(a) of Table 2</th>
<th>Column 1(a) of Table 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>C&lt;sup&gt;0&lt;/sup&gt; Range</td>
<td>k[C(C(t))]&lt;sup&gt;0&lt;/sup&gt;</td>
</tr>
<tr>
<td>1.00 to 0.95</td>
<td>-0.496</td>
</tr>
<tr>
<td>0.95 to 0.90</td>
<td>-0.204</td>
</tr>
<tr>
<td>0.90 to 0.86</td>
<td>-0.306</td>
</tr>
<tr>
<td>etc.</td>
<td>etc.</td>
</tr>
</tbody>
</table>
| ... | 0.00 | 0.00 | ... | 0.00 | 0.00 | ...

TABLE 2
Columns (a): Concentration-time data from the given model and R and k values. Data in column 1(a) were used to determine k[C(C(t))]<sup>0</sup> by eq. 1, which was subsequently used to predict values in columns (b) according to eq. 2

<table>
<thead>
<tr>
<th>(a): C/Co = 1-R[1-(1-e&lt;sup&gt;-kt&lt;/sup&gt;)/kt]</th>
<th>(b): Eq. 1 &amp; 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>R &amp; K</td>
<td>1.0 &amp; 1.0/min</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>1 (min)</td>
<td>1 (a)</td>
</tr>
<tr>
<td>0.0</td>
<td>1.000</td>
</tr>
<tr>
<td>0.5</td>
<td>0.787</td>
</tr>
<tr>
<td>1.0</td>
<td>0.245</td>
</tr>
<tr>
<td>6.0</td>
<td>0.016</td>
</tr>
</tbody>
</table>

Continuous flotation
To illustrate the application of this proposed KBS to a practical system, the sphalerite flotation data of Frew and Davey [2] were used. For this purpose, the various data objects are defined as those given above. Although not shown here, objects which define the effect of reagent additions, feed grades and particle sizes on the flotation rate constant, must also be included. Frew and Davey [2] described flotation by simple first order kinetics. From their data it is clear that the rate constant k[C(C(t))]<sup>0</sup> has a constant value of 0.11 min<sup>-1</sup> for the pivot-object defined above for cells 1-4. Assuming ideal flow, using the appropriate unit-operation-object, applying eqs. 1-3 and the given retention times, the results summarized for cells 1-4 in table 4 could be produced for Experiment 1 [2].

From the simulation results for cells 1-4 and the process conditions within cells 5-7, appropriate adjustments-objects would suggest that the pivot-rate k[C(C(t))]<sup>0</sup> should be adjusted by a factor of 2. Subsequently, these cells as well as cells 8-10 could be simulated by following the same method. From the above it is clear that the comparison is excellent.

A similar approach must be followed for all other components involved, e.g. in the above example galena, pyrrhotite, non-sulphide gangue and water. For each of these an associated adjustment-object must be defined. For example, for the gangue under the prevailing conditions, the rate of water recovery is a factor α = 0.035 of that of the sphalerite in cells 1-4, 0.027 in cells 5-10 and for the gangue α = 0.012 for all cells.

Optimization of a flotation circuit
Once the various simulations have been performed for various flotation scenarios and recycle possibilities in order to determine the appropriate separation factors, a linear programming model for the valuable element, in this case sphalerite, could be developed according to [22]. From table 1 the separation factor for the sphalerite in the rougher and scavenger banks respectively could be determined to produce the following two constraints:

0 m<sub>VBS</sub> ≤ y<sub>VBS</sub> ≤ 11.98 m<sub>VBS</sub> 1;
0 m<sub>VBS</sub> ≤ y<sub>VBS</sub> ≤ 3.8 m<sub>VBS</sub> 2.

The constraint for the cleaners, which were grouped here into one bank, could be estimated from the given data [2] to be (kinetic data not given for these banks):

0 m<sub>VBS</sub> ≤ y<sub>VBS</sub> ≤ 90 m<sub>VBS</sub> 3.

The result of this simulation is the depicted circuit structure (fig. 2) for a maximization of the sphalerite recovery. The recovery for this structure is 98.4 %. However, if the concentrate recycle of the scavenger is forced to report to the rougher, i.e. the structure as given by [2], the recovery drops to 98.2 %. With a view to determine the grade, a similar model can be developed for the other elements in order to minimize their flow for the given optimal structure for the sphalerite (see [22]).

TABLE 3
Columns (a): Concentration-time data from the given model and R and k values. Data in column 1(a) were used to determine k[C(C(t))]<sup>0</sup> by eq. 1, which was subsequently used to predict values in columns (b) according to eq. 2

<table>
<thead>
<tr>
<th>(a): C/Co = 1-R[1-(1-e&lt;sup&gt;-kt&lt;/sup&gt;)/kt]</th>
<th>(b): Eq. 1 &amp; 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>R &amp; K</td>
<td>1.0 &amp; 1.0/min</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>1 (min)</td>
<td>1 (a)</td>
</tr>
<tr>
<td>0.0</td>
<td>1.000</td>
</tr>
<tr>
<td>0.5</td>
<td>0.607</td>
</tr>
<tr>
<td>1.0</td>
<td>0.368</td>
</tr>
<tr>
<td>4.0</td>
<td>0.018</td>
</tr>
</tbody>
</table>

TABLE 4
Comparison of the practical data for Experiment 1 (rougher & scavenger) of Frew and Davey [2] with those produced by the proposed KBS

<table>
<thead>
<tr>
<th>Cell N°</th>
<th>Plant ZnS</th>
<th>KBS ZnS</th>
<th>Plant Gangue</th>
<th>KBS Gangue</th>
<th>Plant Water</th>
<th>KBS Water</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.75</td>
<td>0.76</td>
<td>0.995</td>
<td>0.995</td>
<td>0.96</td>
<td>0.98</td>
</tr>
<tr>
<td>2</td>
<td>0.57</td>
<td>0.58</td>
<td>0.99</td>
<td>0.99</td>
<td>0.94</td>
<td>0.96</td>
</tr>
<tr>
<td>4</td>
<td>0.36</td>
<td>0.34</td>
<td>0.985</td>
<td>0.985</td>
<td>0.92</td>
<td>0.92</td>
</tr>
<tr>
<td>7</td>
<td>0.076</td>
<td>0.077</td>
<td>0.98</td>
<td>0.98</td>
<td>0.88</td>
<td>0.88</td>
</tr>
<tr>
<td>10</td>
<td>0.016</td>
<td>0.016</td>
<td>0.97</td>
<td>0.95</td>
<td>0.85</td>
<td>0.84</td>
</tr>
</tbody>
</table>
From the above flow rates it is clear that they do not differ much from those given in [2] for Experiment 1, despite the difference in structure.

Fault-Diagnosis

In order to illustrate the identification of a process on the basis of its batch flotation curve, eq. 5 is applied to the pivot-data of the flotation example [2] for which k = 0.11 min⁻¹. The integral will produce a value of 9.09. If another curve is compared to this, say with a rate constant k = 0.16 min⁻¹, the integral produces a value of 6.25, hence α = 9.09 / 6.25 = 1.45. The search routine will attempt to produce a match for this α and suggest appropriate operating conditions.

If the search rule for cells 1-4 is considered, it is clear that the result for this search would be the pivot-conditions as given. However, if the search ranges were to change to those given below, the result would have pointed to an increased Zn feed grade of 17.9 %:

and Zn recovery is ≥ 72 %;
and Zn recovery is ≤ 73 %;
and Cells 1-4 or rougher bank;
then Zn content of feed = 17.9 %;

From the simulation of Experiments 2 and 3 for cells 1-4 [2], it would have become clear via the Total %Zn in feed-object that the total Zn feed (fresh & recycle) to the rougher has to increase so as to produce the measured recovery.

DISCUSSION OF RESULTS

The simulation of batch and continuous flotation by use of an object-oriented KBS, was discussed above. Furthermore, it was shown how the data of these simulations could be used to optimize a flotation plant and perform fault diagnosis.

The results of these simulations serve to illustrate the following regarding this simulation approach:

- It was shown how a generalized non-linear first order rate equation can be used to approximate two different well-known theoretical flotation models. Although not shown here, it has been demonstrated that this same equation could be applied to accurately simulate leaching, adsorption and pyrometallurgical reduction reactions.
- This form of the rate equation permits the definition of a generalized knowledge base which can interact with the rate variable and the final recovery via objects. This permits the definition of a well-defined kinetic model for any type of reaction under consideration, be it leaching, adsorption, reduction or flotation.
- The application of this kinetic model was used to simulate the recovery of sphalerite, gangue and water in continuous rougher and scavenger banks by taking the process conditions within a cell into consideration. The results in this case were very good.

The fault-diagnosis/process identification can also be performed by utilizing the adjustment information a and b of the generalized kinetic model. By comparing this information to the pivot-data, the model can suggest the possible conditions within the reactor under consideration.

- It was demonstrated how the simulation and process identification activities could be integrated with a linear programming optimization model. A circuit was produced which gave nearly the same results as the industrial flotation plant, with the difference being that the concentrate of the scavenger reports to be cleaner and not to the rougher. If this stream is forced to report to the rougher, the recovery drops by 0.2 %.

From the results of the KBS, it is clear that the KBS could render an off-line process well-defined, and it is able to perform simulations that standard simulation programs could not perform.

SYMBOLS USED

- C, C', Cₙ Concentration in a batch, concentration in stage i and data point n of element e respectively.
- k[C(t)] Rate variable as a function of C(t) / min⁻¹.
- mₑγ, Yₙ Mass flow rates of tailings and concentrate of element e at bank j respectively / t h⁻¹.
- r Rate of flotation / min⁻¹.
- R Final recovery.
- t Time variable / min.
- Vₙᵢ Active stage volume / m³.
- Vₑᵢᵣᵢ Pulp flow rate in and vₑᵢᵣᵢ out of a stage /m³ min⁻¹.
- [(ymᵢᵢ)ᵤᵢᵢ,uᵢᵢ] Lower bound on the separation factor of element e at bank j. [(ymᵢᵢ)ᵤᵢᵢ,uᵢᵢ] upper bound.
- δᵢ Time increment / min.
- Φ Fraction of pulp short-circuiting reactor stage.
- α, β Adjustment factors/equations.
- Sh, Re, Sc Sherwood, Reynolds and Schmidt numbers.

REFERENCES


A KNOWLEDGE BASED SYSTEM FOR THE SIMULATION OF BATCH AND CONTINUOUS CARBON-IN-PULP SYSTEMS

M.A. Reuter Hons.-B.Eng., M.Eng., M.S.A.I.M.M.

Department of Metallurgical Engineering, University of Stellenbosch, Stellenbosch, 7600, South Africa.

SYNOPSIS

A knowledge based kinetic model for the adsorption of gold and silver cyanides on activated carbon is proposed. The kinetic model requires only one set of experimental adsorption data to produce predictions of adsorption behaviour in batch, counter current cascade and column reactors respectively. Whereas most kinetic models require a prediction method to estimate the parameters in the model, the proposed model uses simple algebra to estimate its single parameter directly from the experimental data selected from a database. It is clear that if a set of adsorption data is available at operating conditions which are different from those of existing adsorption kinetic models, e.g. different levels of oxygen, carbon blinding, carbon poisoning etc., the model will still be able to produce predictions. This model is subsequently used to predict the performance of packed and moving columns, and counter current cascade reactors.

Due to the simplicity of the model, its minimal memory and data requirements, and the fact that it requires batch data in a concentration-time format, it is suitable for application in a knowledge based system. Hence, a section of this paper is devoted to the discussion of a knowledge based system that is currently being developed. It is clear that all simulations are very much dependent on the correct choice of batch data. To ensure that the correct data are chosen, heuristic knowledge regarding the process is also included e.g. rules of thumb, operator knowledge etc., which controls the choice being made on the basis of operating parameters.

The proposed knowledge based model is tested for a wide variety of experimental operating conditions. A good correspondence between the experimental data and the predictions is obtained.
INTRODUCTION

The kinetic modelling of the adsorption of gold cyanide and silver cyanide on activated carbon has received considerable attention in the past number of years. This is reflected by the large number of kinetic adsorption models that are discussed in the literature. Several models such as semi-empirical\(^1\), adsorption-desorption\(^2\) and branched-pore\(^3,4\) models have been employed in the simulation of unit operations. Menne\(^5\) used a kinetic adsorption-desorption model to predict the gold adsorption kinetics of industrial pulps, while Nicol et al.\(^6\) used a semi-empirical model to predict the behaviour of multistage adsorption circuits. Jansen van Rensburg and Van Deventer\(^7,8\) formulated a branched-pore model to predict the performance of gold and silver adsorption on activated carbon in packed-bed and moving-bed columns while Van Deventer\(^9\) used the same model to predict the performance of counter current adsorption cascades.

Most of the above-mentioned models, however, have a relatively large number of parameters that have to be estimated in order to fit the data. These parameters, which could include film transfer coefficients, intraparticle diffusion coefficients and equilibrium constants, sometimes yield good predictions, but are numerically tedious to estimate. Furthermore, the need for an isotherm or pseudo-isotherm in these predictions involves a considerable amount of experimental work.

Consequently, a single-parameter kinetic model for the adsorption of gold and silver cyanides on activated carbon is proposed here. Contrary to the published models, this single parameter is not estimated, but calculated directly from batch concentration-time data by simple algebra. Since concentration-time data in a data bank form the basis from which all predictions originate, the model is termed a knowledge based kinetic model.

This model is exceptionally suited for application in a knowledge based system, since the lack of parameters imposes very few additional demands on the concentration-time database. Furthermore, the simplicity of the model minimises the computational effort, so that more time can be devoted to the search for concentration-time data in the database. The inclusion of heuristic knowledge of the process in the database to assist in selecting the correct concentration-time data, is another attractive feature of this approach.

The next section will be devoted to the development of the knowledge based kinetic model, and subsequently its application in the simulation of batch reactors, counter current cascades and packed carbon columns will be discussed. Finally, the application of this kinetic model in a knowledge based system, which is currently being developed, will be discussed.
KNOWLEDGE BASED KINETIC MODEL

Usually the rate of a reaction is defined by\(^{10}\):

\[
\text{d}X_r = - \frac{\text{d}X}{\text{d}t}
\]

where

\[X = \text{Dimensionless concentration of adsorbed component in solution}\]

It is assumed here that the rate of adsorption can be represented by a first order rate equation:

\[
\text{d}X_r = K(\text{load}) X
\]

where

\[K(\text{load}) = \text{a function of the loading load on the carbon. This implies that } K(\text{load}) \text{ changes as the loading changes.}\]

The results of Van Deventer\(^3,4,11\) show that the diffusion into the macropores could control the rate of adsorption at higher loadings (liquid diffusion coefficient is on average 1000 times as large as the macropore surface diffusion coefficient). Since \(k_f\) is usually assumed to remain constant for a particular set of operating conditions, one can assume that the rate \(K(\text{load})\) is only dependent on the loading load of the adsorbed component (Au or Ag cyanide) on the carbon. If however \(K(\text{load})\) becomes a function of \(k_f\), a new batch curve must be selected from the data bank to calculate the corresponding \(K(\text{load})\) values. The estimation of \(K(\text{load})\) will be discussed subsequently.

Using equation 2 it may be seen readily that the value for \(K(\text{load}_i)\) at a particular stage \(i\) of a batch adsorption process, may be given by:

\[
K(\text{load}_i) = - \frac{\text{d}X_i}{\text{d}t} X_i
\]

By using finite differences, an expression for \(K(\text{load}_i)\) may be formulated from \(N\) dimensionless concentration-time data points. This expression is subsequently used to define an expression for the rate \(r\):

\[
r = \sum_{i=1}^{N} a_i K(\text{load}_i) X = - \sum_{i=1}^{N} a_i \frac{X_i - X_{i-1}}{X_{av} \delta t} X
\]
where
\[ a_i = \begin{cases} 1 & \text{if } \text{load}_{i-1} \leq \text{load} \leq \text{load}_i \text{, then } a_i \text{ becomes } 1, \text{ else } a_j = 0 \text{ for all } j \neq i. \end{cases} \] For each load value equation 4 reduces to \[ r - K(\text{load})X, \] a simple first order equation.

\[ X_{av} = \frac{X_i + X_{i-1}}{2} \]

Note that equilibrium is catered for implicitly in this model, since after a certain loading load the slope, i.e. \( K(\text{load}) \), becomes zero.

In the next section it will be demonstrated how this model may be used to predict the performance of batch stirred tank reactors, counter current adsorption cascades and packed bed adsorption columns.

**Simulation of a batch stirred tank reactor**

By substituting equation 4 into 1, the differential equation that defines the change of dimensionless concentration in a batch reactor is given by:

\[ r = \frac{dX}{dt} = \sum_{i=1}^{N} a_i K(\text{load}_i) X \]

Since the loading on the carbon in the simulated batch reactor must be referred back to the original batch data, the load is calculated from:

\[ \frac{d \text{load}}{dt} = r \]

When equations 5 & 6 are used to predict the performance of a batch stirred tank reactor, the following procedure is used:

- The database of the kinetic model is searched for a set of batch adsorption concentration-time data, which corresponds to the type of activated carbon being used and the operating conditions present in the reactor.

- From the dimensionless batch concentration-time data points, values for \( K(\text{load}_i) \) may be determined using the defining equation 4, after having smoothed the data with a cubic spline. These data are stored in an array which has the format \( \{\text{load}_{i-1}, \text{load}_i, K(\text{load}_i)\} \).

- At time zero the initial loading load on the carbon is equal to 0 and hence the corresponding value for \( K(\text{load}_i) \) is \( K(\text{load}_1) \).
By means of an appropriate numerical method (Runge-Kutta or Euler), the resulting differential equations 5 & 6 may be solved to produce the next value for \( X \) in the batch reactor and the next carbon loading \( \text{load} \).

The \( K(\text{load}_i) \) that corresponds to this new loading \( \text{load} \) is chosen and the previous step repeated for as long as is required.

The concentration-time data selected from the data-base corresponds obviously to particular operating conditions e.g. mass of carbon (M), stirring speed, carbon mesh size (\( d_p \)), volume of reactor (V) etc. prevalent during their determination in a batch adsorption experiment. When equations 5 & 6 are used to predict the adsorption kinetics at operating conditions other than those represented by the chosen data, the rate \( r \) in equations 5 & 6 is modified by substituting \( r \) by \( r_1 \):

\[
{r}_1 = \left[ {r}/ {r}_1 \right] \cdot {r}
\]

where

\[
{r} = \rho_c V d_p / 6 k_f M
\]

\[
{r}_1 = \left[ \rho_c V d_p / 6 k_f M \right]_1
\]

This equation follows directly from the liquid phase rate equation at a particular \( k_f \) as proposed by Van Deventer. If, however, \( k_f \) varies markedly between these different sets of operating conditions, a new set of concentration-time data must be obtained from the databank which corresponds to the input conditions, i.e. M, V, \( d_p \), stirring speed (agitation) and \( \rho_c \).

If \( r \) is equal to \( r_1 \) (i.e. \( r= r_1 \)), but the initial concentration in the simulated batch reactor is \( C_{01} \), while the chosen data correspond to an initial concentration of \( C_0 \), the differential equation that defines the change in concentration in the simulated batch reactor may be written as:

\[
\frac{dX}{dt} = \left[ C_0/C_{01} \right] \cdot {r}_1
\]

The simulation of counter current adsorption cascades

The mass balance equation over the \( j \)-th continuous stirred tank reactor in a cascade, where each reactor contains \( W \) loading fractions, is given by:

\[
\frac{dX_j}{dt} = \left[ X_{j-1} - X_j \right]/ \tau_j - \sum_{k=1}^{W} w_{kj} \cdot {r}_{kj}
\]
The calculation of \( w_{f_j} \), the discrete loading fraction, is discussed elsewhere. The loading in each loading fraction \( f \) in the \( j \)-th stirred tank reactor is given by:

\[
\frac{d \text{ load}_{f_j}}{dt} = r_{f_j} \tag{10}
\]

The solution of these equations, i.e. equation 9 for each reactor and equation 10 for each loading fraction \( f \) in reactor \( j \), is conducted in the same way as for the batch reactor.

The simulation of packed carbon adsorption columns

The simulation of activated carbon adsorption columns is well established. Various models exist that define the change of concentration in adsorption columns, two of which are plug-flow models and tanks-in-series. While the first model assumes that no axial dispersion takes place the second model accounts for backmixing.

Since the diameters of the experimental columns are considerably larger than the mean size of the activated carbon particles, it is assumed here that no axial dispersion takes place. Hence the plug flow model is used here to define the change in concentration in the packed bed column. If the equation developed by Van Deventer and Van Rensburg is used as a basis, it can be shown that adsorption in a column can be described by the following dimensionless equation:

\[
\frac{\partial X}{\partial \Phi} + \frac{\partial X}{\partial \Theta} + \frac{1}{v} r_1 = 0 \tag{11}
\]

where

\[
\Phi = \frac{z}{L} \\
\Theta = \frac{tv}{L} \\
r_1 = \text{Kinetics for a batch reactor as defined by equation 5 to 7.}
\]

If each section \( \Phi \) of the column is considered to be a stirred tank reactor and the absolute starting concentration is 0 ppm, the loading on the carbon at locations \( \Phi \) in the column may be calculated from:

\[
\frac{d \text{ load}_{\Phi}}{d \Theta} = [(C_{av}/C_0)^{\frac{1}{v}}] r_1 \tag{12}
\]
where

\[ C_{av} = \] Is the average starting concentration (ppm) in the column at all locations \( \Phi \), throughout the adsorption. This concentration may be estimated to give the best fit for breakthrough curves for a variety of operating (i.e. different \( \nu \), \( L \) and \( r \)) conditions for a particular column.

\[ C_o = \] The inlet concentration (ppm).

The solution procedure is similar to that given for the batch reactor, the only difference being that the partial differential equation is solved by means of an implicit method"}.12

KNOWLEDGE BASED SYSTEM

Although the knowledge based system being developed at the University of Stellenbosch has not been completed entirely, the basic elements thereof will be discussed and illustrated by means of figures from the system in this section.

The basic elements of the knowledge based system may be divided into two distinct sections, i.e. the Expert System and the Mathematical Modelling System respectively. The expert system is programmed entirely in Prolog and makes extensive use of the artificial intelligence (AI) capabilities of this AI-language. The principal activities of this system are:

- The management of a database consisting of concentration-time data, heuristic knowledge of the process, etc.

- The inference section, which selects the correct data from the database, subject to the input conditions. The rules that form part of this section, control the quality of the data produced.

This expert system produces a file from the chosen concentration-time data which represents the input to the next section of the system, i.e. the Mathematical Modelling Section, which is programmed in Pascal. The basis of this system is the knowledge based kinetic model, which was derived in the previous sections. The principal activities of this system are:

- The manipulation of the data produced by the expert system, i.e. the calculation of \( K(\text{load}) \), using a cubic spline.

- The simulation of reactor systems, that use the data produced by the expert system to simulate the selected carbon-in-pulp unit operation.
Figure 1 shows the pop-up main menu of this Mathematical Modelling System and the self explanatory menu-options that may be selected. This figure depicts the View/Edit Data option being exercised, i.e. the data produced by the expert system may be viewed, edited and saved if so required. The data points that are being edited in this figure are the data used to simulate the counter current cascade, which will be discussed later on. Figure 8, a graphics output from this system, also depicts these data. Note that the Import Data menu option permits the importation of data from databases other than the Expert System, e.g. Lotus 1-2-3, dBASEIII or user data-files.

The input data for the simulation of a counter current cascade are depicted by figure 2. These input conditions correspond to those used in the simulation of the counter current cascade, which is discussed later in this paper.

---

Fig. 1 The main pop-up menu for the mathematical modelling system.
Fig. 2 The input conditions for the simulation of a five reactor counter current cascade system.

SOURCES OF DATA

All the experimental data for the adsorption of gold cyanide and silver cyanide on activated carbon in batch reactors, counter current cascades and packed column adsorption columns were taken from the papers and theses by Van Deventer and Jansen van Rensburg 3,4,7-9,11,16, which may also be consulted for the experimental methods.

RESULTS AND DISCUSSION OF MODEL PREDICTIONS

The model predictions will be discussed under three headings viz. batch stirred tank reactor, counter current cascades and, packed and moving bed column reactors respectively. Figures depict the experimental results and the subsequent model predictions.

Batch stirred tank reactor

The simulation results for the batch stirred tank reactor all reflect an excellent correspondence with the experimental data.

It must be noted here that the data that were used in simulations 1 to 4 reflect an independence of $k_f$. This may, however, not be the case when a multi-component system is being simulated. As pointed out earlier, a new set of batch data must be selected by the expert system from the data bank, which defines the adsorption under these multi component adsorption conditions. (See figure 8 later).
A time increment of $\Delta t=0.5$ hour was applied during the solution of equation 5 by Euler's method. $K(load_i)$ was computed by taking $\Delta t=1$ hour.

Simulation 1

Figure 3 depicts experimentally determined data for the adsorption of gold cyanide on Le Carbone charcoal (sample A) with the following experimental conditions: stirring speed=150rpm, $V=21$, $M=2g$ and $1.4 \leq d_p \leq 1.7mm$. The concentration-time data for $C_0=25ppm$ were selected to determine the values for $K(load_i)$. The subsequent predictions by the knowledge based kinetic model are given as solid lines in figure 3.

![Figure 3](https://scholar.sun.ac.za)

**Fig. 3** The effect of initial concentration on the adsorption kinetics of gold cyanide in a batch stirred tank reactor.

Simulation 2

The effect of different masses of carbon is illustrated by figure 4. The experimental conditions for the adsorption of silver cyanide on Le Carbone charcoal (sample B) were: stirring speed=100rpm, $V=51$, $C_0=72ppm$ and $1.4 \leq d_p \leq 1.7mm$. The adsorption curve for $M=5g$ was selected to determine the values for $K(load_i)$. Predictions for the other masses of carbon were based on this choice.
Fig. 4 The effect of different masses of carbon on the adsorption kinetics of silver cyanide in a batch stirred tank reactor

Simulation 3

Figure 5 depicts adsorption data for a variety of operating conditions other than those used by simulation 2, but using the same Le Carbone (sample B) charcoal. The operating conditions for curves A, B and C respectively are:

A: V=2l; M=0.57g; 1.4≤d_p≤1.7; stirring speed=180rpm and C_o=25ppm.

B: V=2l; M=2g; 1.7≤d_p≤2.0; stirring speed=240rpm and C_o=38ppm.

C: V=2l; M=2g; 1.7≤d_p≤2.0; stirring speed=240rpm and C_o=20ppm.

In order to predict these profiles, the same K(load_i) values as those used for simulation 2 were applied. From the predictions it is clear that the stirring speed does not have a significant effect. This is in accordance with Van Deventer\(^4\), who stated that above 150rpm the effect of stirring speed is negligible at high initial concentrations. The slight inaccuracy that may exist in the predictions may be attributed to the slight difference in particle size between the data of simulation 2 and those of simulation 3.
This simulation demonstrates that the proposed model can predict adsorption data over a wide range of operating conditions, given one set of adsorption data for a particular carbon.

![Graph](image_url)

**Fig. 5** The effect of a wide range of operating conditions on the adsorption kinetics of silver cyanide in a batch stirred tank reactor.

**Simulation 4**

This simulation demonstrates the ease with which competitive adsorption of gold and silver cyanide can be described by the model, hence eliminating the use of complex multi-component equilibrium isotherms. Figure 6 depicts the experimental data which were used to determine the values for \( K(\text{load}_i) \) for both silver and gold cyanide respectively. The experimental conditions were: stirring speed=240rpm, \( M=2.5g \), \( C_0(\text{Au})=34ppm \), \( C_0(\text{Ag})=28ppm \), \( V=2l \) and \( 1.7 \leq d_p \leq 2mm \).
The simultaneous adsorption of gold cyanide and silver cyanide in a batch stirred tank reactor with \( M = 2.5g \).

Figure 7 depicts the resulting prediction using these data in a batch reactor with experimental conditions: stirring speed = 240rpm, \( M = 4g \), \( C_0(Au) = 34ppm \), \( C_0(Ag) = 28ppm \), \( V = 2l \) and \( 1.7 \leq d_p \leq 2mm \).
The simultaneous adsorption of gold cyanide and silver cyanide in a batch stirred tank reactor with $M=4g$.

**Counter current cascade**

The data used for this simulation are given by the dots in figure 8, the graphics output from the *Mathematical Modelling System*. As pointed out earlier, some of these data are also given in figure 1. These data were taken from adsorption experiments conducted using an industrial leach slurry, containing a wide range of components (*The slurry had a specific density of 1.4, pH=8.7, mass fraction of solids=0.49, activated carbon:Le Carbone G210 AS, Components:8.3ppm Au, 0.6ppm Ag, 24.1ppm Cu, 8.1ppm Ni, 3.7ppm Fe and 363.8ppm Ca, 2.8ppm Mg and 28.9ppm free CN\(^-\)*). The solid line in figure 8 represents the prediction of the knowledge based model in a batch reactor. It is clear that the correspondence between experimental and predicted values is excellent.
The simulation of the performance of a batch adsorption reactor using an industrial slurry. By use of the slopes calculated from these experimental data, a five bank counter current cascade could be simulated. Figure 9 represents the loading on the carbon over seven cycles in the first of five reactors, the loading profile over the five reactors after seven cycles, and the concentration profile over the five reactors after seven cycles. These estimations are all produced by the Mathematical Modelling System (see figure 2 for the input). The agreement in all cases is reasonable, indicating that the approach can be used to simulate counter current cascades.
The simulation of the performance of a batch adsorption reactor using an industrial slurry.

By use of the slopes calculated from these experimental data, a five bank counter current cascade could be simulated. Figure 9 represents the loading on the carbon over seven cycles in the first of five reactors, the loading profile over the five reactors after seven cycles, and the concentration profile over the five reactors after seven cycles. These estimations are all produced by the Mathematical Modelling System (see figure 2 for the input). The agreement in all cases is reasonable, indicating that the approach can be used to simulate counter current cascades.
The adsorption in a continuous counter current cascade with $Q=4.44$ l/h, $M=40g$, $\beta=0.4$ and $T=16h$. 
Packed and moving bed column reactors

As in the case of the batch reactor, excellent agreement was obtained between simulations and experimental data. A voidage fraction of \( \varepsilon = 0.42 \) and a density of 476 kg m\(^{-3}\) for the activated carbon were used throughout the simulation of the columns.

A \( L/d_L \) ratio of 50 was used for all computations involving columns and \( \Delta t \) was taken as 0.5 s. A value of 1 hour was taken for \( \Delta t \) during the computation of \( K(\text{load}_1) \).

Simulation 1

This simulation illustrates the effect of different flow rates through the packed bed and the effect of packed bed height on the performance of the adsorption column. The column has an inner diameter of 31 mm. Figure 10 depicts the experimental silver cyanide adsorption batch data that were used to determine the values for \( K(\text{load}_1) \), which in turn were used to predict the breakthrough curves given in figure 11. The experimental conditions for the batch experiment are given in figure 10, while these conditions for the column were: \( C_0 = 4.5 \text{ ppm} \), \( 0 \leq d_p \leq 2.36 \text{ mm} \) and \( \text{pH} = 8.5 \). For this simulation the value for \( (C_{av}/C_0) \) is taken as 0.1, to predict all three curves.

Fig. 10 The adsorption of silver cyanide in a batch stirred tank reactor.
Fig. 11 The prediction of silver cyanide breakthrough curves in a packed bed reactor with various bed heights and an inner diameter of 31mm.

Simulation 2

A similar simulation was conducted with a gold cyanide solution, using a packed column with a height of 50cm and inner diameter of 31mm. The batch data which were used to determine $K(\text{load}_i)$, as well as the relevant experimental conditions, are given in figure 12. The experimental conditions for the column are identical to that for the batch reactor. The value for $(C_{av}/C_0)$ is taken as 0.03 (Co=5ppm) during the course of the simulation. Figure 13 reflects a close correspondence between the simulated and experimentally determined results.
Fig. 12 The adsorption of gold cyanide in a batch stirred tank reactor.
The prediction of gold cyanide breakthrough curves in a packed bed reactor with a bed height of 50cm and an inner diameter of 31mm.

Simulation 3

The final simulation illustrates an application to a moving bed adsorption column, which has an inner diameter of 25mm and a bed height of 34cm. The batch data that were used to determine $K(loa_{d})$, were obtained from Van Deventer and Jansen van Rensburg. These data represent the adsorption kinetics for a clarified industrial leach solution (7.43ppm Au).

Figure 14 depicts the breakthrough curves for the cyanide leach solution (7.43ppm Au) fed at 61/h to the periodic countercurrent moving bed. The column was operated in cycles of 24 hours, i.e. after 24 hours the bottom 40% of the column was removed from the column, while fresh carbon was added to the top. The value for $(C_{av}/C_{o})$ is equal to 0.77 for these experimental conditions.
CONCLUSIONS

It is clear from the results that the knowledge based approach that is represented here, produces excellent predictions for a wide variety of adsorption experiments in batch stirred tank reactors, packed bed column reactors as well as moving bed column reactors.

The inclusion of this model in a knowledge based system is relatively easy. Due to the simplicity of the model and the low demands that it imposes on the database, more time may be devoted to the search, up-dating and management of the database. It is intended to make the programmes described in this paper commercially available in due course.

Although the effect of oxygen and other operating conditions e.g. carbon blinding and poisoning were not discussed here, it is clear that, given a set of adsorption data that includes these effects, that predictions for batch and column runs could be made. Furthermore, the ease with which experimental data are manipulated, makes the knowledge based kinetic model easier to apply and more practical than previously published models, which usually require the estimation of a large number of parameters.
LIST OF SYMBOLS

\(a_i\) 0 or 1 depending on the loading of the carbon

\(C_0\) Initial concentration in a batch reactor or inlet concentration for a column reactor (ppm)

\(C_{av}\) Average starting concentration in column (ppm)

\(dL\) Incremental bed height used in computations (m)

\(dt\) Incremental time used in computations (s)

\(d_p\) Average particle size of activated carbon (m)

\(\text{load}\) Dimensionless loading on carbon

\(\text{load}_{lj}\) Dimensionless loading on carbon in loading fraction \(l\) in reactor \(j\)

\(k_f\) Film diffusion coefficient (m.s\(^{-1}\))

\(K(\text{load}_i)\) \(-\frac{[X_i - X_{i-1}]}{X_{av} \delta t}\) (s\(^{-1}\))

\(L\) Adsorption column length (m)

\(M\) Mass of carbon (kg)

\(r\) Rate of reaction (s\(^{-1}\))

\(r_{lj}\) Rate of reaction (s\(^{-1}\)) for loading fraction \(l\) in reactor \(j\)

\(r_l\) Rate of reference adsorption reaction (s\(^{-1}\))

\(t\) Time (s)

\(T\) Period for one adsorption cycle (hour)

\(V\) Volume of reactor (m\(^3\))

\(W\) Number of loading fractions in a reactor \(j\)

\(w_{lj}\) Loading fraction \(l\) in reactor \(j\)

\(X\) Dimensionless concentration

\(X_j\) Dimensionless concentration in reactor \(j\)

\(X_{av}\) \((X_i + X_{i-1})/2\)

\(z\) Axial distance variable in column
Greek

\( \beta \) Fraction of carbon and pulp moved to previous reactor in a cascade.

\( \delta t \) Time increment for the calculation of \( K(\text{load}) \) (s)

\( \rho_c \) Apparent density of activated carbon (kg.m\(^{-3}\))

\( \tau = \rho_c V_d p / 6 k_f M \)

\( \tau_j \) \( \tau \) for reactor \( j \)

\( \nu \) Liquid velocity (m.s\(^{-1}\))

\( \phi \) Dimensionless axial distance in column, \( z/L \)

\( \theta \) Dimensionless time, \( t \nu/L \)

References


MINERALS, MATERIALS AND INDUSTRY

Papers presented at the Fourteenth Congress of the Council of Mining and Metallurgical Institutions, organized by the Institution of Mining and Metallurgy and held in Edinburgh, Scotland, from 2 to 6 July, 1990

The Institution of Mining and Metallurgy
Knowledge-based computer simulation of
gold leaching in batch and continuous
systems

I. W. van der Merwe
Department of Metallurgical Engineering, University of Stellenbosch, Stellenbosch, South Africa
J. S. J. van Deventer
Department of Metallurgical Engineering, University of Stellenbosch, Stellenbosch, South Africa
M. A. Reuter
Institut für Metallhüttenwesen und Elektrometallurgie der RWTH Aachen, Aachen, West Germany

SYNOPSIS

As the dissolution of gold from its ores is an ill-defined problem, the use of knowledge-based computing is proposed for the simulation of various batch and continuous leaching systems. The proposed system consists of three distinct modules:

1) a data base which contains all the information of the different ores, and which can be updated as more knowledge becomes available;

2) an expert section where all available knowledge on known ores are used to identify, classify and compile heuristic data for unknown ores;

3) a simulator which can be used to predict leaching behaviour in both mixed reactor and countercurrent leaching systems by implementing a simple n-th order rate expression. The parameters in this expression are used to characterize each type of ore at a specific set of leaching conditions.

As leaching proceeds, the pH, [CN\(^-\)], level of dissolved O\(_2\), etc. may change, which leads to an evolution of leaching conditions. In this simulator, the evolution has been discretized with respect to time and new parameters have to be retrieved at the end of each time increment.

The proposed system produces realistic simulations. It is evident that the accuracy of prediction will be entirely dependent on the accuracy and population density of the data base. The selective use of knowledge has reduced CPU time drastically in comparison with published empirical models, which involve complicated parameter estimation.

INTRODUCTION

Although a vast amount of expertise has been accumulated over the years, the process of dissolution of gold from its ores can still be regarded as an ill-defined problem, mainly, because there is no quantitative mathematical model to describe the process accurately. Also, the complex chemistry associated with cyanidation of real ores is not understood adequately.

The importance of gold in the South African economy cannot be over-emphasized. However, few major technological developments such as the Carbon-in-Pulp process have been introduced. With escalating capital and running costs, as well as rapidly decreasing ore grades, improved recoveries have become essential. This means that new innovations will be required.

The leaching process in the gold recovery circuit has long been identified as a particularly weak link which is seriously hampering further improvement of recovery efficiency.

The kinetic modelling of gold dissolution has received considerable attention in the past, but no attempt to formulate a phenomenological model for this unit operation has yet been published. The reason for this is the complex chemistry of the process as described by Adamson\(^1\) and King\(^2\). They recognised the importance of well controlled high levels of cyanide and oxygen, but did not propose a comprehensive model.

Lorenzen\(^3\) studied galvanic interactions during the electrochemical dissolution process while Gasparrini\(^4\) described the effect of ore mineralogy on the leaching
process. It is obvious that the degree of liberation of gold from its associated minerals will have a pronounced effect on leaching kinetics. Various authors\textsuperscript{5-8} suggest that pretreatment of the ore (i.e. flotation and roasting) can also influence the leaching process. In some ores, pregr-robbing inhibits the rate of leaching due to the re-adsorption of dissolved gold onto the ore constituents such as clays and organic material.

In addition to the factors mentioned above, other process parameters, not discussed in this paper, such as pH, particle size, viscosity, etc. can influence the process to some extent. Nevertheless, the levels of free cyanide and oxygen in the leaching reactor appear to be the most prominent factors influencing the process.

As a result of this complex chemistry and little insight into the kinetics of the process, gold leaching plants are seldom optimally designed, and process control is usually inefficient. In order to improve process control, a number of empirical correlations has been proposed.\textsuperscript{10,11} These methods use historical plant operating data to effect process control. Normally, a number of key variables are identified and included into a mathematical model of the process. These existing rate expressions are applicable to only a limited range of plant conditions, and Brittan and Van Vuuren\textsuperscript{10} pointed out the need for a unique mathematical model for each individual plant. Most of these models have a relatively large number of parameters that have to be estimated in order to fit the data. Furthermore, as these models are developed only after the commissioning of the plant, they cannot be used in the plant design calculations.

However, this type of ill-defined problem is not unique to the gold industry and receives increasing attention in the literature.\textsuperscript{10-15} Recently, knowledge-based systems have found increasing application for treating other ill-defined engineering problems. Although considerable attention is paid in the literature to the application of expert systems and artificial intelligence to chemical engineering problems, little emphasis has yet been placed on such concepts in the minerals industry.

A computer system which stores knowledge explicitly and manipulates that knowledge to solve problems in a specific domain, is referred to as a knowledge-based system.\textsuperscript{12} Schect et al.\textsuperscript{13} described a system which comprises of two modules: the first uses theoretical and empirical formulæ, while the second uses knowledge-based rules to analyse the results produced by the first. Reuter and Van Deventer\textsuperscript{14} proposed a knowledge based model to simulate the behaviour of carbon-in-pulp processes and adjust a data base for changes in the characteristics of the carbon and slurry.

It is the objective of this paper to propose the use of knowledge-based computing for gold dissolution. The system comprises of a knowledge-base which controls the input to the theoretical calculation of gold dissolution via an elementary n-th order rate equation. The parameters for this rate expression are calculated directly from batch concentration-time data and operator experience, by use of simple algebra.

The system was written in Turbo Pascal and can be implemented on IBM PC compatible computers. A very large data base could require an AT machine for efficient calculations. This paper describes the overall program structure, the data base, an option to identify and characterize ores, the dynamic equations, and selected examples of simulation results.

**OVERALL STRUCTURE**

**The standard condition:**

The kinetics of the dissolution of gold from cyanided pulps cannot be modelled in a straightforward manner. It has been found experimentally that the following model fits the concentration-time data from a batch test on Witwatersrand ores very accurately:\textsuperscript{15}

\[
\frac{dg}{dt} = \frac{eVd}{d} = -k(g-g^*)^n \tag{1}
\]

\[
\frac{dc}{dt} = d_oV(1-\varepsilon)k(g-g^*)^n \tag{2}
\]

Therefore:
\[
\frac{dc}{dt} = \frac{d_0(1-e)}{d_1 e} k(g^*-g)^n
\]  

(3)

Thus, by performing a batch test on the ore, this simple first order model can be fitted to the experimental data, and the leaching parameters \( k, n \) and \( g^* \) may be estimated by regression analysis.

However, as the leaching parameters are all functions of the mineralogy and leaching conditions, these data will apply only to one set of conditions. Furthermore, when the effect of each process variable is to be determined, the interactions from the other variables must also be taken into account. By viewing the leaching condition not as consisting of a number of different process variables, but as one unit, the net effect of such a condition on the leaching parameters may be determined. For this purpose, a standard starting condition for the batch process can be defined for each ore type (Table I).

**Table I: Standard leaching condition**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Std Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cyanide level</td>
<td>250</td>
<td>(g/t)</td>
</tr>
<tr>
<td>Oxygen level</td>
<td>8.2</td>
<td>(g/t)</td>
</tr>
<tr>
<td>Gold in solution</td>
<td>0</td>
<td>(g/t)</td>
</tr>
<tr>
<td>Ore grade</td>
<td>6.4</td>
<td>(g/t)</td>
</tr>
<tr>
<td>Liquid fraction</td>
<td>45</td>
<td>(%)</td>
</tr>
<tr>
<td>Temperature</td>
<td>20</td>
<td>(°C)</td>
</tr>
<tr>
<td>pH</td>
<td>12</td>
<td>(-)</td>
</tr>
<tr>
<td>Particle size</td>
<td>-74</td>
<td>(micron)</td>
</tr>
<tr>
<td>Viscosity</td>
<td>300</td>
<td>(cP)</td>
</tr>
<tr>
<td>Ore density</td>
<td>2500</td>
<td>(kg/m³)</td>
</tr>
</tbody>
</table>

Any condition which differs in whatever way from the above, will be seen as a non-standard condition. It follows therefore, that any change in the leaching parameters \( k, n, g^* \), caused by a non-standard condition, must be seen as the net effect of deviations from the standard condition, and therefore as taking all interactions into account. This change in the leaching parameters is expressed as a percentage, and is estimated through experience, or directly from plant, heuristic data or experimental data.

Furthermore, when two different ores exhibit the same leaching characteristics at the standard leaching condition, they may have the same mineralogy. This can then be used as a means of classifying an unknown ore in terms of known ores as will be described later.

It should be noted that the variables used in this definition can be edited, i.e. additional variables can be added.

**The System:**

From the above discussion it would be clear that a knowledge-based system for simulating gold leaching processes should comprise of the following components illustrated in Figure I:

- A data base for entering/editing all the relevant experimental and heuristic data;
- An inference procedure for utilizing the data in the data base to effect the identification and classification of unknown ores, and simulating of different leaching systems;
- A working memory which contains the current input and status of the specific problem being solved. In order to increase the working memory, the different modules of the system communicate via files.
- A user friendly interface used to communicate with the above three components.

The system is initiated via the Main Menu. Initially the DATA BASE must be run to supply appropriate kinetic and heuristic data.

In the data base, the different ores are classified in terms of their basic mineralogy (i.e. refractory, sulphidic, oxide, quartzitic, etc.). After choosing the ore being treated, the concentration-time data from the batch test performed at the standard condition can be entered. Equation 1 is then fitted to the data to provide the leaching parameters at the standard condition.

Following this, the heuristic data can be entered. These data can be divided into two matrices which are coupled through the matrix index: (1) a matrix consisting of leaching conditions different from the standard condition, and (2) a matrix containing the percentage
change in the leaching parameters due to deviations from the standard condition as defined in (1).

The second matrix also contains two factors describing the depletion of the oxygen and cyanide levels. As a result of the leaching reaction, reactants are being consumed, and products formed. This causes the leaching condition in the reactor to be changing continuously. It is therefore also necessary to adjust the leaching condition in the reactor as time proceeds. For the purpose of this paper, the levels of oxygen and cyanide in the pulp have been identified as having a large effect on the leaching parameters \((k,n,g^*)\). The factors influencing the concentration of these two variables are numerous, and depletion factors based on experience have been included in the heuristic data.

After saving the heuristic data, the effects of individual process variables on the leaching characteristics can be entered. These data are normally obtained by experiment and are used whenever new unknown ores must be classified. The entering of these data will normally require a number of additional batch tests, which must be carried out with all the process variables at the standard condition, while only the variable under consideration must be changed. At least four tests must be done for each variable to be considered if a wide range of data is required.

After saving the data, and subsequently returning to the main menu, either the IDENTIFY or SIMULATE options can be selected.

The IDENTIFY option leads to a pop-up menu which enables the user to classify an unknown ore in terms of known ores. In this case, the effects of the individual process variables, entered through the DATA BASE, are used extensively. As is the case with all knowledge-based systems, the accuracy of this type of operation is heavily dependent on the scope and quality of the data available.

The classification is done by comparing the leaching characteristics of the unknown ore with those of known ores at the standard condition. Any known ore, which exhibits the same leaching characteristics as the unknown ore, is then further investigated to compare the effects of the different process variables on the leaching characteristics.

As the title implies, the SIMULATE option gives access to the simulation of different leaching systems. Extensive use is made of the heuristic data entered through the DATA BASE. As previously mentioned, the condition in the reactor is changing continuously. Because of the dependency of the model parameters \((k,n,g^*)\) on the leaching conditions, the condition in the reactor must be estimated before the leaching parameters can be determined. For the purpose of this paper, only the depletion of oxygen and cyanide will be considered, although the model is capable of considering any other change with time.

It is assumed that the depletion of oxygen and cyanide (including oxidative decomposition of \(CN^-\) in air) proceeds through a first order reaction:

\[
\frac{dC_a}{dt} = -bc_a
\]  

(4)

This can then be solved analytically for the case of the batch reactor to yield equation (5):

\[
C_a = C_{ao}e^{-bt}
\]

(5)
THE DATA BASE

Figure 2 depicts the menu for the data base.

As mentioned previously, the different ores in the data base are classified in terms of their basic mineralogy (i.e. refractory, sulphidic, oxide, quartzitic, etc.). This approach provides a pointer to the data used in subsequent steps. Choosing any ore will invoke a checking procedure. This procedure checks the current status of data available on the chosen ore. The results of this check are shown in the status screen. (Figure 2 reflects the status of the ore named VRE3. The value of '1' in front of the ore name points to the ore mineralogy: refractory in this case) Initially all the values in this window are FALSE and will only change to TRUE if the minimum amount of data needed is available. After choosing the ore being treated, the concentration time data at the standard set of reaction conditions should be entered.

Once the concentration-time data have been entered, the leaching curve (equation 1) could be fitted through the data to produce the standard leaching parameters: \( k_{\text{std}} \), \( n_{\text{std}} \) and \( g^*_{\text{std}} \). These data are stored and used in subsequent steps of the programme.

As will be seen later, a very basic knowledge of the effects of different process variables on the standard leaching curve is necessary to classify unknown ores in terms of known ores. The various types of effects used in this system are listed below:

Linear:

\[ q = \phi_0 + \phi_1 z \]

Polynomial:

\[ q = \phi_0 + \phi_1 z + \phi_2 z^2 + \phi_3 z^3 + \ldots \] (7)

Exponential:

\[ q = A e^{\phi_2 z} \] (8)

Table II: Effect of temperature (Deg C) on the standard leaching curve of 1VRE3.

<table>
<thead>
<tr>
<th>Leaching Parameter</th>
<th>Effect</th>
<th>Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>( k )</td>
<td>Linear</td>
<td>( \phi_0 = 0.2 )</td>
</tr>
<tr>
<td>( n )</td>
<td>Polynomial</td>
<td>( \phi_0 = 0.2 ), ( \phi_1 = 0.3 ), ( \phi_2 = 0.12 )</td>
</tr>
<tr>
<td>( g^* )</td>
<td>Exponential</td>
<td>( A = 0.1 ), ( P = 0.07 )</td>
</tr>
</tbody>
</table>

The next option in the data base menu is the editing of heuristic data (Figure 4). As described previously, the heuristic data consist of two matrices which are coupled through the matrix index. As can be seen, the condition shown in the menu differs from the standard condition (Table I). As a result, the value of the leaching parameters \( k \), \( n \) and \( g^* \) will change by the percentages shown. The changes shown here are the net effect of the total change from the standard condition. This implies that, for the simulation of the different leaching systems, a detailed knowledge of the interactions between the
process variables is unnecessary.

![](image)

Figure 4: Editing of heuristic data

THE IDENTIFY OPTION

When the leaching of an unknown ore is to be simulated, it can be done only after obtaining a substantial amount of data on the leaching characteristics of the ore. This is normally done by extensive experimentation and trial runs which are, invariably, a very lengthy and expensive process. The IDENTIFY option enables the user to take advantage of available data on known ores to estimate leaching characteristics and optimum plant conditions for unknown ores. For this exercise, a small number of experiments are needed as each new ore is introduced into the system.

Any two ores, leached under the standard condition, will exhibit their own standard batch leaching curves. When the two curves differ widely, it can be said that the two ores are quite different. When the curves differ by only a slight margin, it could be said that the ores may be similar. However, because leaching curves are not unique to a specific gold ore, a small margin of difference cannot be taken as enough evidence that the two ores are quite the same. To complete the comparison between the ores, the effects of the various process variables on the leaching curves must be investigated. If there is still close conformity, the two ores can be considered to be similar for practical purposes.

Experience has shown, that although there are numerous process variables influencing gold dissolution, a relatively small number can be viewed as more important (See Table I). When investigating the effects of the various process variables, no attention is given to the interaction between the different variables. This means that all but the variable under consideration, must be kept at the standard condition. The results of the experiments must be presented in such a way to describe the type of effect, i.e. linear, polynomial or exponential (Table II).

This reasoning can be applied to a situation where a new, unknown ore is introduced into the system. The only data available on this ore, are the standard batch leaching curve, and the effects of a number of process variables on the leaching parameters \((k,n,g^*)\). After entering these data into the data base, the IDENTIFY option can be used to compile heuristic data for the simulation of leaching characteristics in the following way:

The first step involves the setting of a sensitivity level for the comparison of the unknown ores to the known ores. This indicates the margin of difference that will be tolerated, and will depend on the number of known ores available. The larger the number of known ores, the smaller the level of tolerance.

For the next step, the standard leaching parameters of the unknown ore are compared to those of the known ores of similar mineralogy. Only those known ores, with standard leaching characteristics differing by a margin of not more than the given level from the unknown ore, will be examined further.

The following step involves the comparison of the effect of each of the process variables on the leaching parameters. When the search is completed, the ores selected could be used to estimate the leaching characteristics of the unknown ore. The different sets of data are then combined to produce one set of heuristic data for the unknown ore. This is then stored and can be updated by the DATA BASE.

THE SIMULATE OPTION

On selecting this option, the user is prompted to choose from the following systems:

- Perfectly mixed batch reactor,
- Continuous mixed reactors in series, and
- Continuous counter-current column.

Each of these will be discussed in detail below:
Perfectly mixed hatch reactor

For the purpose of this paper the dissolution reaction is assumed to proceed according to equations 1 and 3.

These equations can be solved simultaneously by using a fourth order Runge Kutta method. However, to do this, the values of the leaching parameters must be known. As the reaction proceeds, the reaction conditions will change, and hence also the leaching parameters. As there are no mathematical equations available to describe the dependency of the leaching parameters on the leaching conditions, the data must be extracted from the data base. This is done in the following way:

When commencing the reaction, the initial set of leaching conditions is known (P1). With P1, the data base is searched for the set of reaction conditions which is in best agreement with P1. Call this P1a. It should be noted that only the heuristic data applicable to the specific ore are searched. The condition P1a corresponds to a certain matrix index, which in turn acts as a pointer to the following five values:

- Change in k (kv) (%)
- Change in n (nv) (%)
- Change in g' (gv) (%)
- Cyanide depletion factor \( (hr^{-1}) \)
- Oxygen depletion factor \( (hr^{-1}) \)

Thus, in order to calculate the concentration-time data for the batch reactor, the following steps must be taken:

1. Get the standard leaching parameters for the ore being treated.
2. Get the current plant conditions, (P1).
3. Search the data base for the set of heuristic conditions which best resembles the current plant conditions, (P1a).
4. At P1a retrieve the factors kv, nv and gv to calculate the leaching parameters k, n and g'. Solve equation (1) and then equation (3) using a fourth order Runge Kutta method.
5. At P1a retrieve the depletion factors for cyanide and oxygen and solve equation (5).
6. Check numerical stability by performing a total gold mass balance on the reactor system.
7. Increase the time by one increment and repeat process from step (1).

Because the depletion factors for cyanide and oxygen are continuously updated, pseudo first order reaction kinetics can be assumed.

Continuous mixed reactors in series

In a series of perfectly mixed reactors, all reactors can be viewed as identical, with the output of the first reactor being the input of the next reactor. Non-ideal flow characteristics in the reactor system have also been incorporated, as shown in figure 5. The values of the theoretical dead volume and bypass streams are given as percentages. Such values can be obtained from tracer tests, or estimated by comparing real plant behaviour with that predicted from a flow model. By using this system, the volume and flow through the reactor can be calculated as follows:

\[
V = V(1 - V_d) \quad (9)
\]

\[
F = F(1 - F_b) \quad (10)
\]

Similar to the batch reactor the following equations can be written for the dynamic simulation of the ith reactor in the system:

**Solid phase:**

\[
\frac{dg_1}{dt} = \frac{u}{Vd_1} L \left( g_{1-1} - g_1 \right) - k(g_1 - g^*)^n \quad (11)
\]

**Liquid phase:**

\[
\frac{dc_1}{dt} = \frac{u}{\epsilon V} \left( c_{1-1} - c_1 \right) + \frac{(1-\epsilon)d}{\epsilon k(g_1 - g^*)} \quad (12)
\]
If the kinetic parameters for the depletion of the cyanide and oxygen are assumed to be identical to those in the batch reactor, then the concentration-time data can be calculated in the same way as for the batch reactor by utilising equations (11) and (12). The process would start at reactor number 1 and be repeated through to the last reactor, before the next time interval is solved.

The continuous countercurrent column

In this type of operation the pulp enters the column at the top and leaves at the bottom, while the clear leaching solution enters at the bottom and leaves as clear solution at the top.

In modelling this operation two distinct streams were defined: i) the pulp stream containing a mixture of leaching solution and solids, and ii) the clear leaching solution stream. As the leaching solution leaving the top of the column contains no solids, it is assumed that the solid phase exhibits only a downward flow. However, the liquid phase can move in either direction. It is therefore possible that the entering pulp can have a lower specific gravity than the pulp leaving at the bottom, or vice versa.

By viewing the column as a series of non-ideal reactors containing a CSTR unit, it is possible to facilitate a complex flow pattern. As can be seen from figure (6b), dead volume on each reactor and a bypass on the pulp stream are used to include non-ideal flow characteristics.

It should be clear that before any attempt to calculate the concentration-time, the various streams throughout the reactor must be calculated. In order to avoid an iterative solution, it is assumed that any change in specific gravity takes a linear profile through the reactor. When the number of mixed reactors used in the simulation is known, the change in the liquid fraction for each mixed reactor can be calculated.

As the flow of solids is only downward and constant, it can be shown that:

$$y = (F_{pi-1}) (F_{bi}) (\epsilon_{i-1}) \quad (13)$$

$$x = \left[ F_s \epsilon_i - y (1 - \epsilon_i) \right] / (1 - \epsilon_i) \quad (14)$$
\[ F_{p_i} = F_S + (F_{p_{i-1}})(F_{b_i})(\epsilon_{i-1}) + x \]  

(15)

\[ F_{L_{i+1}} = (F_{p_i})(\epsilon_i) + F_{L_{i}} - (F_{p_{i-1}})(\epsilon_{i-1}) \]  

(16)

With these equations all relevant streams can be calculated. In order to save memory space, these calculations must be performed for each cycle. As these are straightforward calculations, they do not take up much CPU time.

Under normal operating conditions only the inlet conditions at opposite ends of the column are known. This causes some difficulty in the numerical work, because the mass balance equations require values at both sides of the mixed reactor system. This has been overcome by making use of the ability of the Runge-Kutta method to use historical data to estimate future data. Firstly, a short summary is presented of the equations needed:

**For steady state:**

From a mass balance for gold around the \( i \)th reactor on the solid phase:

\[ F_S(1-F_{b_i})(g_{p_{i-1}}) = F_S(1-F_{b_i})g_i + k(g_i-g^*)nVd_o(1-\epsilon_{iR}) \]  

(17)

Then, a gold balance in the liquid phase, at the mixing point MP in Figure 6(b):

\[ C_{p_{i}} = (F_{p_{i-1}})(F_{b_i})(\epsilon_{i-1})(C_{p_{i-1}}) + xC_i/\{(F_{p_i})(\epsilon_i)\} \]  

(18)

A gold balance in the solid phase around point MP yields:

\[ g_{p_{i}} = ((F_{b_{i-1}})(F_{b_i})(1-\epsilon_{i-1}))(g_{p_{i-1}}) + (F_{p_{i-1}})(1-F_{b_i})(1-\epsilon_{i-1})g_i/\{(F_{p_{i}})(1-\epsilon_i)\} \]  

(19)

Then, similar to equation (17), a gold balance in the liquid phase around the \( i \)th reactor yields:

\[ C_{i+1} = (xC_i + F_{L_i}C_i - k(g_i-g^*)nVd_o(1-\epsilon_{iR}) - (F_{p_{i-1}})(1-F_{b_i})(\epsilon_{i-1})(C_{p_{i-1}})/\{(F_{p_{i}})(\epsilon_i)\} \]  

(20)

A similar approach is followed in calculating the cyanide and oxygen concentration:

For the cyanide/oxygen in the liquid phase around the \( i \)th reactor:

\[ C_{A{Li}+1} = (xC_{A_i} + F_{L_i}C_{A_i} + bC_{A_i}Vd_1(\epsilon_{iR}) - (F_{p_{i-1}})(1-F_{b_i})(\epsilon_{i-1})(C_{A_{pi-1}})/\{(F_{p_{i}})(\epsilon_i)\} \]  

(21)

For the pulp stream around the \( i \)th reactor:

\[ C_{A_{pi}} = ((F_{p_{i-1}})(F_{b_i})(\epsilon_{i-1})(C_{A_{pi-1}}) + (xC_{A_{i}}))/\{(F_{p_{i}})(\epsilon_i)\} \]  

(22)

**Simulation of gold dissolution**

To estimate the dynamic behaviour of the ore in the countercurrent column reactor, the following steps must be taken:

1. Get the number of mixed reactors used in the simulation.
2. Get the current plant conditions.
3. Estimate the outlet conditions for the pulp stream at the bottom of the column.
4. Starting at the bottom, calculate all the relevant streams for reactor number \( r \), using equations 13-16.
5. Get the leaching conditions in reactor number \( r \). With these conditions, retrieve the necessary data from the data base as for the batch reactor. Calculate the relevant mass balances using equations 17 - 22. Estimate the levels of oxygen and cyanide for the next reactor.
Proceed to the next reactor until the top of the column is reached. Compare the calculated outlet values for the clear stream with the given values. If the difference is larger than a given tolerance, repeat the procedure from step 3. If the difference is smaller than the given tolerance, check for numerical stability. If the stability is acceptable, proceed to the next time increment. If a numerical instability has occurred, terminate the simulation, and prompt the user for a smaller time interval.

SIMULATION RESULTS

As the system can still be improved, the field data available at present are not sufficiently extensive. This data base will have to be extended for ores from specific regions. However, the available data have been entered into the data base in order to test the system. The following are some examples of the type of results that can be obtained:

Experimental conditions:

Throughout the experiments, the system was tuned to use the standard condition as the leaching condition. Deviation from this condition was effected by changing only one variable at a time. This approach provides easier management of experimental data.

Batch reactor

For the batch reactor, a total reaction time of 18 hours was used for experiments on a Witwatersrand ore. As can be seen from figure 7, the data show typical leaching characteristics.

As expected, a higher initial cyanide level will give a faster and better recovery. When the system was used to simulate the same conditions, excellent agreement with the experimental data was obtained (Figure 7). Similar results were obtained when the effect of oxygen was investigated (Figure 8).

Continuous leaching in a series of mixed reactors

In this case, the following values were entered into the system via the various menu options:

- Number of reactors: 11
- Volume of each reactor: 400 m³
- Dead volume per reactor: 10%
- Bypass per reactor: 10%
- Flowrate: 10t/hr

The leaching profile is given in figure 9. From this figure it is clear that very little leaching is taking place in the last two reactors at steady state. It would seem therefore that this plant was overdesigned. By specifying various scenarios, the system can be used in design applications. As will be indicated later, the dynamic response of the system can be tested as well.
Figures 10 and 11 present examples of some scenarios which can be investigated. As can be expected, a higher flowrate will result in a reduced recovery (Figure 10). This is mainly due to a decrease in the actual reaction time in the reactor, but can also be as a result of an increased bypass or dead volume formation in practice. It should be clear that, by using the system, a sensitivity analysis can be performed to investigate ill-defined areas such as the effect of non-ideal flow.

A very recent development in gold leaching is to control the level of free cyanide in the leaching pachucas. Any such system can be simulated easily by varying, either the actual flow pattern definition via the menu options, or by adjusting the depletion factor for cyanide (Figure 12). This can be very helpful when improvements on leaching plants are planned.

The dynamic simulation of the continuous process can be used to investigate the response of the leaching system due to a sudden change in leaching conditions. For example, if the flowrate should be increased suddenly from a lower to a higher value, a response such as figure 13 can be expected. For simplicity, only three reactors are given here. It is clear that the reaction time for the first reactor is shorter than that for the third reactor. By analysing the response of the leaching system being investigated, it should be possible to provide for such instabilities in order to prevent gold losses.

Continuous countercurrent column

Simulating the counter-current column is more time consuming than any of the other leaching systems. This comes as a result of the more complex flow pattern existing in the column.

Here the leaching system used consisted of one column which was subdivided in eleven mixed reactors as described earlier. The flowrate was adjusted to 10 tonne pulp per hour, and the column diameter to 5 m. A total volume of 400 m$^3$ for the reactor was assumed.

As this is a relatively new development in gold dissolution technology, very little industrial data could be obtained. However, when using the system at the steady state, realistic results were produced. As can be seen...
from figure 14, the ore grade is depleted as it moves down the column. In contrast with this, the cyanide level decreases from the bottom to the top. Figure 15 illustrates the effect of a higher initial level of cyanide fed to the column.

Simulating the column in a dynamic way can be done in much the same way as for the mixed reactor system. This process takes much longer to simulate on a PC, because of the more complex flow pattern and repetitive numerical solution of coupled differential equations.

CONCLUSIONS

The dissolution of gold is an ill-defined process and can therefore not be simulated in a straightforward way by means of mathematical equations only. The system proposed here consists of three modules:

1) A data base which contains all the information of the different ores;

2) An expert section where all available knowledge on known ores are used to identify, classify and compile heuristic data for unknown ores;

3) A simulator which can be used to predict leaching behaviour in both mixed reactor and countercurrent leaching systems by implementing a simple n-th order rate expression.

It can be concluded that, although only limited field data are available at present, the system appears to give realistic simulations of the following reactor arrangements:

- Batch reactor,
- Continuous mixed reactors in series,
- Continuous countercurrent column.

In addition to this, a sensitivity analysis on any realistic change in the leaching conditions can be performed. Thus, the effect of any process variable on the recovery efficiency of gold can be investigated. This can be very useful when designing new gold leaching plants.

As expected, the system is heavily dependent on the accuracy and population density of the available data. By using heuristic data about the process, a detailed understanding of the various interactions between the process parameters is unnecessary. Furthermore, heuristic data for unknown ores can be compiled by utilising the knowledge about other, well known, ores. This would normally require a small number of additional batch leaching experiments.

Finally, it would seem that for a mathematically ill-defined process such as gold leaching, the ease with which experimental data are manipulated, makes the knowledge-based system easier to apply and more practical than previously published models, which usually require the estimation of a large number of parameters.

NOMENCLATURE

A Constant in exponential expression

b Rate constant for cyanide and oxygen depletion (hr⁻¹)

[CN⁻] Concentration of free cyanide in liquid phase (g/t liquid)
\( [\text{CN}^-]_0 \) Initial concentration of free cyanide in liquid phase (g/t liquid)
\( \dot{c} \) Concentration of gold in liquid phase (g Au/t liquid)
\( \dot{z}_A \) Concentration of cyanide or oxygen (g/t liquid)
\( \dot{z}_A^{\text{Li+1}} \) Concentration of cyanide or oxygen in clear stream at inlet to reactor i (g/t liquid)
\( \dot{z}_A^{L_0} \) Initial concentration of cyanide or oxygen (g/t liquid)
\( \dot{z}_A^{\text{Pi}} \) Concentration of cyanide or oxygen in pulp stream at inlet to reactor i (g/t liquid)
\( \dot{z}_p \) Gold concentration at pulp inlet (g Au/t liquid)
\( \dot{z}_L \) Gold concentration at leaching solution inlet (g Au/t liquid)
\( \dot{z}_L^{\text{Li+1}} \) Gold concentration in clear stream at inlet to reactor i (g Au/t liquid)
\( \dot{t}_o \) Density of ore (t/m\(^3\))
\( \dot{t}_L \) Density of leaching solution (t/m\(^3\))
\( \dot{f}_B \) Flowrate of pulp (ton/hr)
\( \dot{f}_{\text{bi-1}} \) Bypass (%) of reactor i-1 (%)
\( \dot{f}_L \) Flowrate of clear stream (t liquid/hr)
\( \dot{f}_{\text{Li+1}} \) Flowrate of clear stream into reactor i (t liquid/hr)
\( \dot{f}_{\text{Li-1}} \) Flowrate of clear stream leaving reactor i (t liquid/hr)
\( \dot{f}_P \) Flowrate of pulp stream (t pulp/hr)
\( \dot{f}_{\text{Pi-1}} \) Flowrate of pulp stream entering reactor i (t pulp/hr)
\( \dot{f}_{\text{Pi+1}} \) Flowrate of pulp stream leaving reactor i (t pulp/hr)
\( \dot{q} \) Grade of ore (g Au/t ore)
\( \dot{q}_0 \) Initial grade of ore (g Au/t ore)
\( \dot{g}^r \) Refractory gold in ore (g Au/t ore)
\( \dot{g}_v \) Change in g (%) of ore
\( \dot{k} \) Rate constant for gold dissolution
\( \dot{k}^v \) Change in k (%)
\( \dot{n} \) Exponent in eq(1) for gold dissolution
\( \dot{n}^v \) Change in n (%)
\( \{O_2\} \) Concentration of dissolved oxygen in liquid phase (g/t liquid)
\( \{O_2\}_0 \) Initial concentration of dissolved oxygen in liquid phase (g/t liquid)
\( \dot{p} \) Coefficients in eq(6)-(8).
\( q \) Effect of process variable (eq. 6-8)
\( t \) Time (hrs)
\( v \) Volume (m\(^3\))
\( v_d \) Dead volume (%)
\( x \) Liquid from reactor i (t liquid/hr)
\( y \) Liquid from bypass around reactor i (t liquid/hr)
\( z \) Process variable in eq(6)-(8)
\( \dot{e} \) Liquid fraction (%)
\( \dot{e}_0 \) Initial liquid fraction (%)
\( \dot{e}_i \) Liquid fraction in reactor i (%)
\( \dot{e}_{i-1} \) Liquid fraction of pulp entering reactor i (%)

REFERENCES


XXII. Internationales Symposium APCOM
17. - 21. 9. 1990
Berlin, Bundesrepublik Deutschland

XXII. International Symposium APCOM
17. - 21. 9. 1990
Berlin, West - Germany

XXII Международный симпозиум APCOM
17 по 21 сентября 1990
Берлин, Федеративная Республика Германия

Band I
Volume I
том I
Ein wissensbasiertes System zur Simulation kampagnenartig und kontinuierlich arbeitender Carbon-in-Leach Systeme

M.A. Reuter
J.S.J. van Deventer

University of Stellenbosch, Stellenbosch, Südafrika


A Knowledge Based System for the Simulation of Batch and Continuous Carbon-in-Leach Systems

M.A. Reuter
J.S.J. van Deventer

University of Stellenbosch, Stellenbosch, South Africa

A knowledge based model is proposed here for the simulation of adsorption, leaching and a combination of these in the form of a Knowledge Based System. The basis of the model is a database which manipulates concentration-time data, which covers adsorption up to the maximum equilibrium loading possible and leaching up to final leach grade for adsorption and leaching processes respectively. These data can subsequently be used to perform fault diagnosis, economic evaluation and process simulation of batch and continuous systems. The kinetic model used for the process simulation is identical for both adsorption and leaching and its one parameter is a function of the data in the database. In contrast with previously published models, no curve fitting is performed, the one parameter is determined directly from the concentration-time data by simple algebra. This permits the formulation of a generalized model which could be used for the simulation of any leaching and any adsorption system under any chemical process conditions. The application of this model is limited in this paper to carbon-in-pulp, leaching and carbon-inleach systems. Predictions of the
Научно обоснованная система для симуляции цикличных и нецикличных СИЛ-систем

М.А. Ройтер
Й.С.Й. ван Девентер

Университет Стелленбосх, г. Стелленбосх, ЮАР

Предлагается научнообоснованная модель для симуляции, адсорбции и выщелачивания и для сочетания обеих форм научнообоснованной системы. Основой модели служит база данных которая обеспечивает обработку данных имеющихся во времени обогащения и которая в адсорбционных процессах гарантирует адсорбцию до максимально достижимого веса загрузки либо в процеессе выщелачивания наивысшую степень выщелачивания. Кроме того те же данные могут быть использованы для диагностики ошибок, экономической оценки и для симуляции процессов, как цикличных так и нециклично работающих систем. Использованная для симуляции процессов кинетическая модель идентична, как для адсорбции так и для выщелачивания и ее единственным параметром является функция данных, содержащаяся в банке данных. В отличии от ранее опубликованных моделей не осуществляется адаптация кривых, а параметр определяется с помощью простой алгебры непосредственно из имеемсяся во времени обогащения данных. Такое позволяет формулировать общую модель, служащую для симуляции любых процессов выщелачивания и адсорбции при любых химических условиях. Как пример применения в настоящей публикации представляются "carbon - in - pulp, leaching and carbon - in - leach - systems". Прогнозами предложенной научнообоснованной модели представляются - при хорошем соответствии - ранее опубликованные данные.
INTRODUCTION

Two major disadvantages of the gold adsorption and leaching models which have been formulated to date are (i) their large number of parameters (equilibrium isotherms and kinetic parameters) which have to be determined and (ii) their limited scope of application as these parameters only hold for the experimental conditions under which they were determined.

Numerous models have already been formulated to define the adsorption of gold onto activated carbon e.g. semi-empirical\(^1\)\(^-\)\(^2\), adsorption–desorption\(^3\) and branched pore models\(^4\)\(^-\)\(^5\). Each of these models is characterized by a number of parameters which hold for particular process conditions only. Subsequently, these models have been used to simulate continuous operations\(^2\)\(^-\)\(^6\)\(^-\)\(^8\). These models are, however, not capable of performing fault diagnosis, i.e. establish why a plant is performing in a particular fashion. Neither are these models capable of considering chemical process conditions e.g. pH, ionic strength, carbon activity, or the physical process conditions e.g. the type of mixing, non-ideal flow, non-ideal carbon transfer etc.. The same discussion as above also holds for the established leaching models\(^2\)\(^-\)\(^9\).

The model proposed here consists of a database containing adsorption concentration–time data e.g. (Nicol et al. p.40 which covers the adsorption up to the equilibrium loading)\(^6\) and leaching concentration–time data e.g. (Nicol et al. p.75\(^7\). The reason for defining the model on this basis is fivefold viz.:

- These concentration–time data characterize the adsorption and leaching processes under the prevailing process conditions. A set of slopes and the equilibrium loading, which are produced from these data, facilitate the searching of data, adjustment of data as a function of the chemical process variables and form the basis for fault diagnosis.

- If the effects of the various chemical and physical process conditions on the rate and loading are known, these effects can be included in a very simple procedure to adjust these slopes and equilibrium loadings accordingly. This is done in the form of adjustment factors and polynomials.

- If the chemical process conditions change drastically and adjustment factors are not accurate enough, it is a simple task to search the database for data that correspond closest to the required process conditions.

- The slope and equilibrium data make it possible to define a very simple kinetic model which is identical for both adsorption and leaching, and which may subsequently be used to simulate continuous adsorption and leaching systems under various physical process conditions e.g. reactor arrangement, carbon flow pattern etc..
A general knowledge based model may be defined which is independent of the adsorption or leaching system under consideration, i.e. resin-in-pulp, ion-exchange may also be included, for example.

In association with each of these data sets is an economic data set, which includes price weights for each of the adjustment factors, all physical process variables etc.. This permits e.g. the investigation into the effect of carbon poisoning on the economical performance of the circuit.

To summarize, the model proposed here is based on a database permitting modelling, economic evaluation and simulation under all physical and chemical process conditions catered for in the database. This will be shown in the subsequent sections.

FORMULATION OF MODELS

The database, adsorption (carbon-in-leach), leaching and carbon-in-leach (CIL) models will be formulated subsequently. These models form part of a user-friendly simulation programme, the Knowledge Based System, which is written in Turbo Pascal. This System is driven by mouse via various graphics and text menu interfaces. Runge-Kutta methods are used throughout to perform the numerical integrations.

Database Model

This model cannot be defined in terms of formulae, hence only a general description will be given here. It is characterized by the following activities that it can perform:

- The model can search the database for the required adsorption-time data which corresponds to the inputs. These inputs include:
  - the type of carbon (in total 1125 different sets of data),
  - type of data, i.e. changes in [Au], [Ag], [CN-] etc.,
  - up to 30 user-defined inputs, which may include inputs such as carbon blinding, stirring mechanism, pH, scale-up etc., and
  - 16 of these user-defined inputs may be subdivided into 15 sub-divisions or adjustment factor increments and the balance (14) into regression equations that define the adjustment factors over a range of inputs.

All these inputs may be edited by the user.

- The database model produces the respective rate variables \( k(y(t)) \) and \( k(g(t)) \) from the selected concentration-time data. The sum of the time-weighed \( k(y(t)) \) and \( k(g(t)) \) (equations 1&2) values over a given period of time and the equilibrium loading and final leach grade are the basis for the search of data.

- Continuous leaching, carbon-in-pulp and carbon-in-leach systems may subsequently be simulated. The equations that define this activity will be defined in later sections below. The physical process conditions, which form the inputs to these physical models, include:
  - carbon concentration, pulp density and ore grade,
  - number, size and configuration of reactors,
carbon and pulp flow arrangements, non-ideal flow through reactors, carbon short-circuiting, etc.

Given leaching or adsorption concentration-time data, which have been transformed into slope and equilibrium loading data, the database model can estimate what type of chemical process conditions were prevailed during their estimation. This is a valuable additional tool which can be used during fault finding.

Database house keeping, organization etc. is performed continuously during the various abovementioned activities.

This model is illustrated best by depicting one of the numerous user interfaces in Figure 1. Note that all the factors are unity, since the carbon which was used was assumed to be fresh.

A similar model may be defined for leaching systems.

Adsorption Model

This model has already been formulated in dimensionless form by Reuter and Van Deventer\(^1\) and is given here again, but in practical units. The basis of the model is an adsorption concentration-time and corresponding loading-time curve that covers the adsorption kinetics up to the equilibrium loading capacity e.g. data p.40 Nicol et al.\(^1\). From this curve and the corresponding concentration-time the rate variable \(k[y(t)]\), which is defined here to be only a function of the loading \(y(t)\), may be determined. This rate variable is subsequently used to predict the change in concentration and carbon loading at any carbon concentration \(M_c\), any solution density and any starting concentration at the prevailing chemical process conditions.

\[
\frac{dC}{dt} = -(a_1/a_2)k[y(t)]C = r_1
\]
\[
\frac{dy}{dt} = a_1 k[y(t)] C = r_2
\]

where:
\[
k[y(t)] = -\frac{2(C_{a+1} - C_a)}{\delta t(C_{a+1} + C_a)}
\]
\[
a_1 = \frac{(M/M_c)_1}{\delta t}
\]
\[
a_2 = \frac{(M/M_c)_2}{\delta t}
\]

The rate variable \(k[y(t)]\) is also a function of the database, permitting modelling at chemical process conditions other than those prevailing during which the rate variable was determined. This renders the model totally general.

**Leaching Model**

As for the adsorption model, \(k[g(t)]\), which is defined here to be only a function of the grade \(g(t)\), is determined from a leaching curve, which covers the leaching up to the equilibrium leaching point. As for the adsorption model, \(k[g(t)]\) is also a function of the database.

\[
\frac{dC}{dt} = k[g(t)] g(t) = r_3
\]
\[
\frac{dg}{dt} = -\beta k[g(t)] g(t) = r_4
\]

where:
\[
k[g(t)] = \frac{2(C_{a+1} - C_a)}{\delta t(g_{a+1} + g_a)}
\]
\[
\beta = \frac{(M_c/M_a)}{\delta t}
\]

**Carbon-in-Leaching Model**

As stated by Nicol et al., the adsorption and leaching reactions are independent of one another. Hence the CIL-Model proposed here is a linear combination of the above-mentioned models.

**Continuous Models for CIL-Systems**

The equations that define the change of concentration, loading on the carbon and element content in the ore (equations 5 to 7 respectively) will be given subsequently (see LIST OF SYMBOLS for the meaning of the symbols). Ideal flow is assumed in the formulation of these models (equations 5 to 7), although in the Knowledge Based System non-ideal flow in the form of short-circuiting and dead volume is formulated.

\[
\frac{dC_1}{dt} = \frac{V_{p,in}}{V_{act}} C_1 - \frac{V_{p,out}}{V_{act}} C_1 + r_1 + r_3
\]
\[
\frac{dy_1}{dt} = \frac{m^{CC}_C}{M^{CC}_{C,t}} [y_{1+1} - y_1] + \frac{m^{CC}_C}{M^{CC}_{C,t}} [y_{1-1} - y_1] + r_2
\]
Depending on \(m^{cc_c}, \, m^{cc_c,t}, \, m^{cc_c} \) and \(M^{cc_c,t}\) any type of carbon transfer mode, ranging from batch to continuous, co-current to counter-current or a combination of these may be simulated.

VALIDATION OF THE MODELS AND DISCUSSION OF RESULTS

The results produced by each of the above-mentioned kinetic models will now be compared to results produced by well established models or published industrial data. Note that in all cases ideal transfer of carbon and ideal flow through the stages was assumed.

**Adsorption Model**

The performance of the adsorption model is compared to the well established Dixon Model\(^2\). Table 1 compares the concentration-time data and Table 2 the loading-time data. All entries in the (a) columns give the results produced by the given Dixon Models (pulp density 1460kg/m\(^3\), 1:1 mass ratio solids:liquid). All the predictions at other carbon masses and starting concentrations in the (b) columns are based on the concentration-time data of the entries in the (a) column, which covers the adsorption up to the maximum equilibrium loading i.e. ±3540 (Table 1) and ±10861 (Table 2). (\(\delta t=1h\) was taken to define \(k\{g(t)\}\). Only a few points are shown.)

Table 1

Data in column 1(a) are used to calculate \(k\{y(t)\}\), which is applied to predict the solution gold concentrations [ppm] in all columns (b). The model given by Williams et al.\(^{11}\) was applied.

<table>
<thead>
<tr>
<th>Time</th>
<th>1(a)</th>
<th>1(b)</th>
<th>2(a)</th>
<th>2(b)</th>
<th>3(a)</th>
<th>3(b)</th>
<th>4(a)</th>
<th>4(b)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 g/l</td>
<td>30.0</td>
<td>30.0</td>
<td>30.0</td>
<td>30.0</td>
<td>5.0</td>
<td>5.0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1 g/l</td>
<td>12.2</td>
<td>12.8</td>
<td>25.3</td>
<td>25.1</td>
<td>4.0</td>
<td>4.1</td>
<td>.0027</td>
<td>.004</td>
</tr>
<tr>
<td>0.5 g/l</td>
<td>11.1</td>
<td>11.5</td>
<td>25.1</td>
<td>25.1</td>
<td>3.5</td>
<td>3.2</td>
<td>.0025</td>
<td>0.0</td>
</tr>
<tr>
<td>0.25 g/l</td>
<td>10.6</td>
<td>10.8</td>
<td>25.1</td>
<td>25.1</td>
<td>3.0</td>
<td>3.0</td>
<td>.0025</td>
<td>0.0</td>
</tr>
<tr>
<td>0.1 g/l</td>
<td>10.6</td>
<td>10.6</td>
<td>25.1</td>
<td>25.1</td>
<td>2.7</td>
<td>2.7</td>
<td>.0025</td>
<td>0.0</td>
</tr>
<tr>
<td>0.05 g/l</td>
<td>10.6</td>
<td>10.6</td>
<td>25.1</td>
<td>25.1</td>
<td>2.7</td>
<td>2.6</td>
<td>.0025</td>
<td>0.0</td>
</tr>
<tr>
<td>0.01 g/l</td>
<td>10.6</td>
<td>10.6</td>
<td>25.1</td>
<td>25.1</td>
<td>2.7</td>
<td>2.6</td>
<td>.0025</td>
<td>0.0</td>
</tr>
</tbody>
</table>

\[\frac{dy}{dt} = 0.12C(3600-y)-0.022y\] (b):Eq.1&2

\[\frac{dC}{dt} = \frac{V_{p, in}}{V_{act}} - \frac{C_{in}}{V_{act}} \]
Table 2  Concentration-time data corresponding to the loading data in column 1(a) were used to calculate \( k[y(t)] \), which is applied to predict the loadings in [ppml] on the carbon in all columns (b).

\[
(a): \frac{dy}{dt} = 0.014C(11000-y)-0.004y \quad (b): Eq.1&2
\]

<table>
<thead>
<tr>
<th>Time</th>
<th>1(a)</th>
<th>1(b)</th>
<th>2(a)</th>
<th>2(b)</th>
<th>3(a)</th>
<th>3(b)</th>
<th>4(a)</th>
<th>4(b)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Me</td>
<td>0.5 g/l</td>
<td>5 g/l</td>
<td>15 g/l</td>
<td>15 g/l</td>
<td>15 g/l</td>
<td>15 g/l</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cs</td>
<td>30 ppm</td>
<td>1 ppm</td>
<td>1 ppm</td>
<td>15 ppm</td>
<td>15 ppm</td>
<td>15 ppm</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>3628</td>
<td>3600</td>
<td>94</td>
<td>92</td>
<td>46.5</td>
<td>46.1</td>
<td>693.5</td>
<td>692</td>
</tr>
<tr>
<td>2</td>
<td>5909</td>
<td>5893</td>
<td>127</td>
<td>126</td>
<td>48.5</td>
<td>48.5</td>
<td>727.2</td>
<td>727</td>
</tr>
<tr>
<td>4</td>
<td>8427</td>
<td>8413</td>
<td>143</td>
<td>143</td>
<td>48.6</td>
<td>48.6</td>
<td>729.0</td>
<td>729</td>
</tr>
<tr>
<td>9</td>
<td>10400</td>
<td>10390</td>
<td>145</td>
<td>145</td>
<td>48.6</td>
<td>48.6</td>
<td>729.0</td>
<td>729</td>
</tr>
<tr>
<td>19</td>
<td>10844</td>
<td>10835</td>
<td>145</td>
<td>145</td>
<td>48.6</td>
<td>48.6</td>
<td>729.0</td>
<td>729</td>
</tr>
<tr>
<td>29</td>
<td>10861</td>
<td>10848</td>
<td>145</td>
<td>145</td>
<td>48.6</td>
<td>48.6</td>
<td>729.0</td>
<td>729</td>
</tr>
</tbody>
</table>

As may be seen from the above, the proposed model compares favourably with the results of the Dixon Model.

Leaching Model

As in the case of the published leaching models, each model holds for the process conditions under which the parameters were estimated. Hence the data in the (a) and (b) columns are used respectively to determine the appropriate \( k[g(t)] \) values, and then to determine the values in the respective (c) columns. The entries in (b) were estimated via the model (b) in Table 3, and which are an approximation of the data given by Nicol et al., as these were not explicitly given in their paper.

Table 3  Data in columns (a) and (b) used to calculate \( k[g(t)] \), which is used to estimate the ore grades in columns (c).

\[
(a): \frac{dg}{dt}=-0.7(g-0.19)^2 \quad (b): \frac{dg}{dt}=-2(g-0.19)^2 \quad (c): Eq.3&4
\]

<table>
<thead>
<tr>
<th>RESULTS OF</th>
<th>EQUATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>( (a) )</td>
<td>( (c) )</td>
</tr>
<tr>
<td>TIME</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>2.3</td>
</tr>
<tr>
<td>1</td>
<td>1.03</td>
</tr>
<tr>
<td>2</td>
<td>0.705</td>
</tr>
<tr>
<td>4</td>
<td>0.476</td>
</tr>
<tr>
<td>8</td>
<td>0.334</td>
</tr>
<tr>
<td>16</td>
<td>0.256</td>
</tr>
<tr>
<td>23</td>
<td>0.23</td>
</tr>
</tbody>
</table>

The leaching model produces results which are almost in total agreement with the Mintek leaching models².
Continuous CIP Model

The $k[y(t)]$ for this simulation is produced via equations 1 and 2 from the data given by (Nicol et al. p. 40), taking a pulp density of 1460 g/l, 1:1 liquid to solids mass ratio, $[Au]_v=0.8$ ppm in the first stage throughout and 25g/l carbon. These data are summarized in Table 4.

Table 4 Summary of the calculated $k[y(t)]$ values according to equations 1&2, using the data of Nicol et al.²

<table>
<thead>
<tr>
<th>Loading Range [ppm]</th>
<th>Slope</th>
<th>Time [h]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00 to 4000</td>
<td>-3.42</td>
<td>0 to 50</td>
</tr>
<tr>
<td>4000 to 6804</td>
<td>-2.40</td>
<td>50 to 100</td>
</tr>
<tr>
<td>6804 to 8176</td>
<td>-1.17</td>
<td>100 to 150</td>
</tr>
<tr>
<td>8176 to 9168</td>
<td>-0.85</td>
<td>150 to 200</td>
</tr>
<tr>
<td>9168 to 9986</td>
<td>-0.70</td>
<td>200 to 250</td>
</tr>
<tr>
<td>9986 to 10307</td>
<td>-0.27</td>
<td>250 to 300</td>
</tr>
<tr>
<td>10307 to 10512</td>
<td>-0.18</td>
<td>300 to 350</td>
</tr>
<tr>
<td>10512 and larger</td>
<td>-0.00</td>
<td>350 and larger</td>
</tr>
</tbody>
</table>

These values for $k[y(t)]$ may now be used to simulate a five stage carbon-in-pulp system as described by Nicol et al.². (Volume of reactors 400l and other data as in Tables 5&6)

The results of two simulations are summarized in Tables 5 and 6, at two different pulp flow rates with a 48h period between two successive carbon transfers (100% transferred). The entries in columns (a) are measured process data, the entries in columns (b) as simulated by the model suggested by Nicol et al.² and the entries in columns (c) as simulated by the model proposed in this paper.

* This value could be calculated from equation 1 (Nicol et al.)¹ up to a loading of 4000ppm. Corresponding concentration-time data for Fig.1¹ should be available in order to calculate $k[y(t)]$, but since these were not available 0.8 was used as an average concentration over all loadings.
Table 5  Comparison of process data, predictions by Nicol et al\textsuperscript{2} and the model proposed in this paper.

<table>
<thead>
<tr>
<th>STAGE</th>
<th>Solution (g/t)</th>
<th>Loading (g/t)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(a)</td>
<td>(b)</td>
</tr>
<tr>
<td></td>
<td>(g/t)</td>
<td>(g/t)</td>
</tr>
<tr>
<td>Feed</td>
<td>2.26</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1.12</td>
<td>6503</td>
</tr>
<tr>
<td>2</td>
<td>0.39</td>
<td>3693</td>
</tr>
<tr>
<td>3</td>
<td>0.15</td>
<td>1972</td>
</tr>
<tr>
<td>4</td>
<td>0.08</td>
<td>550</td>
</tr>
<tr>
<td>5</td>
<td>0.03</td>
<td>189</td>
</tr>
<tr>
<td>Mc</td>
<td>25g/l</td>
<td>25g/l</td>
</tr>
</tbody>
</table>

Table 6  Comparison of process data, predictions by Nicol et al\textsuperscript{2} and the model proposed in this paper.

<table>
<thead>
<tr>
<th>STAGE</th>
<th>Solution (g/t)</th>
<th>Loading (g/t)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(a)</td>
<td>(b)</td>
</tr>
<tr>
<td></td>
<td>(g/t)</td>
<td>(g/t)</td>
</tr>
<tr>
<td>Feed</td>
<td>2.02</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0.76</td>
<td>3092</td>
</tr>
<tr>
<td>2</td>
<td>0.24</td>
<td>1600</td>
</tr>
<tr>
<td>3</td>
<td>0.039</td>
<td>168</td>
</tr>
<tr>
<td>4</td>
<td>0.015</td>
<td>49</td>
</tr>
<tr>
<td>5</td>
<td>0.002</td>
<td>18</td>
</tr>
<tr>
<td>Mc</td>
<td>25g/l</td>
<td>25g/l</td>
</tr>
</tbody>
</table>

The amount of carbon predicted by the model proposed here i.e. 23.5 and 22g/l respectively, is much closer to the real process value of 25 g/l, than that predicted by Nicol et al\textsuperscript{2} of 16 and 17 g/l respectively.

Continuous CIL Model

Using the same values for $k[y(t)]$ in Table 4 above and the entries in column (b) of Table 3 to produce $k[g(t)]$, it was possible to simulate an eight stage CIL-system, with a three hour retention time in the three leaching reactors. The remaining process variables remained as for the CIP-simulation described above, with the further addition that leaching also takes place in the five adsorption stages.
Table 7 Comparison of predictions produced by Nicol et al.²’s CIL-model and the model proposed in this paper.

<table>
<thead>
<tr>
<th>STAGE</th>
<th>SOLIDS (g/t)</th>
<th>SOLUTION(g/t)</th>
<th>CARBON(g/t)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(a)</td>
<td>(b)</td>
<td>(a)</td>
</tr>
<tr>
<td>FEED</td>
<td>2.3</td>
<td>2.300</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0.6</td>
<td>0.70</td>
<td>1.70</td>
</tr>
<tr>
<td>2</td>
<td>0.4</td>
<td>0.41</td>
<td>1.90</td>
</tr>
<tr>
<td>3</td>
<td>0.38</td>
<td>0.31</td>
<td>2.02</td>
</tr>
<tr>
<td>4</td>
<td>0.25</td>
<td>0.29</td>
<td>0.43</td>
</tr>
<tr>
<td>5</td>
<td>0.25</td>
<td>0.27</td>
<td>0.09</td>
</tr>
<tr>
<td>6</td>
<td>0.24</td>
<td>0.26</td>
<td>0.02</td>
</tr>
<tr>
<td>7</td>
<td>0.23</td>
<td>0.25</td>
<td>0.006</td>
</tr>
<tr>
<td>8</td>
<td>0.23</td>
<td>0.25</td>
<td>0.002</td>
</tr>
</tbody>
</table>

The proposed model predicts a carbon concentration of 23g/L, once again much closer than that produced by the model proposed by Nicol et al.² of 16g/L. There would have been a closer correspondence between the leaching data if appropriate batch data had been supplied by Nicol et al.². From Table 7 it is clear that the correspondence is reasonable.

Database Model

The results of this example are summarized in Table 8, which are an extension to the CIP-example discussed above. If the results in Table 5 are used as a basis [column (a) in Table 8], two additional simulations are given showing the effect of a typical physical parameter e.g. continuous counter-current carbon transfer [column (b): 48h retention time] and a typical chemical process parameter e.g. carbon poisoning [column (c): 50% decrease in the rate k(g(t))] on the performance of the CIP-circuit.

Table 8 The effect of continuous counter-current transfer of the carbon and carbon poisoning on the performance of the circuit.

<table>
<thead>
<tr>
<th>STAGE</th>
<th>Solution (g/t)</th>
<th>Loading (g/t)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(a)</td>
<td>(b)</td>
</tr>
<tr>
<td>Feed</td>
<td>2.020</td>
<td>2.020</td>
</tr>
<tr>
<td>1</td>
<td>0.510</td>
<td>0.510</td>
</tr>
<tr>
<td>2</td>
<td>0.130</td>
<td>0.130</td>
</tr>
<tr>
<td>3</td>
<td>0.032</td>
<td>0.032</td>
</tr>
<tr>
<td>4</td>
<td>0.008</td>
<td>0.008</td>
</tr>
<tr>
<td>5</td>
<td>0.002</td>
<td>0.002</td>
</tr>
</tbody>
</table>

From these results it is clear that continuous counter-current operation differs marginally from the counter-current operation. On the other hand the carbon poisoning causes a marked increase in the concentration of the gold in the stages, hence a large gold loss.
the carbon poisoning causes a marked increase in the concentration of the gold in the stages, hence a large gold loss.

DISCUSSION AND SIGNIFICANCE

The discussion and results above serve to illustrate the following regarding the Knowledge Based System:

- The database can be applied in the simulation of adsorption, leaching and a combination of these systems.
- It is shown how this very simple model can be used successfully to simulate various continuous systems.
- Chemical process variables are included in the model, permitting the simulation of the above-mentioned systems at all process conditions catered for by the database.
- The user has total control over all process variables, physical or chemical, via user-friendly menu interfaces.
- Although not shown here, the model can perform fault-diagnosis and economic modelling.

To summarize: the model proposed here permits the inclusion of all process variables present in leaching and adsorption systems in a totally generalized modelling system.

LIST OF SYMBOLS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C, C_t, C_n</td>
<td>Concentration in a batch reactor, concentration in stage i and data point n respectively [ppm].</td>
</tr>
<tr>
<td>g, g_t, g_n</td>
<td>Ore grade in a batch reactor, grade in stage i and grade data point n respectively [g/ton].</td>
</tr>
<tr>
<td>k</td>
<td>Rate variable as a function of ( y(t) ) and ( g(t) ) ([h^{-1}]).</td>
</tr>
<tr>
<td>( M_c, M_t, M_s )</td>
<td>Mass concentration of carbon, liquid and solids respectively ([g/l]).</td>
</tr>
<tr>
<td>( m^{co}_c )</td>
<td>Mass flow rate of the carbon counter-current and ( m^{co}_c ) for co-current flow ([kg/h]).</td>
</tr>
<tr>
<td>( M^{cc}, M^{ct} )</td>
<td>Mass of carbon being transferred counter-current and ( M^{cc}, M^{ct} ) co-current ([kg]).</td>
</tr>
<tr>
<td>r</td>
<td>Rate equation ([ppm/h]).</td>
</tr>
<tr>
<td>t</td>
<td>Time ([h]).</td>
</tr>
<tr>
<td>( V_{act} )</td>
<td>Active stage volume ([m^3]).</td>
</tr>
</tbody>
</table>
\[ \nu_{\text{in}} \quad \text{Pulp flow in and} \quad \nu_{\text{out}} \quad \text{out of a stage [m}^3/\text{h}. \]

\[ y, y_i, y_n \quad \text{Carbon loading in a batch reactor, loading in stage } i \text{ and loading data point } n \text{ [ppm].} \]

\[ \delta t \quad \text{Time increment.} \]

REFERENCES


XVIIth International Mineral Processing Congress
Dresden/FRG, September 23 - 28, 1991

PREPRINTS
Volume I
Comminution and Classification
Modelling and Process Control
A KNOWLEDGE BASED SYSTEM FOR THE SIMULATION AND OPTIMIZATION OF METALLURGICAL PLANTS

M.A. Reuter* and J.S.J. van Deventer

Department of Metallurgical Engineering, University of Stellenbosch, Stellenbosch, 7600, South Africa.

SUMMARY

This paper illustrates the application of an object-orientated knowledge based system (KBS) to perform fault-diagnosis and dynamic simulation of batch and continuous flotation processes. A two-stage linear programming model could be used in association with the KBS to optimize flotation circuits. The dynamic behaviour of a flotation process is fingerprinted by the gradient of change in the concentration of a species. The KBS is defined by various facts, objects, rules and functions, which capture both deep and shallow knowledge regarding the flotation process. It is evident that the accuracy of prediction is entirely dependent on the accuracy and population density of the data base. The proposed KBS produces realistic simulations of published data for the flotation of sphalerite in rougher and scavenger banks.

INTRODUCTION

Numerous factors influence the flotation of valuable components in a flotation cell [1,2]. Although a number of fundamental models have been proposed for flotation, the parameters of these models usually lump together numerous process variables [1]. Hence, little information regarding the process can be extracted from these parameters. Therefore, it is the objective of this paper to propose a knowledge-based system (KBS) model which would enable the inclusion of all possible flotation variables, e.g. frother & collector addition, pH, aeration rate etc. to render the as yet ill-defined flotation process well-defined. This is accomplished here by using a hybrid KBS approach, i.e. both qualitative (experiential and heuristic) knowledge and quantitative (equation orientated) knowledge are used to describe the kinetics of flotation.

The modelling of various ill-defined chemical engineering problems have been performed by using KBS techniques. Most of these applications are, however, confined to steady-state circuit design [3-5], the selection of equipment, processes or reagents e.g. DECADE for the selection of catalysts [6,7], synthesis of operating procedures [3,8-10], malfunction diagnosis [11-14] and control synthesis [3,15-17]. Little emphasis has been placed to date on the application of these techniques to the modelling of mineral processing systems [18]. Although the dynamic simulation of ill-defined problems by the use of these techniques has found little application in the literature, qualitative simulation has been

* Present address: Institut für Metallhüttenwesen und Elektrometallurgie, RW-Technische Hochschule, Aachen, Germany.
explored in this regard [19]. It has already been shown how a KBS and a dynamic equation orientated system can be combined to simulate CIP and CIL systems [20,21].

In the above-mentioned systems various architectures have been applied to formalize the knowledge e.g. object orientated [3,14], frame orientated [5], hybrid systems in a blackboard [6,7], hierarchical [8,11] and goal-tree-success-tree [17] systems. Of these systems the object orientated approach was used to define a KBS that can embrace both the dynamics of a process as well as do process malfunction diagnosis on the basis of the chemical and physical environment within the reactor. Although this model could be applied to leaching, adsorption or pyrometallurgical processes, it is applied here to simulate the rougher and scavenger banks of a sphalerite flotation process [1]. It is also demonstrated how the KBS system can suggest an optimal flotation circuit by coupling the dynamic equations and the knowledge base to a linear programming model [22].

KNOWLEDGE REPRESENTATION

The representation of shallow (experiential and heuristic) and deep (equation orientated) knowledge for application in knowledge based systems is probably one of the most important aspects of this modelling approach. In the approach discussed here, the batch flotation concentration-time recovery data of the valuable element, as well as all other species involved, form the basis of the knowledge base. These data are transformed into a useful form by a generalized kinetic model, the rate constant also being a function of all the effects of the flotation process variables. The organization of the data in this manner permits the interaction between the batch flotation concentration-time data, process variables, kinetic model and mathematical modelling. This may be realized by knowledge representation techniques such as facts, objects (frames), rules and functions [19]. These techniques are used here in the following manner:

*Facts* may include for example the type of reaction under consideration, be it leaching, adsorption or reduction, flotation or the type of reactor system (batch, CSTR, counter-current etc.),

Each *Object* (Frame) is structured so as to include all relevant information regarding a particular process variable and its effect on the process. An example here is the *pivot-object*, that defines both shallow and deep knowledge regarding a process under consideration (It refers here to the said sphalerite flotation process with the conditions of Experiment 1 [2] forming the pivot-conditions).

<table>
<thead>
<tr>
<th>TYPE OF FLOTATION SYSTEM</th>
<th>: Sphalerite flotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISTINGUISHING CHARACTERISTIC</td>
<td>: Type of ore body</td>
</tr>
<tr>
<td>n FACTOR ADJUSTMENT-OBJECTS</td>
<td>: Particle size, etc.</td>
</tr>
<tr>
<td>k REGRESSION-OBJECTS</td>
<td>: Total %Zn in feed</td>
</tr>
<tr>
<td>Collector addition, etc.</td>
<td>: Time $t$</td>
</tr>
<tr>
<td><strong>KINETIC DATA</strong> (i sets of kinetic data e.g. change of reagents)</td>
<td>: $0.5 \rightarrow 0.946$</td>
</tr>
<tr>
<td></td>
<td>: etc.</td>
</tr>
<tr>
<td><strong>TYPE OF OPERATION</strong></td>
<td>: Batch/Continuous</td>
</tr>
<tr>
<td></td>
<td>: (function-objects &amp; unit-operation-object)</td>
</tr>
<tr>
<td><strong>COST DATA</strong></td>
<td>: Function of operation</td>
</tr>
</tbody>
</table>
This object comprises of various other objects which include amongst others the following unit-operation-object:

- **UNIT OPERATION**: CONTINUOUS
- **REACTION**: 
- **MINERAL CONTENT OF FEED**: 11.8%
- **% SOLIDS IN FEED**: 43.3%
- **RETENTION TIME OF CELLS 1-4**: 11.2 min
- **NUMBER OF REACTORS IN SERIES**: 4
- **FEED FLOWRATE OF THE PULP**: 43.3 t/h
- **% SHORT CIRCUITING OF FEED**: 0
- **% DEAD VOLUME OF REACTOR**: 0

or the adjustment-object comprising of n possible objects with m sub-divisions which define the effects of the deviation of process variables on the pivot process conditions:

- **ADJUSTMENT-OBJECT LABEL**: Particle size
- **SUB-DIVISION OF ADJUSTMENT-OBJECT**: \( d_{50} = 80 \mu m \)
- **n-1 INTERACTION FACTORS**: \( \alpha = 1 \)
- **SLOPE FACTOR**: \( \alpha = \beta = 1 \)
- **FINAL RECOVERY FACTOR**: \( \alpha = \beta = 1 \)
- **COST FACTOR**: \( \alpha = \beta = 1 \)

or the k regression-objects (similar to the object above) that define the effects of process variables via various types of regression equations (logarithmic, polynomial & power) e.g.:

- **ADJUSTMENT-OBJECT LABEL**: Total %Zn in feed
- **k-1 INTERACTION FACTORS**: \( \alpha = 1 \)
- **SLOPE REGRESSION EQUATION**: \( 0.0588(\%) + 0.33 = \alpha \)
- **FINAL RECOVERY REGRESSION EQUATION**: \( 1 = \beta \)
- **COST REGRESSION EQUATION**: \( 1 = \beta \)

This regression rule (taken from data of Frew & Davey [2]) would imply that for a feed grade of 11.5%, \( \alpha \) would be unity (no deviation from pivot), hence corresponding to the pivot-rate which is 0.11 min\(^{-1}\) or the corresponding pivot \( k[C^2(t)]_n \) values (see below).

The Rules (production rules) defined here fall into various categories which include amongst others the following adjustment and search rules respectively:

```
if Sphalerite flotation and Type of ore body and Zn content of feed = 17.9% and \( d_{50} = 80 \mu m \) and etc.
then slope factor = \( \alpha = 1 \times 1.38 = 1.38 \) and load factor = \( \beta = 1 \)
if Zn recovery is \geq 64% and Zn recovery is \leq 66% and Cells 1-4 of rougher bank then Zn content of feed = 11.8% and \( d_{50} = 80 \mu m \) etc.
```

The user has control over the search accuracy of this search rule.

*Functions* include the kinetic models, differential equations that define the non-ideal flow through the continuous reactors or equations that describe fundamental aspects of the process. A function would hence contain the following elements:

\[ \text{...} \]

411
TYPE OF OPERATION (defining label)

- Kinetic model (differential equation: see below)
- Unit-operation-object (as above)
- Non-ideal flow model (differential equation: see below)
- Regression adjustment-objects (as adjustment-object above with factor replaced by a regression equation)
- Linear programming optimization model
- Fundamental correlations (e.g. \( Sh = a(Re)^b(Sc)^c + d \))

This knowledge representation is integrated in a user-friendly simulation programme which is written in Turbo Pascal. Figure 1 depicts the integration of these data.

**Figure 1** Overall structure of KBS.
DEVELOPMENT OF THE KNOWLEDGE BASED MODEL

As has been mentioned above, a generalized kinetic model has been developed which may be used to describe the kinetics of leaching, adsorption and pyrometallurgical reduction reactions. In this section it will be shown how this model can be applied in conjunction with a knowledge base to describe batch flotation kinetics and the flotation in continuous cells. Furthermore, it will be shown how optimal flotation circuits can be determined and fault-diagnosis can be performed by using this kinetic model.

Knowledge Based Kinetic Model

As has been defined for leaching, adsorption and pyrometallurgical reactions [20,21], the rate of depletion of the valuable element in the slurry of a batch cell has been defined by a non-linear first order rate equation, in which the rate variable \( k[C^e(t)] \) is only a function of \( C^e(t) \), the concentration of element \( e \) (e.g. galena, water, gangue, etc.). No flotation species (e.g. fast and slow floating species within element \( e \)) are catered for in this model, as these are implicitly included in the rate variable \( k[C^e(t)] \) and the accompanying knowledge base, which cover all possible particle size distributions, cell operating conditions and mineral compositions. The value of \( k[C^e(t)] \) is determined as given below from a flotation curve, which covers the flotation up to the final recovery \( R \).

\[
\frac{dC^e}{dt} = -k[C^e(t)] C^e(t) \tag{1}
\]

where: \( k[C^e(t)]_n = -\frac{2(C^e_{n+1} - C^e_n)}{St(C^e_{n+1} + C^e_n)} \)

The value of \( k[C^e(t)] \) corresponds to the pivot process variables. Any deviation from these pivot-data is reflected by the \( a \) and \( b \) values of the adjustment objects. If these values deviate from 1, the following equation may be applied to predict the kinetics of the flotation process, were \( k[C^e(t)]_n \) is as determined by equation 1 and the discrete ranges of \( C^e(t)_n \) scaled linearly to fit between 1 and \( (1-R) \) [\( R = \) final recovery for eq. 1]:

\[
\frac{dC^e}{dt} = -a \cdot [k[C^e(t)] C^e(t) - (1-b)] = r \tag{2}
\]

Continuous Flow Model

Ideal flow is assumed in the formulation of the continuous flow model for the depletion of an element \( e \) in cell \( i \), although in the KBS non-ideal flow in the form of short-circuiting \([C^e_{i-1} = \psi C^e_{i-1} + (1-\psi)C^e_{i+1}] \) and dead volume \([V_{act} = V_{total} - V_{dead}] \) is incorporated. The theoretical values of the dead volume and bypass streams can be obtained from tracer tests, or can be estimated by comparing real plant behaviour with that predicted from an ideal flow model.

\[
\frac{dC^e_i}{dt} = \frac{V_{p,in}}{V_{act}} C^e_{i-1} - \frac{V_{p,out}}{V_{act}} C^e_i + r \tag{3}
\]
Linear Programming Optimization Model

The constraints which define this model have been developed by Reuter and Van Deventer [22] and the reader is hence referred to this reference for further details. Note, however, that the linear programming model that is defined here is a simplification of that defined in [22], as no flotation species are defined. Of all the constraints in the linear programming model that are of importance, the one that defines the separation of element e at bank j and links the KBS to the optimization routine, is repeated here:

\[(y_{mcj})^L m_{cj} \leq y_{cj} \leq (y_{mcj})^U m_{cj}\]  \hspace{1cm} (4)

Values for \((y_{mcj})^L\) and \((y_{mcj})^U\) are determined as follows: (i) the expected operating conditions in a bank of cells are selected, which produce an adjusted kinetic model (eq.2), (ii) this can subsequently be used to simulate a bank of continuously operated flotation cells (eq.3) and finally from this a separation factor for each element e can be determined.

Process Identification and Fault-Diagnosis

The definition of the rate variable \(k[C^e(t)]\) in the manner as described by eq. 1 is very useful to perform fault diagnosis using kinetic data only. Consider a first order reaction and the following time-weighted slope integral:

\[
\int_0^\infty t kC dt = \int_0^\infty t ke^{-kt} dt = 1/k
\]  \hspace{1cm} (5)

If this value for \(1/k\) is compared to that of another curve with rate constant \(k'\), the ratio \(k'/k\) gives a value for the adjustment factor \(\alpha\), which can subsequently be used in conjunction with \(\beta\) to search through all adjustment-rules to determine what the chemical environment in a batch flotation cell is. This has been generalized for the rate variable \(k[C^e(t)]_n\) in the following manner:

\[
1/k^n = \sum_n t_n k[C^e(t)]_n C^e(t)_n \delta t
\]  \hspace{1cm} (6)

For the process identification in a continuous bank of cells, the \(\alpha\) and \(\beta\) values are adjusted continuously, until the recovery corresponds to that obtained from the plant. The \(\alpha\) and \(\beta\) values are an indication of the deviation from the data in the pivot-object which subsequently points to the appropriate adjustment-objects, which then suggest the appropriate operating conditions.

SIMULATION RESULTS

Various examples will be given in the subsequent section to illustrate the application of the developed models.

Batch Flotation

The generality of the proposed kinetic model [eqs.1 & 2] is illustrated by comparing its performance to that of two well known kinetic models. A number of values for \(k[C^e(t)]_n\)
for the concentration-time data in Tables 2 and 3 are summarized in Table 1. These values are subsequently used to produce the results in the (b) columns of Tables 2 and 3.

It is clear from the results summarized in Tables 2 and 3 that one model can predict the same results as two different well known theoretical models. It may be mentioned here that the same model has also been applied to adsorption, leaching and pyrometallurgical reactions, with results comparable to flotation above [20,21].

Table 1 Summary of the calculated \( k[Ce(t)]_n \) values according to equation 1, using the data in column 1(a) in Table I.

<table>
<thead>
<tr>
<th>Column 1(a) of Table 2</th>
<th>Column 1(a) of Table 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Ce = 1 - R[1 - (1 - e^{-kt})/kt] )</td>
<td>( Ce = 1 - R[1 - e^{-kt}] )</td>
</tr>
<tr>
<td>( Ce ) Range</td>
<td>( k[Ce(t)] )</td>
</tr>
<tr>
<td>1.00 to 0.95</td>
<td>-0.496</td>
</tr>
<tr>
<td>0.95 to 0.90</td>
<td>-0.487</td>
</tr>
<tr>
<td>0.90 to 0.86</td>
<td>-0.476</td>
</tr>
<tr>
<td>etc.</td>
<td>etc.</td>
</tr>
</tbody>
</table>
| ... to 0.00 | 0.00 | ... to 0.00 | 0.00 | ...

Table 2 Columns (a): Concentration-time data from the given model and \( R \) and \( k \) values. Data in column 1(a) were used to determine \( k[Ce(t)] \) by eq. 1, which was subsequently used to predict values in columns (b) according to eq. 2.

<table>
<thead>
<tr>
<th>(a): ( C/Co = 1 - R[1 - (1 - e^{-kt})/kt] )</th>
<th>(b): Eq. 1 &amp; 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R ) &amp; ( k )</td>
<td>( 1.0 ) &amp; ( 1.0 / \text{min} )</td>
</tr>
<tr>
<td>( \beta ) &amp; ( \alpha )</td>
<td>1.0 &amp; 1.0</td>
</tr>
<tr>
<td>( t[\text{min}] )</td>
<td>1(a)</td>
</tr>
<tr>
<td>0.0</td>
<td>1.000</td>
</tr>
<tr>
<td>0.5</td>
<td>0.787</td>
</tr>
<tr>
<td>1.0</td>
<td>0.632</td>
</tr>
<tr>
<td>2.0</td>
<td>0.432</td>
</tr>
<tr>
<td>4.0</td>
<td>0.245</td>
</tr>
<tr>
<td>6.0</td>
<td>0.166</td>
</tr>
</tbody>
</table>

Continuous Flotation

To illustrate the application of this proposed KBS to a practical system, the sphalerite flotation data of Frew and Davey [2] were used. For this purpose, the various data objects are defined as those given above. Although not shown here, objects which define the effect of reagent additions, feed grades and particles sizes on the flotation rate constant, must also be included. Frew and Davey [2] described flotation by simple first order kinetics. From their data it is clear that the rate constant \( k[Ce(t)]_n \) has a constant value of 0.11 min\(^{-1}\) for the pivot-object defined above for cells 1-4. Assuming ideal flow, using the appropriate unit-operation-object, applying eqs. 1-3 and the given retention times, the results summarized for cells 1-4 in Table 4 could be produced for Experiment 1 [2].

415
Table 3: Concentration-time data from the given model and \( R \) and \( k \) values. Data in column 1(a) were used to determine \( k[C(t)] \) by eq. 1, which was subsequently used to predict values in columns (b) according to eq. 2.

\[
\begin{array}{cccccc}
R & k & 1.0 & 1.0/min & 0.5 & 0.1/min & 0.1 & 4.0/min \\
\beta & a & 1.0 & 1.0 & 0.5 & 0.1 & 0.1 & 4.0 \\
\hline
\hline
t(min) & 1(a) & 1(b) & 2(a) & 2(b) & 3(a) & 3(b) \\
\hline
0.0 & 1.000 & 1.000 & 1.000 & 1.000 & 1.000 & 1.000 \\
0.5 & 0.607 & 0.607 & 0.976 & 0.976 & 0.914 & 0.914 \\
1.0 & 0.368 & 0.368 & 0.952 & 0.952 & 0.902 & 0.902 \\
2.0 & 0.135 & 0.136 & 0.909 & 0.909 & 0.900 & 0.900 \\
4.0 & 0.018 & 0.018 & 0.835 & 0.835 & 0.900 & 0.900 \\
\end{array}
\]

Table 4: Comparison of the practical data for Experiment 1 (rougther & scavenger) of Frew and Davey [2] with those produced by the proposed KBS.

<table>
<thead>
<tr>
<th>Cell No</th>
<th>Plant ZnS</th>
<th>KBS ZnS</th>
<th>Plant Gangue</th>
<th>KBS Gangue</th>
<th>Plant Water</th>
<th>KBS Water</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.75</td>
<td>0.76</td>
<td>0.995</td>
<td>0.995</td>
<td>0.96</td>
<td>0.98</td>
</tr>
<tr>
<td>2</td>
<td>0.57</td>
<td>0.58</td>
<td>0.99</td>
<td>0.99</td>
<td>0.94</td>
<td>0.96</td>
</tr>
<tr>
<td>4</td>
<td>0.36</td>
<td>0.34</td>
<td>0.985</td>
<td>0.985</td>
<td>0.92</td>
<td>0.92</td>
</tr>
<tr>
<td>7</td>
<td>0.076</td>
<td>0.077</td>
<td>0.98</td>
<td>0.96</td>
<td>0.88</td>
<td>0.88</td>
</tr>
<tr>
<td>10</td>
<td>0.016</td>
<td>0.016</td>
<td>0.97</td>
<td>0.95</td>
<td>0.85</td>
<td>0.84</td>
</tr>
</tbody>
</table>

From the simulation results for cells 1-4 and the process conditions within cells 5-7, appropriate adjustment-objects would suggest that the pivot-rate \( k[C(t)] \) should be adjusted by a factor of 2. Subsequently, these cells as well as cells 8-10 could be simulated by following the same method. From the above it is clear that the comparison is excellent.

A similar approach must be followed for all other components involved, e.g. in the above example galena, pyrrhotite, non-sulphide gangue and water. For each of these an associated adjustment-object must be defined. For example, for the gangue under the prevailing conditions, the rate of water recovery is a factor \( \alpha = 0.055 \) of that of the sphalerite in cells 1-4, 0.027 in cells 5-10 and for the gangue \( \alpha = 0.012 \) for all cells.

**Optimization of a Flotation Circuit**

Once the various simulations have been performed for various flotation scenarios and recycle possibilities in order to determine the appropriate separation factors, a linear programming model for the valuable element, in this case sphalerite, could be developed according to [22]. From Table 1 the separation factor for the sphalerite in the rougher and scavenger banks respectively could be determined to produce the following two constraints:

\[
0 \ m_{ZnS1} \leq y_{ZnS1} \leq 11.98 \ m_{ZnS1} \\
0 \ m_{ZnS2} \leq y_{ZnS2} \leq 3.8 \ m_{ZnS2}
\]
The constraint for the cleaners, which were grouped here into one bank, could be estimated from the given data [2] to be (kinetic data not given for these banks):

\[ 0 \leq m_{ZnS} \leq 90 \quad \text{m}_{ZnS} \]

The result of this simulation is the depicted circuit structure (fig. 2) for a maximization of the sphalerite recovery. The recovery for this structure is 98.4%. However, if the concentrate recycle of the scavenger is forced to report to the rougher, i.e. the structure as given by [2], the recovery drops to 98.2%. With a view to determine the grade, a similar model can be developed for the other elements in order to minimize their flow for the given optimal structure for the sphalerite (see [22]).

**Figure 2** Circuit structure and sphalerite flow rates in t/h for optimal circuit as produced by the KBS (flow rates t/h)

![Diagram](image)

From the above flow rates it is clear that they do not differ much from those given in [2] for Experiment 1, despite the difference in structure.

**Fault-Diagnosis**

In order to illustrate the identification of a process on the basis of its batch flotation curve, eq. 5 is applied to the pivot-data of the flotation example [2] for which \( k = 0.11 \text{ min}^{-1} \). The integral will produce a value of 9.09. If another curve is compared to this, say with a rate constant \( k = 0.16 \text{ min}^{-1} \), the integral produces a value of 6.25, hence \( \alpha = 9.09/6.25 = 1.45 \). The search routine will attempt to produce a match for this \( \alpha \) and suggest appropriate operating conditions.

If the search rule for cells 1-4 is considered, it is clear that the result for this search would be the pivot-conditions as given. However, if the search ranges were to change to those given below, the result would have pointed to an increased Zn feed grade of 17.9%:

- *and* Zn recovery is \( \geq 72\% \)
- *and* Zn recovery is \( \leq 73\% \)
- *and* Cells 1-4 of rougher bank
- *then* Zn content of feed = 17.9%
  - \( d_{50} = 80 \mu m \)
  - \( \alpha = 1.38 \)
  - \( \beta = 1 \)
  - etc.

From the simulation of Experiments 2 and 3 for cells 1-4 [2], it would have become clear via the Total %Zn in feed-object that the total Zn feed (fresh & recycle) to the rougher has to increase so as to produce the measured recovery.
DISCUSSION OF RESULTS

The simulation of batch and continuous flotation by use of an object-orientated KBS, was discussed above. Furthermore, it was shown how the data of these simulations could be used to optimize a flotation plant and perform fault diagnosis.

The results of these simulations serve to illustrate the following regarding this simulation approach:

- It was shown how a generalized non-linear first order rate equation can be used to approximate two different well-known theoretical flotation models. Although not shown here, it has been demonstrated that this same equation could be applied to accurately simulate leaching, adsorption and pyrometallurgical reduction reactions.

- This form of the rate equation permits the definition of a generalized knowledge base which can interact with the rate variable and the final recovery via objects. This permits the definition of a well-defined kinetic model for any type of reaction under consideration, be it leaching, adsorption, reduction or flotation.

- The application of this kinetic model was used to simulate the recovery of sphalerite, gangue and water in continuous rougher and scavenger banks by taking the process conditions within a cell into consideration. The results in this case were very good.

- Fault-diagnosis/process identification can also be performed by utilizing the adjustment information \( \alpha \) and \( \beta \) of the generalized kinetic model. By comparing this information to the pivot-data, the model can suggest the possible conditions within the reactor under consideration.

- It was demonstrated how the simulation and process identification activities could be integrated with a linear programming optimization model. A circuit was produced which gave nearly the same results as the industrial flotation plant, with the difference being that the concentrate of the scavenger reports to the cleaner and not to the rougher. If this stream is forced to report to the rougher, the recovery drops by 0.2%.

From the results of the KBS, it is clear that the KBS could render an ill-defined process well-defined, and it is able to perform simulations that standard simulation programmes could not perform.

SYMBOLS USED

\[
\begin{align*}
C_e, C_{e_i}, C_{e_n} & \quad \text{Concentration in a batch reactor, concentration in stage i and data point n of element e respectively.} \\
k[C^e(t)] & \quad \text{Rate variable as a function of } C^e(t) \text{ / min}^{-1} \\
m_{ej}, y_{ej} & \quad \text{Mass flow rates of tailings and concentrate of element e at bank j respectively / t h}^{-1} \\
r & \quad \text{Rate of flotation / min}^{-1}
\end{align*}
\]
REFERENCES


[17] Birkey, G.J., T.J. McAvoy and M. Modares: An expert system for distillation control design

Miner. Engng. 2(1989)1, pp. 87-92


Proceedings: APCOM '90, Berlin, September 1990

[22] Reuter, M.A. and J.S.J. van Deventer: The use of linear programming in the optimal design of flotation circuits incorporating regrind mills
Int. J. Min. Proc. 28(1990)1, pp. 15-43
PERSONAL COMPUTERS AND THE PROCESS INDUSTRY

24 and 25 April 1989
CSIR Conference Centre

SOUTH AFRICAN INSTITUTION OF CHEMICAL ENGINEERS (N-TVL BRANCH)
A KNOWLEDGE BASED SYSTEM FOR THE SIMULATION OF METALLURGICAL PLANTS

by

M.A. Reuter and J.S.J. van Deventer

Department of Metallurgical Engineering
University of Stellenbosch
Stellenbosch
7600, South Africa.

SYNOPSIS

The application of a knowledge based model for the simulation of metallurgical plants is discussed. Other than most knowledge based systems, the model permits the dynamic simulation of metallurgical unit operations. Hence, it will be able to control metallurgical plants in a dynamic supervisory capacity.

As is the case with most knowledge based systems, the model consists of a data base and a set of rules etc. to infer data from the data bank. This heuristic data and concentration-time data are then used to simulate carbon-in-pulp batch, counter-current, fixed bed column and moving bed column reactors respectively. Results produced by the model are given and compared to experimental data. It is also possible to infer operating conditions given concentration-time data.

A discussion of the Turbo Pascal programme, that is currently still being developed by the author, is given with reference to the various user interfaces produced by the programme, which include menus, graphs etc.
INTRODUCTION

In the past number of years Artificial Intelligence concepts have found increased application in various Chemical Engineering disciplines. A large portion of these applications focus on fault diagnosis of chemical processes and selection of equipment (Bunn et al.). They discuss various applications of expert systems in chemical engineering, which include topics such as emergency isolation, control of flare systems, pressure relief systems and valve sequencing. In a specific application, Shiozaki et al. uses sign directed graphs to explain the faults which are observed in a chemical system. In another application, Rich et al. used Artificial Intelligence and Knowledge-Based Expert System methodologies to develop a prototype process independent expert system, MODEX, which can be used in the automated diagnosis of faults in chemical process plants.

In a paper by Stephanopolous, the use and future of expert systems in Chemical Engineering is discussed. A section of the article is devoted to the discussion of the Design Kit which is currently being developed at the Laboratory for Intelligent Systems for Process Engineering. This programme is used to develop process flow sheets and the accompanying control-loop structure of complete plants. Not only does the Design Kit focus on equation orientated simulation, but also on reasoning capabilities, which utilizes heuristic knowledge and other qualitative information to help the engineer to justify his selection of equipment, layout etc..

Norman et al. discuss the control of a grinding mill by means of an expert system. Whereas most normal supervisory control systems make use of state space models, the output of this system is limited to suggesting appropriate courses of action, depending on the level of plant operation. A similar approach is followed by Inomata et al. to control reactors.

From the above discussion it would be clear that the principal component in an expert system is its knowledge base and the model that infers data from it. A typical expert system would then comprise the following components, viz.

- a knowledge base;
- an inference procedure for utilizing the data in the knowledge base;
- a working memory which contains the current input and status of a specific problem being solved, and;
- a user-friendly interface used to communicate with the above three components.

However, none of these mentioned expert systems include a dynamic state space process model, which permits the dynamic control of plants. It is shown in this paper how a general kinetic model is incorporated in a knowledge based system to simulate the dynamic behaviour of various continuous carbon-in-pulp systems.

The various elements of the knowledge based system (still under development) will be discussed under the headings:

- The overall structure;
- Knowledge Based System (KBS), and;
- Mathematical Modelling System (MMS).

Each of these headings will be discussed by referring to the user interfaces produced by the programme.

THE OVERALL STRUCTURE

The programme is controlled via a main menu from which either the KBS is run or the MMS. These two systems communicate via files. Initially the KBS must be run before the MMS, since appropriate kinetic data must be available before the MMS can operate. Figure 1 depicts the main menu.
The main pop-up menu for the System.

The selection of these items lead to menus which control the KBS and the MMS respectively.

**THE KNOWLEDGE BASED SYSTEM**

This system is introduced by a menu depicted by figure 2.

**Fig. 2** The main menu of the KBS.

From this menu two courses may be followed viz. i) the selection of concentration-time data for a specific carbon-slurry system subject to heuristic information regarding the process and reaction, and ii) secondly, the inference of operating conditions, given concentration-time adsorption data for a specific carbon.

The first option of the this menu firstly produces the carbon-type menu and subsequently the various data available for the adsorption in various slurry types. This is depicted by figure 3.
The selection of carbon-type and sub-type.

Once these concentration-time data have been selected, the appropriate heuristic data for the selected slurry-carbon system is loaded. A menu gives access to these heuristic data, which makes it possible to conduct the following activities:

- editing the data, editing menu entries and the heuristic knowledge;
- edit the interaction rules existing between the different data;
- edit the heuristic rules which effect the concentration-time data, and;
- adjust the basic concentration-time data using the heuristic information regarding the reaction.

These options are all reflected by figure 4.

Exiting from this menu-option results in the saving of all relevant data which is required by the MMS.

The second menu-option of the KBS leads to data inference section, where possible operating conditions may be determined, given concentration data for adsorption in a particular slurry-carbon system. This option is illustrated by figure 5.
Determining the operating conditions given concentration-time data for a carbon-slurry system.

The MMS system, which runs subservient to the KBS, will be discussed in the next section.

MATHEMATICAL MODELLING SYSTEM

After having run the KBS and subsequently returning to the main-menu, the option that leads to the KBS may be selected. This choice leads to the main menu of the MMS and is depicted by figure 6.

Fig. 6 The main pop-up menu for the MMS.

Each of these menu-options will not be discussed in detail. The modelling options will be discussed by referring to various simulation results.

KINETIC MODEL/BATCH REACTOR

Various carbon-in-pulp kinetic adsorption models are discussed in the literature e.g., semi-empirical\[^9\], adsorption-desorption\[^10\] and branched-pore\[^11,12\] models. A number of these models have been employed to simulate industrial plants e.g. Menne\[^13\], Nicol et al.\[^14\] (counter current cascades) and Jansen van Rensburg and Van Deventer\[^15,16\] (moving bed and column reactors). Most of these models, however, have a relatively large number of parameters that have to be estimated in order to fit the data. This makes their application cumbersome. As a consequence, a single-parameter kinetic model for the adsorption of gold and silver cyanides on activated carbon was proposed\[^17\]. Contrary to the above-mentioned models, the single parameter is not estimated, but calculated directly from batch concentration-time data by simple algebra. Since concentration-time data in a data bank form the basis from which all predictions originate, the proposed model is termed a **KNOWLEDGE BASED KINETIC MODEL**. Refer to Reuter et al.\[^17\].
A typical result produced by this kinetic model, having selected the batch reactor option in the main menu of the MMS, is depicted in figures 7 and 8.

**Fig. 7**
Menu for the entry of the basic operating conditions.

**Fig. 8**
Concentration-time curve produced by the MMS in a batch reactor.

It is clear from figure 8 that the simulated data closely resembles the experimental data.

**CONTINUOUS REACTOR SYSTEMS**

From the main menu of the MMS (figure 6), it is clear that continuous counter-current stirred tank reactors, packed bed adsorption columns and moving bed adsorption columns can be simulated. As for the batch reactor, the basic operating conditions may be selected via an appropriate input menu. These continuous simulation results are given by figures 9 to 11. Although these results are produced by the MMS on the graphics screen as is the case for the batch reactor, the results have been plotted in order to compare them to experimental results.

Another output of the MMS is the loading distribution with time in the reactor systems. This output is also produced on the graphics screen. Figure 12 depicts the loading distribution within a moving bed column reactor.
Fig. 9

Comparison of the results produced by the MMS and experimental results in a counter-current system.

Fig. 10

Comparison of the results produced by the MMS and experimental results in a packed column adsorption reactor.

Fig. 11

Comparison of the results produced by the MMS and experimental results in a moving bed column reactor.
Fig. 12 The loading distribution in a moving bed column adsorption reactor.

From the results it is clear that the system is capable of simulating experimental continuous carbon-in-pulp systems reasonably accurately.

**OTHER ACTIVITIES**

From the main menu it can be seen various other options are also possible. These include the editing of the batch concentration data, importing data from other sources e.g. dBASEIII, Lotus 1-2-3 and user files, looking at the spline data produced during the course of the simulation and viewing the configuration of the reactor systems.

Particular details of each simulation and of the kinetic model may be found in the paper by Reuter *et al.*

**CONCLUSIONS**

From the above discussion it is clear that the system simulates various continuous carbon-in-pulp adsorption systems reasonably well.

A well defined menu system makes its operation relatively simple.

The fact that the system can include heuristic knowledge e.g. oxygen level, carbon blinding, ionic strength, carbon age etc. during the simulation of continuous carbon-in-pulp systems, makes it superior to other systems that implement kinetic models that can only accommodate physical process parameters.

**REFERENCES**


# Workshop on Expert Systems in Mineral and Metal Processing

**Time and Location:**
26 - 28 August 1991

Helsinki University of Technology
Otaniemi, Espoo
Finland

**Organised by**
The Finnish Society of Automation
Asemäki 12 C
SF-00520 Helsinki
Finland

**Phone:** 358 0 1590 211
**Fax:** 358 0 1590 210

## Session 1: Grinding and Separation

## Session 2: Iron- and Steelmaking

## Session 3: Continuous Casting and Rolling

## Session 4: General Applications

## Session 5: New Methods

## Invited Lectures
KNOWLEDGE BASED SIMULATION AND IDENTIFICATION OF METALLURGICAL REACTORS

M.A. Reuter* and J.S.J. van Deventer
Department of Metallurgical Engineering, University of Stellenbosch, 7600 Stellenbosch, South Africa

*Institut für Metallhüttenwesen und Elektrometallurgie, RWTH Aachen, Integrastraße 3, 5100 Aachen, Germany (present address)

Abstract. This paper illustrates the application of a knowledge based system to perform fault-diagnosis and dynamic simulation of batch and continuous metallurgical and mineral processing systems. The proposed approach is essentially useful to simulate ill-defined dynamic processes where existing fundamental and empirical models fall owing to their lack of generality. The basis of the model is a set \( \{ n(t); P(t), C(t) \} \), which is derived directly from kinetic data or concentration time data of a system of continuous stages in series and its associated pivot knowledge. The generalized first order kinetic model is defined by use of the above set and linked to a knowledge base via adjustment objects containing \( \alpha \) (rate) and \( \beta \) (final recovery) adjustment factors. This approach permits the dynamic simulation of a variety of metallurgical and mineral processing systems at any process conditions catered for by the knowledge base. The same set may be used to establish deviations from the pivot conditions quantified as \( \alpha \) and \( \beta \) values, which hence permits the establishment of process conditions within the metallurgical reactor. In contrast with most existing models, no curve-fitting is required, as the kinetic model utilizes experimental data directly, so that no fitting variances are associated with the data. Also, adjustment factors are defined in an intelligent way in the knowledge-based system so as to relate the operating conditions to the kinetics of the process under consideration.

Keywords. Kinetics; batch and continuous simulation; reactor identification; knowledge based; neural networks.

INTRODUCTION

Schumann (1942) explained that the kinetic behaviour of minerals during flotation will in many cases supply worthwhile information regarding the process and hence the effect of the various process conditions. He also demonstrated that changes of process conditions within flotation cells have a measurable effect on the flotation response. These observations based on the kinetics of the process form the basis of the simulation and model based diagnostic procedure discussed in this paper.

Numerous models exist to describe the kinetics of leaching, flotation, reduction and other processes (Soehn and Wadsworth, 1979; Dowling, Klimpel and Aplan, 1986). Taking flotation as an example, Dowling, Klimpel and Aplan (1986) could show that the kinetic flotation model based on a first order process with a rectangular distribution was superior to thirteen other models. However, by fitting this model to the given data, the fitting procedure produced a fitted final recovery of >100%. Other flotation models performed badly due to large confidence limits on the fitted parameters. Models for leaching, precipitation, reduction and adsorption would exhibit a similar behaviour to that of the flotation models when fitted to kinetic data, unless, however, the kinetic model is an exact representation of the process being considered. This is usually not the case owing to some mixed kinetic mechanism underlying the process.

The large number of models that are available for defining the kinetics of the above-mentioned processes contain a varying number of parameters. When fitted to kinetic data, these parameters are determined only within a certain confidence limit by fitting procedures. Furthermore, each parameter usually lumps various process conditions, making a generalized characterization of the process via these inexact parameters even more difficult. It is clear from the above points that a generalized simulation and diagnostic procedure cannot be defined on the basis of these diverse and inexact models.

It is hence the objective of this paper to develop a generalized kinetic model with the following characteristics: (i) it must fit the data exactly (no variances), irrespective of the process type, (ii) the parameters of the model must be a function of all the possible process conditions (e.g. via objects) catered for by the knowledge base, (iii) the proposed model and its accompanying knowledge base must be simple (≤2 parameters) and must have an analytic solution for batch as well as continuous reactor systems, and (iv) it must facilitate diagnosis of the process conditions within the respective reactors via its one or two parameter(s).

Various knowledge based systems have been developed to perform simulation and diagnostic activities in chemical and metallurgical systems (e.g. Stephanopoulos and others, 1987; Fusillo and Powers, 1988; Hoskins and Himmelblau, 1988; Tzouanas and others, 1986; Tucker and Lewis, 1986). Various architectures have been applied in the above-mentioned and various other examples not mentioned here. These include object orientated (e.g. Venkatsubramanian and Rich, 1988), frame orientated (e.g. Beltramani and Hotard, 1988), goal-tree-success-tree (e.g. Bickley, McAvoy and Modares, 1986) and other architectures. The dynamic simulation of ill-defined problems by the use of these techniques has, however, found little application in the literature. Dalle Molle, Kupers and Edgar (1988), and Kupers (1984) proposed the use qualitative simulation for this purpose.
Reuter and Van Deventer (1991a, 1991b) demonstrated how a knowledge-based system (KBS) can be combined with a system of dynamic equations to simulate leaching, pyrometallurgical reduction, resin adsorption, carbon-in-pulp and carbon-in-leach systems. The methodology applied is based on independent adjustment objects in an hierarchical network using backtracking chaining for its solution as for e.g. in EMYCIN (Shortliffe and Buchanan, 1975). No diagnostic system has really attempted to address diagnostics in metallurgical reactors on a model basis that includes all possible process conditions and a particular unit operation via its parameters. Model based diagnosis, as it is suggested here, has been reported for the diagnosis of electronic circuits (Davies, 1984).

In contrast to the previous papers by Reuter and Van Deventer (1991a, 1991b), this paper will demonstrate how a simple analytic model may be applied to simulate and diagnose batch and continuous metallurgical reactors by applying a suitable knowledge base. The incorporation of a neural network in this approach is also discussed.

**KNOWLEDGE BASED MODEL**

The simplest kinetic expression is a first order kinetic model. Due to its two possible parameters k and the final recovery R it would not be capable of describing many processes very accurately. However, if k is not a constant but a variable e.g. k(C) it has been shown (Reuter and Van Deventer, 1991a&b) that this kinetic expression can describe a variety of metallurgical processes.

\[
\frac{dC(t)}{dt} = -k(C(t))n C(t)
\]

This definition implies that if the process is first order in nature f(C) will be equal to k for a second order process f(C) = kC, a linear function of concentration. If the process is characterized by mixed kinetics f(C) is a complex function of C. This proposal would suggest that k=f(C) is reaction specific and a "finger-print" of a particular reaction, and could hence be used as a basis for diagnosis and diagnostic activities. Note that this model is essentially a one-parameter model, as k(C(t)) also contains information regarding the final recovery i.e. where k(C(t)) becomes zero. It is hence essential that k(C(t)) must cover the process over the full extent of the reaction.

As the approach discussed here attempts to define a methodology not based on statistics, the definition of k(C(t)) should not be based on a function of various parameters to be fitted. In view of this, k(C(t)) was defined as a discrete function of C(t), which is a discrete set of average values for k(C(t)) and their associated validity ranges, calculated directly from the practical kinetic data via 1 in a discretized form:

\[
k(C(t)) = \frac{-dC(t)}{dt} C(t)
\]

\[
C(t)_{av} = \frac{1}{t} \int_{t_0}^{t} C(t) \, dt
\]

C(t)_{av} is taken to be the arithmetic average between two successive data points separated by the time interval \( \Delta t \). C(t)_{av} could also be taken as the geometric mean, but most examples have, however, been done by applying eq. 2. Hence, for discrete data points, eq. 2 becomes for the arithmetic mean (no curve fitting):

\[
k(C(t)) = \frac{-1}{\Delta t} \frac{C(t_{n+1}) - C(t_n)}{C(t_{n+1}) C(t_n)}
\]

hence

\[
\frac{dC(t)}{dt} = -k(C(t))n C(t)
\]

where k(C(t))n holds for the interval \( C(t_n) \leq C(t) \leq C(t_{n+1}) \). The process conditions for which this set \( k(C(t))n \) are determined, are termed the pivot conditions and it is postulated that this set remains independent of the process conditions, i.e. the kinetic mechanism does not change. This definition permits the simulation of processes in which the kinetic data are not a monotonic decreasing function in time i.e. it could be S-shaped for example. An example is the induction period in the concentration retention time data for the Tennessee copper rougher circuit (Dowling, Klimpel and Aplan, 1986), which cannot really be defined by the available theoretical models. It is exactly this induction period which could have a decisive influence during diagnosis.

Equation 4 has an analytic solution for the given range:

\[
C(t) = C(t_0) \exp(-k(C(t))n(t-t_0))
\]

To summarize: the set \( k(C(t))n \) is the basis of the model from which all other models are derived.

**Batch Model**

Any deviation from the pivot conditions is reflected by a and \( \beta \) values, which are derived from appropriate independent adjustment objects or adjustment object groups (dependent objects). In this instance eq. 4 is rewritten as:

\[
\frac{dC(t)}{dt} = -a \cdot k(C(t))n [C(t) - C_0 \cdot (1-\beta)]
\]

which holds for

\[
C_0 \cdot \beta \cdot [C_0 \cdot C(t_{n+1})] = C(t) - C_0 \cdot \beta \cdot [C_0 \cdot C(t_n)]
\]

This expression also has an analytic solution, which holds for the validity range.

The discussed batch approach may also be applied to continuous reactors if concentration-retention time data for a continuous bank are applied to simulate these reactors, as was done by Dowling, Klimpel and Aplan (1986). This implies that the set \( k(C(t))n \), \( C(t_0) \), \( t_0 \) is calculated from concentration-retention time data for the bank.

**Continuous Model**

This may be defined by appropriate differential equations for a microscopic fluid (Reuter and Van Deventer, 1991a&b) or by the following expression for a macroscopic fluid, which defines the average exit concentration for continuous reactors:

\[
C(t) = \frac{1}{t_0} \int_0^{t_0} E(t) \exp(-k(C(t))n(t-t_0)) \, dt
\]

This equation may be solved analytically for simple E(t) functions, by a Gaussian quadrature (Press and others, 1989) or by Euler's method:

\[
C = C(t_0) + E(t_0) \cdot \Delta t
\]

\( C(t_0) \) is calculated via eq. 5 or 6. In these equations E(t) represents an arbitrary retention time distribution (RTD), which may be a function.
of dead volume, fraction plug and/or mixed flow, number of reactors, retention time etc. (Levenspiel, 1972).

The linear differential equations for microscopic fluids may also be integrated to produce simple analytic solutions.

**Basis for Fault Diagnosis**

The basis of the proposed approach is to determine by what $a$ and $b$ factors the process to be diagnosed deviates from the pivot conditions. By means of a backward chaining process adjustment objects which produce these $a$ and $b$ values, are determined. The use of neural networks to perform the adjustment of $a$ and $b$ values, or to determine process conditions from given $a$ and $b$ values, is presently being investigated.

**Batch reactors.** The $b$-factor is determined easily, i.e. simply the ratio of the final recoveries, where the slopes are zero. The $a$-factor is determined via a time weighed slope integration. Consider in dimensionless form for a first order equation (similar for a second order equation)

$$\frac{1}{k'} \int_{0}^{t} t \cdot k' \cdot C \cdot dt = \frac{1}{k'} \int_{0}^{t} t \cdot k' \cdot e^{-k' \cdot t} \cdot dt$$

(9)

which has been generalized to

$$\frac{1}{k'} = \int_{n}^{0} \frac{t \cdot k' \cdot C(t)}{n} \cdot C(t_{n})' \cdot dt$$

(10)

By comparing the time weighted slope $1/k_{pivot}$ of the pivot set $\{k(t_{n}), C(t_{n}), t\}$ to that of the process to be diagnosed, an estimate for the $a$-factor may be established from the ratio of the time weighted slopes integrations, i.e. $a = k'/k_{pivot}$. For second order processes this ratio also approaches $k'/k_{pivot}$ for $t \to \infty$ and for third order it approaches $(k'/k_{pivot})^3$. For reactions of order 4 or larger, the result of the time weighted integral is also $1/k'$. The time weighted slope $1/k_{pivot}$ for the pivot data is also an ID-tag for the data, which can hence be applied to identify ore types etc. from the kinetic data and its associated pivot data.

It is clear that this approach can also be used to determine the reaction mechanism for unknown reactions if an extensive data base is available.

If the batch simulation approach is used to simulate continuous reactors, the above method may also be applied to diagnose continuous reactor banks.

**Continuous reactors.** If the process is considered to have macroscopic fluid properties a similar integration as that demonstrated by eq. 9 may be performed for continuous reactors.

$$C_{e} \cdot \int_{0}^{t} \frac{E(t) \cdot t \cdot C(t)}{C_{C}'} \cdot dt$$

(11)

where

$$E(t) = \frac{1}{t} \cdot \{V/V_{p}\} \cdot \exp\{-t \cdot (V/V_{p})/(t \cdot R \cdot V_{p}/V)\}$$

or any other suitable function

For N ideal continuous reactors in series ($V_{p}=0$), each with a mean retention time $t'$ and accommodating a first order reaction with a final recovery R, the above integration would produce

$$C_{e} \cdot (1-R) \cdot [R/(1+k' \cdot t')]^{N}$$

(12)

Hence, given a specific RTD for a continuous reactor system, R and $k'$ values calculated from plant data may be compared to the $R_{pivot}$ and $k'_{pivot}$ values of the pivot data, from which $a$ and $b$ values, and hence appropriate operating conditions may be estimated directly.

**Knowledge Base**

Details of the knowledge base are too lengthy to discuss here. It suffices to say that the knowledge is represented in an hierarchical network, the top end (super-class) represents the system being considered and the bottom end the independent adjustment factors mentioned above. These may be of a regression type, i.e. permitting the definition of the Arrhenius equation and fundamental correlations such as $Sh=f(Sc, Ra)$ or of a factor type e.g. (1 of $n$ possible):

- **PROPERTY**
  - $[Na-jarosite]$
  - $1000kg/m^3$
- n-1 INTERACTION FACTORS
  - SLOPE FACTOR
  - $0.39$
  - RECOVERY FACTOR
  - $8-1$
  - COST FACTOR
  - $1$

This adjustment object indicates that the rate decreases by a factor of $0.39$ if the Na-jarosite seed concentration has a value of $1000kg/m^3$ (see Table 8). For this example the adjustment rule, constructed from the pivot adjustment objects, may be written as:

- $Na-jarosite precipitation (CLASS)$
- and Distilling property (SUB-CLASS)
- and $([Fe]^{3+})=30kg/m^3$
- and $([Na]+)=8kg/m^3$
- and $([Zn]^{2+})=100kg/m^3$
- and $([H_2SO_4])=30kg/m^3$
- and $[Na-jarosite]=300kg/m^3$
- and temperature is $95^\circ C$

then $a=1$

$\beta=1$

cost factor=1

These and other objects are grouped into a larger object, the pivot object, defined for each data set. A neural network based architecture is currently being considered in addition to the adjustment object approach.

**Summary**

The strength of the approach discussed in this paper is that it applies the same slope-concentration data set $\{k(C(t_{1}), C(t_{n}), t)\}$ to perform simulation and diagnosis in both batch and continuous processes.

**VALIDATION OF THE KNOWLEDGE BASED MODEL**

The various models discussed in the previous section will be validated by comparing the results with theoretical values. The kinetic data in column 1(a) of all tables and its associated knowledge refer to the pivot object. In the theoretical examples knowledge is not included. Also consult Reuter and Van Deventer (1991a&b) for a variety of other practical and theoretical examples of batch processes.

**Batch reactors**

Two theoretical examples will be given in this section viz.: a first order flotation model with a rectangular distribution of rate constants (Tables 1 and 2) and a second order rate equation (Table...
3 and 4). Cₜ is in both examples. Table 1 and 3 give the respective slopes calculated for each example via eq. 3 (6 geometric mean) and Tables 2 and 4 give the results produced for various adjustment factors via equations 4 to 6.

**TABLE1** Summary of the calculated \( k(C_{\text{tot}})_{\text{m}} \) values according to eq. 3, using the data in column 1(a) in Table 2.

<table>
<thead>
<tr>
<th>C Range</th>
<th>( k(C_{\text{tot}})_{\text{m}} ) (m/s)</th>
<th>( k(C_{\text{tot}})_{\text{m}} ) (m/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0000 to 0.1500</td>
<td>0.495</td>
<td>0.000-0.102</td>
</tr>
<tr>
<td>0.9500 to 0.9965</td>
<td>0.687</td>
<td>0.012-0.204</td>
</tr>
<tr>
<td>9.9564 to 0.8615</td>
<td>0.787</td>
<td>0.064-0.364</td>
</tr>
<tr>
<td>etc.</td>
<td>etc.</td>
<td>etc.</td>
</tr>
</tbody>
</table>

**TABLE2** Columns (a): Concentration-time data from the given model and R and k values. Columns (b) according to eq. 4-6.

<table>
<thead>
<tr>
<th>(a)</th>
<th>Column (b)</th>
<th>Column (b)</th>
<th>Column (b)</th>
<th>Column (b)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C/Co</td>
<td>C/Co,If(I/Co,k,'t)</td>
<td>C/Co</td>
<td>C/Co</td>
<td>C/Co</td>
</tr>
<tr>
<td>0.00</td>
<td>1.000</td>
<td>0.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>0.50</td>
<td>0.666</td>
<td>0.627</td>
<td>0.645</td>
<td>0.605</td>
</tr>
<tr>
<td>0.05</td>
<td>0.500</td>
<td>0.450</td>
<td>0.496</td>
<td>0.466</td>
</tr>
<tr>
<td>0.00</td>
<td>0.025</td>
<td>0.015</td>
<td>0.015</td>
<td>0.015</td>
</tr>
<tr>
<td>1.00</td>
<td>0.150</td>
<td>0.120</td>
<td>0.120</td>
<td>0.120</td>
</tr>
<tr>
<td>1.00</td>
<td>0.175</td>
<td>0.142</td>
<td>0.142</td>
<td>0.142</td>
</tr>
<tr>
<td>2.00</td>
<td>0.200</td>
<td>0.161</td>
<td>0.161</td>
<td>0.161</td>
</tr>
<tr>
<td>3.00</td>
<td>0.221</td>
<td>0.183</td>
<td>0.183</td>
<td>0.183</td>
</tr>
<tr>
<td>4.00</td>
<td>0.241</td>
<td>0.205</td>
<td>0.205</td>
<td>0.205</td>
</tr>
</tbody>
</table>

**TABLE3** Summary of the calculated \( k(C_{\text{tot}})_{\text{m}} \) values according to eq. 3 (6 geometric mean), using the data in column 1(a) in Table 4.

<table>
<thead>
<tr>
<th>C Range</th>
<th>( k(C_{\text{tot}})_{\text{m}} ) (m/s)</th>
<th>( k(C_{\text{tot}})_{\text{m}} ) (m/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00 to 0.66</td>
<td>16.000</td>
<td>16.330</td>
</tr>
<tr>
<td>0.50 to 0.50</td>
<td>11.643</td>
<td>11.547</td>
</tr>
<tr>
<td>1.00 to 0.40</td>
<td>0.494</td>
<td>0.477</td>
</tr>
<tr>
<td>0.33 to 0.28</td>
<td>0.393</td>
<td>0.350</td>
</tr>
<tr>
<td>0.17 to 0.10</td>
<td>0.117</td>
<td>0.117</td>
</tr>
<tr>
<td>etc.</td>
<td>etc.</td>
<td>etc.</td>
</tr>
</tbody>
</table>

**TABLE4** Columns (a): Concentration-time data from the given model and R and k values. Columns (b) according to eq. 4-6. Columns (c) using geometric mean.

<table>
<thead>
<tr>
<th>(a)</th>
<th>(b)</th>
<th>(c)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.00</td>
<td>0.000</td>
<td>1.000</td>
</tr>
<tr>
<td>4.00</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>5.00</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>6.00</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>7.00</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>8.00</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>9.00</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>10.00</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>0.10</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>0.20</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>0.30</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>0.40</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>0.50</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>0.60</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>0.70</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>0.80</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>0.90</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>1.00</td>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

It is clear from the above examples that no variances exist in \( a \) and \( b \) values, that the simulation is almost exact and that no fitting is performed, which are the proposed criteria for an appropriate kinetic model to be used in a model based diagnostic approach.

**Continuous reactors**

In this section continuous reactors will be simulated by following a macroscopic fluid approach. In each case the theoretical eq. 11 is compared to the results produced by the approximate eq. 7. To calculate the slope data set, eq. 3 was applied throughout to produce the results summarized in Tables 5 and 6. It is clear from the previous examples that a geometric mean would have produced more accurate results. However, in most cases the correspondence is adequate. The integration was performed by a 15 point Gaussian Quadrature (Press and others, 1989). Note that the top integration limit was not taken to be infinity. The given top limit was sufficient to calculate the average concentration from a continuous reactor for both the first and the second order processes. Note, however, the small variations in the answers for the second order processes when the top limit changes from 20 to 10 in Tables 5 and 6 respectively. The given theoretical solution for the second order process is calculated via \( (1/\text{RT}) \cdot \exp(-E/\text{RT}) \) (Levenspiel, 1972: Eisrror integral).

**TABLE5** Comparison of eq. 7 and 11 subject to different parameters for the respective kinetic and RTDs. \( C(t) \) is calculated from the given kinetic models and \( k(C_{\text{tot}})_{\text{m}} \) from these respective data (Integration limits 0 and 20).

<table>
<thead>
<tr>
<th>Equation and parameters</th>
<th>( \alpha = \beta \cdot C_{\text{tot}}(1-\alpha t) ) and ( \alpha = 0.1 )</th>
<th>( e^t )</th>
<th>( 1/(1+t) )</th>
<th>( 1/(1+St) )</th>
<th>( 1/(1+St) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>eq. 7 (Theoretical)</td>
<td>0.5000</td>
<td>0.5000</td>
<td>0.1667</td>
<td>0.0909</td>
<td>0.0909</td>
</tr>
<tr>
<td>eq. 6</td>
<td>0.5000</td>
<td>0.5000</td>
<td>0.1667</td>
<td>0.0909</td>
<td>0.0909</td>
</tr>
<tr>
<td>eq. 11 ( \alpha = 0.1 )</td>
<td>0.3333</td>
<td>0.3333</td>
<td>0.0667</td>
<td>0.2083</td>
<td>0.1250</td>
</tr>
<tr>
<td>eq. 10 ( \alpha = 0.1 )</td>
<td>0.3333</td>
<td>0.3333</td>
<td>0.0667</td>
<td>0.2083</td>
<td>0.1250</td>
</tr>
<tr>
<td>eq. 9 ( \alpha = 0.1 )</td>
<td>0.3333</td>
<td>0.3333</td>
<td>0.0667</td>
<td>0.2083</td>
<td>0.1250</td>
</tr>
<tr>
<td>eq. 8 ( \alpha = 0.1 )</td>
<td>0.3333</td>
<td>0.3333</td>
<td>0.0667</td>
<td>0.2083</td>
<td>0.1250</td>
</tr>
</tbody>
</table>

In each of the examples a good correspondence is achieved between the theoretical and approximate models. For the geometric mean the correspondence appears to be much better which permits a larger \( \alpha \) to be taken. From the first slope in Table 3 calculated via eq. 3 it is clear that \( k(C(t))_{\text{m}} = 16\times19.2 \, ^\circ\text{C} \) for this interval (0.66xCto1), and for the geometric mean \( k(C(t))_{\text{m}} = 20\, ^\circ\text{C} \), which is an exact approximation, since \( a = -20\, ^\circ\text{C} \) is the second order equation for these kinetic pivot data.

Stellenbosch University  https://scholar.sun.ac.za
TABLE 6 Comparison of eq. 7 and 11 subject to different parameters for the respective kinetic and RTDs. By applying the data from column 1(a) of Table 4 as input to calculate \( k(C(t)) \cdot C(t) \), the data may be calculated by eq. 6 (Integration limits 0 and 10).

<table>
<thead>
<tr>
<th>Equation parameters</th>
<th>( \frac{1}{1+10^2t} )</th>
<th>( \frac{1}{1+10^4t} )</th>
<th>( \frac{1}{1+10^8t} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_i ); ( \beta ); ( \alpha ); ( \omega )</td>
<td>0.0176</td>
<td>0.0167</td>
<td>0.0090</td>
</tr>
<tr>
<td>eq. 5; ( t ) (Theoretical)</td>
<td>0.0172</td>
<td>0.0158</td>
<td>0.0099</td>
</tr>
<tr>
<td>eq. 11 ( t_2, t_1, t_0 = 0.0 )</td>
<td>0.0788</td>
<td>0.0786</td>
<td>0.0656</td>
</tr>
<tr>
<td>( t_2, t_1, t_0 = 0.0 ) (Theoretical)</td>
<td>0.0893</td>
<td>0.0814</td>
<td>0.0617</td>
</tr>
<tr>
<td>eq. 11 ( t_2, t_1, t_0 = 0.0 )</td>
<td>0.0315</td>
<td>0.0317</td>
<td>0.0217</td>
</tr>
<tr>
<td>eq. 11 ( t_2, t_1, t_0 = 0.0 )</td>
<td>0.0390</td>
<td>0.0390</td>
<td>0.0390</td>
</tr>
<tr>
<td>eq. 11 ( t_2, t_1, t_0 = 0.0 )</td>
<td>0.0218</td>
<td>0.0218</td>
<td>0.0218</td>
</tr>
<tr>
<td>eq. 11 ( t_2, t_1, t_0 = 0.0 )</td>
<td>0.0158</td>
<td>0.0158</td>
<td>0.0158</td>
</tr>
</tbody>
</table>

Application to practical data

Both examples discussed in this section were from the zinc ferrimettalurgy.

- Zinc-ferrite leaching. The data for this example were taken from Rastas and others (1979), who could describe these data by reaction controlled shrinking core model. The model applied by Rastas and others was a reaction-controlled shrinking core model. As may be seen from Table 7 the correspondence is very good. Also note the use of adjustment factors and their associated objects, e.g. for column 2 the temperature object would be (assuming only two objects viz. temperature and \([H_2SO_4]\)):

TABLE 7 Conversion-time data for No 4 Zinc-ferrite (Rastas and others, 1979).

<table>
<thead>
<tr>
<th>PROPERTY</th>
<th>Temperature, °C</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUB-PROPERTY</td>
<td>85°C</td>
</tr>
<tr>
<td>1 INTERACTION</td>
<td></td>
</tr>
<tr>
<td>FACTOR</td>
<td>1</td>
</tr>
<tr>
<td>GLOPE FACTOR</td>
<td>0.03</td>
</tr>
<tr>
<td>RECOVERY FACTOR</td>
<td>0.5</td>
</tr>
<tr>
<td>COST FACTOR</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PROPERTY</th>
<th>Temperature, °C</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUB-PROPERTY</td>
<td>85°C</td>
</tr>
<tr>
<td>1 INTERACTION</td>
<td></td>
</tr>
<tr>
<td>FACTOR</td>
<td>1</td>
</tr>
<tr>
<td>GLOPE FACTOR</td>
<td>0.03</td>
</tr>
<tr>
<td>RECOVERY FACTOR</td>
<td>0.5</td>
</tr>
<tr>
<td>COST FACTOR</td>
<td>1</td>
</tr>
</tbody>
</table>

Identification of Na-jarosite precipitation. The example being discussed here is the precipitation of \( Fe^{2+} \) as Na-jarosite (Rastas and others, 1979):

\[
3Fe_2(SO_4)_3 + Na_2SO_4 + 12H_2O \rightarrow 2NaFe_2(SO_4)_2(OH)_6 + 6H_2SO_4
\]

The kinetics of this reaction are affected by a variety of factors which include the temperature, the concentration of the Na-jarosite seed, \([Fe^{2+}]_0 \) and \([Na^+]_0 \). The reaction kinetics for this reaction have been represented by a variety of complex models, e.g.:

\[
dFe^3/\alpha dt = k(T)(Fe^{3+})^2(Na^{+})^2[Na-jarosite]^C(H_2SO_4)^d
\]

(Rastas and others, 1979)

or

\[
dFe^3/\alpha dt = k(T)(Fe^{3+})^2(Na^{+})^2[Na-jarosite]^2/2 \cdot k(H_2SO_4)^1/4
\]

(Wang Qian-kun and others, 1985)

From the results in Table 8 it is clear that the proposed simple kinetic model and its associated knowledge base can produce the same and better results as the rather complex and semi-empirical multi-parameter kinetic models given above. To predict the data in column 3(b) of Table 8 during simulation, appropriate adjustment objects would produce the following adjustment rule:

if Na-jarosite precipitation (CLASS)
and Distilling property (SUB-CLASS)
and \([Fe^{2+}] = 30kg/m^3\)
and \([Na^+] = 8kg/m^3\)
and \([Zn^{2+}] = 100kg/m^3\)
and \([H_2SO_4] = 30kg/m^3\)
and temperature is 85°C
then \(a = 1 \cdot 1 \cdot 1 \cdot 0.39 \cdot 0.39 \cdot 0.152 \)
\(b = 1 \cdot 1 \cdot 1 \cdot 1 = 1 \)
cost factor = 1

TABLE 8 Na-Jarosite precipitation-time data (Rastas and others, 1979).

<table>
<thead>
<tr>
<th>Table 8</th>
<th>Data from Rastas and others</th>
<th>Eq. - 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>( x )</td>
<td>( y )</td>
</tr>
<tr>
<td>0.00</td>
<td>30.0</td>
<td>30.0</td>
</tr>
<tr>
<td>0.05</td>
<td>26.1</td>
<td>26.3</td>
</tr>
<tr>
<td>0.10</td>
<td>22.7</td>
<td>23.9</td>
</tr>
<tr>
<td>0.15</td>
<td>19.4</td>
<td>21.1</td>
</tr>
<tr>
<td>0.20</td>
<td>16.1</td>
<td>17.4</td>
</tr>
<tr>
<td>0.25</td>
<td>12.5</td>
<td>16.0</td>
</tr>
<tr>
<td>0.30</td>
<td>9.7</td>
<td>12.8</td>
</tr>
<tr>
<td>0.35</td>
<td>6.2</td>
<td>9.0</td>
</tr>
<tr>
<td>0.40</td>
<td>4.4</td>
<td>6.6</td>
</tr>
</tbody>
</table>

FAULT DIAGNOSIS

Diagnosis in this paper is taken to imply the detection of faults within the respective unit operations, i.e. deviations from process conditions within metallurgical reactors. Fault-diagnosis is discussed under three headings viz. the identification of a class and sub-class of a meta-class, the identification of a class and sub-class of a meta-class. This method is based on comparing the final recovery and time-weighted slope of kinetic data of an as yet unknown class to those available in the knowledge base. This permits the identification of the unknown class. If the batch
flotation kinetic data are given, the possible one body and possible flotation conditions could be established for a meta-class, i.e. for example the flotation for a specific plant or one body. If the knowledge base is constructed consisting only of pivot objects (no adjustment objects), the above process could also be applied to perform process identification. It is clear that the associated final recovery and time weighted slope data define both batch and continuous (C retention time data) processes.

Identification of the process conditions within a batch reactor. Consider the data in columns 1(a) and 2(a) of Table 4. The theoretical time weighted slope columns are 0.1261 and 0.2561 which implies $\beta = 0.5$. Since the inference engine would hence attempt to establish which process conditions would produce the given adjustment factors. The practical data summarized in Tables 7 and 8 could not be applied here as the data do not cover the process to the final recovery and precipitation respectively.

Identification of the process conditions within a continuous reactor. Two methods will be pointed out in this respect. A third possible method based on a microscopic fluid approach is discussed by Reuter and Van Derenter (1991b).

Method 1 applies equation 9 and 10 as for batch processes, to describe the kinetcs of the flotation bank as a whole. This approach implies that the pivot objects are defined accordingly i.e. they do not include a batch. The kinetic data but common concentration-retention time (or stage number) data. All adjustment objects are defined accordingly with reference to these data. One would obviously not be able to perform process identification of batch cells with these data, hence forfeiting generality. This method is hence identical to that discussed previously for batch reactors.

Whereas the batch fault-diagnosis procedure as applied in method 1 applies eq. 9 or 10 to determine an estimate of $a$, equations 7 and 8 or 12 are applied as the basis of diagnostic procedure for method 2. This implies that batch kinetic data form the basis of this method.

It has already been stated that for a first order rate equation and an ideal retention time distribution ($V_m = 1$) the solution of equation 11 is $1/(1+k\cdot t)$. A meta-rule based on this equation has been given above (eq. 12) for estimating $a$ and $\beta$ values. If the solution for the kinetic represented by $e^{-t}$ are used to solve equation 7 or 11, a result of $C_m = -0.333$ is produced for $V_m = 1$ and $t = 2$. For the kinetic model $e^{-2t}$ this result would be $C_m = -0.2$. It is a trivial exercise to work backwards from the given $C_m$ values, via equation 12, to obtain the respective rates of 1 and 2 for $t = 2$. This would imply that $a$ is equal to 2 with reference to the pivot value of $k = 1$. It is obvious that the meta-rule engine in eq. 12 can only produce an exact value for $a$ if the process has first order kinetics. The meta-rule can, however, give an estimate for an initial a value which can be refined by a subsequent iterative procedure. Since the solution of equation 7 is rapid, this iterative procedure is fast. As an example consider the data for $t = 2$ and $V_m = 1$ in Table 5 for the second order equations. If the data for the second order equation $1/(1+k\cdot t)$ are taken as the pivot, the following three $a$-values could be estimated, viz. $1.34$ and $5.9$ for the given integration limits from 0 to 20. From this data the process conditions can subsequently can be estimated. The theoretical $a$ values are 1, 5 and 10 respectively for $\beta = 1$.

DISCUSSION AND SIGNIFICANCE

A simple generalized approach was discussed that permits both the dynamic simulation of batch and continuous processes. The basis is process independent and has been applied to simulate adsorption, leaching, reduction, precipitation and flotation processes. The same basis has been shown to permit diagnosis of batch as well as continuous processes.

The above aspects are possible since the proposed kinetic model gives an exact description of the above mentioned processes without forfeiting accuracy due to curve fitting. The "one parameter" model is well suited for manipulation by adjustment factors in an object architecture. At present a neural network architecture is being investigated to relate the adjustment factors $a$ and $\beta$ respectively to process conditions.

REFERENCES


Section 9  Publications on

Neural Networks:
Theoretical Concepts
A GENERALIZED NEURAL-NEt KINETIC RATE EQUATION

M. A. REUTER
Institut für Metallhüttenwesen und Elektrometallurgie, RWTH-Aachen, Inizstraße 3, 5100 Aachen, Germany

and

J. S. J. VAN DEVENTER and T. J. VAN DER WALT
Department of Metallurgical Engineering, University of Stellenbosch, 7600 Stellenbosch, South Africa

(First received 28 October 1991; accepted in revised form 12 June 1992)

Abstract—This paper illustrates the application of a generalized neural-net kinetic rate equation to perform identification of the process conditions and dynamic simulation of batch and continuous metallurgical and mineral processing systems. The proposed approach is useful essentially to simulate ill-defined dynamic processes where the existing fundamental and empirical models fail owing to their lack of generality. The basis of the kinetic rate equation is a set \( \{k(C), C, t\} \) which is derived directly from batch reactor concentration-time data or concentration-mean-residence-time kinetic data of a system of continuous stages in series and its associated pivot process conditions. The generalized kinetic rate equation is defined by using the above set and linked to a trained neural net via adjustment factors \( a \) (for adjusting the rate) and \( b \) (for adjusting the final recovery), which relate the process conditions within the reactor to the kinetics of the process under consideration. This approach permits the dynamic simulation of a variety of metallurgical and mineral processing systems at any process conditions catered for by the trained neural net. The same set of kinetic data may also be used to establish deviations from the pivot conditions quantified as \( a \) and \( b \) values. Conversely, these \( a \) and \( b \) values permit the identification of process conditions within the reactor, which could be used in process identification. In contrast to most of the existing models, no curve-fitting is required, as the kinetic rate equation utilizes experimental data directly. A number of theoretical, mineral processing and hydrometallurgical case studies are used to demonstrate the applicability of the generalized neural-net kinetic rate equation.

INTRODUCTION

Schumann (1942) explained that the kinetic behaviour of minerals during flotation will in many cases supply worthwhile information regarding the process and, hence, the effect of the various process conditions. He also demonstrated that changes of process conditions within flotation cells have a measurable effect on the flotation response. These observations based on the kinetics of the process form the basis of the simulation and model-based identification procedure discussed in this paper.

Numerous models exist to describe the kinetics of leaching, flotation, reduction and other processes (Sohn and Wadsworth, 1979; Dowling et al., 1986). Taking flotation as an example, Dowling et al. (1986) showed that the kinetic flotation model based on a first-order process for which the rate is a rectangular distribution of rate constants was superior to 13 other models. However, by fitting this kinetic model to the given data, the least-squares fitting procedure produced a fitted final recovery of \( > 100\% \). Other flotation models performed badly owing to a large confidence limits on the fitted parameters. The models for leaching, precipitation, reduction and adsorption would exhibit similar behaviour, unless, however, the kinetic model is an exact representation of the process being considered. This is usually not the case owing to the complex kinetic mechanism underlying most of these processes.

The large number of, in some cases inexact, kinetic models that are available for defining the kinetics of metallurgical and flotation processes make a generalized kinetic modelling approach hardly possible. Basing reactor simulations on such inexact flotation models, for example, is also not an ideal situation. As an attempt to improve the accuracy of kinetic modelling, Reuter and van Deventer (1991) demonstrated how the same generalized kinetic rate equation may be used to model the kinetics of leaching, pyrometallurgical reduction, resin adsorption, carbon-in-pulp and carbon-in-leach systems very accurately. Contrary to the published flotation models, which can produce fitted recoveries of \( > 100\% \) (Dowling et al., 1986), final recoveries, which are in total agreement with the kinetic data, are produced by the generalized kinetic rate equation.

As has been shown by Lynch et al. (1981), the parameters of a three-parameter kinetic flotation model can be related to the reagent levels within the flotation cell via a regression correlation. If a generalized approach is not opted for, it would mean that a regression equation, which relates the process conditions within the flotation cell, would have to be defined for each flotation model. A generalized kinetic rate equation, on the other hand, would permit the
definition of one regression equation only. This was done by Reuter and van Deventer (1991) using a generalized object-orientated knowledge-based methodology, based on independent adjustment objects in an hierarchical network. A backward-chaining method (Shortliffe and Buchanan, 1975) was used to adjust the kinetic parameters of the generalized kinetic rate equation. If suitable mass balance equations, with their parameters such as the mean residence time, etc., are used in conjunction with the generalized kinetic rate equation and associated knowledge base, continuous stirred-tank (CSTR) and plug-flow reactors can be simulated (Reuter and van Deventer, 1991).

A limitation of the object-orientated architecture is the large knowledge base and the accompanying rules which are necessary to describe a particular system. In the present paper it will be demonstrated how a trained neural net (NN) can be used to replace the mentioned object-orientated knowledge base, so as to relate the process conditions to the parameters of the generalized kinetic rate equation. This NN approach gives a more compact and generalized knowledge representation than the object-orientated approach and also includes the interesting features of NNs, i.e. their ability to approximate any functional relationship. A regression approach could have been adopted; however, the selection of an appropriate general regression equation would be problematic. The ability of a neural net to approximate any functional relationship makes the selection of a suitable regression equation for a particular application unnecessary.

NNs are inherently parallel and have the capability to learn non-linear relationships, which may exist between a set of inputs and outputs (Hornik et al., 1989). As a modelling tool, neural nets have already been applied to models and simulate various chemical reactors by multilayer feedforward nets (Bhat et al., 1990). Reuter et al. (1992) used neural nets to model slag-metal equilibrium processes in pyrometallurgical reactors. Bhat and McAvoj (1990) also used a feedforward NN (backpropagation learning) to control the pH in a CSTR. Ungar et al. (1990) discuss the use of linear adaptive networks for the diagnosis of a simple CSTR and the control of a bioreactor. These authors, however, point out that the limitations of these nets are their slow learning (large number of iterations before convergence), rapid forgetting (due to seldom-seen input-output pairs) and the lack of first-principles knowledge. Further applications of NNs to process diagnosis have been reported by Hoskins et al. (1991), who diagnosed a process using two hidden layers in the NN, and Hoskins and Himmelblau (1988), who diagnosed CSTRs. Watanabe et al. (1989) propose a two-stage feedforward NN (backpropagation learning) architecture, each stage with one hidden layer for the diagnosis of chemical processes. Venkatasubramanian et al. (1990) discuss the application of multilayer feedforward NNs (backpropagation learning) to the fault diagnosis of a reactor and a reactor-distillation-column arrangement. These authors analysed the learning, recall and generalization characteristics of various NN topologies, i.e. hidden layers, and hidden nodes, and found that the NNs could diagnose correctly for a variety of case studies.

Venkatasubramanian et al. (1990) also point out that NNs cannot give reasons for their diagnoses as expert systems are able to do, while Ungar et al. (1990) point to the black box nature of these NNs. By including a generalized kinetic rate equation at the output side of a NN when simulating a reactor, first-principles knowledge in the form of an equation may be included in the NN-architecture, rendering the NN less of a blackbox, the aim of this paper. This union of a generalized kinetic rate equation and the NN permits the simulation of batch, plug-flow and mixed-flow reactors at all chemical and physical process conditions catered for by the NN as well as the flow conditions catered for by the mass balance equations.

The trained-neural-net equation and the generalized kinetic rate equation together form a large source of knowledge regarding a specific process. It will also be shown how this knowledge may be used to identify the process conditions within both batch reactors and mixed-flow reactors. The identification process implied here refers to establishing the reagent levels, stirring speeds, temperature, pulp densities, pH, particle size, etc., and their effects on the kinetics within the reactor and not fault diagnosis in the macroscopic sense, such as identifying pumps not functioning, high liquid levels in reactors, etc.

**OVERALL ARCHITECTURE**

In the proposed approach an architecture is adopted in which one set of concentration-time data and the corresponding process conditions form the pivot on which all further predictions are based. This data set is, therefore, called the pivot data set. Any deviation in the process conditions are quantified as non-unit and larger-than-zero adjustment factors, which adjust the rate and the final recovery of the proposed generalized kinetic model.

**Illustrative example**

The architecture is best explained by an illustrative example, which is based on a simple first-order kinetic model $C = 1 - R (1 - e^{-kt})$. In the proposed architecture $R$ and $k$ are determined from the pivot kinetic $C-t$ data and these, together with the accompanying process conditions, e.g. reagent levels, temperature, pH, pulp density, particle size, etc., are termed the pivot data. If the process conditions deviate from the pivot process conditions, $k'$ and $R$ change, the change being quantified as the adjustment factors $\alpha$ and $\beta$. These adjustment factors modify the above simple kinetic model to $C = 1 - \beta R (1 - e^{-k't})$, which implies that $\alpha = 1$ and $\beta = 1$ for the pivot data.

The meaning of an adjustment factor, $\alpha$, is clarified by the following two examples:

- If $T_0$ is the pivot temperature of the above reaction, for example, the rate $k'$ would have to be
For real metallurgical and flotation processes, however, matters are more complicated, and a complex function of a variety of process conditions. This complex function will be defined by a trained neural net. Similarly, changes in the process conditions may modify the final recovery from the above illustrative example is clear that if \( \alpha \) is known from the \( C-t \) data, \( T \) may be determined via the above Arrhenius equation since \( T_0 \), a pivot process condition, is also known. In the proposed architecture the relationship between \( \alpha \) and \( \beta \), and the process conditions, is defined by a trained neural net.

For identifying process conditions within the reactor, the reverse procedure is followed. From the above illustrative example it is clear that if \( \alpha \) and \( \beta \) are adjusted as a function of \( Sh \), e.g. \( \alpha = f(Sh) \), \( \beta = \frac{R}{R_{\text{pivot}}} \), which is also a complex function of the process conditions within real reactors.

Figure 1 formalizes the above discussion and gives a general architecture of the proposed simulation and identification activities. For simulation purposes, a trained neural net forms the front end of a generalized kinetic rate equation, i.e. it is used to adjust the kinetic parameters of a generalized kinetic rate equation via \( \alpha \) and \( \beta \) adjustment factors as a function of the process conditions. In conjunction with suitable mass balance equations, reactor simulation may subsequently be performed. For identification purposes, the generalized kinetic rate equation may be used to produce the input (\( \alpha \) and \( \beta \) adjustment factors) to a neural net, which may subsequently be used to identify the process conditions within the reactor.

The algorithm for training a neural net with appropriate \( \alpha \), \( \beta \) and process conditions data sets for simulation and identification activities is also depicted in Fig. 1. More details on this activity will be given below.

**Neural nets**

Before more details on the type of neural nets implemented in the proposed architecture is given, a short overview of the theory is given.

Lippmann (1987) discussed various NN topologies such as Hopfield nets, Hamming nets, Carpenter-Grossberg classifiers, etc. For the purpose of this paper, the applied multilayer feedforward NNs are not trained by backpropagation using the generalized delta rule (Rumelhart et al., 1986), but by a conjugate-gradient optimization (Powell, 1977) procedure proposed by Barnard and Cole (1989). According to these authors, the heuristic algorithms for choosing the learning rate and momentum term (parameters in the GDR algorithm) cannot compete with optimized conjugate-gradient training as far as speed and robustness are concerned. This improvement attempts to bridge the criticism of Ungar et al. (1990) given in the introduction, concerning the speed of convergence.

The multilayer feedforward nets that will be trained for each of the examples in this paper consist of a number of layers, each containing a number of nodes, viz. the input layer with a bias node with activation 1, one or more hidden layers and the output layer.
put layer. Adjacent nodes in the layers are exhaustively interconnected by weighted branches as may be seen from Fig. 2, which depicts a two-layer net (input layer not counted). The input and output nodes of these nets have linear and the hidden nodes sigmoidal activation functions, respectively. The neural nets are trained by a training data set (Barnard and Cole, 1989) and subsequently tested by a test data set, a random subset of the total data set. An additional input, which quantifies the exactness of the inputs, may be added to the neural net given in Fig. 2. This would enable the neural net to classify inputs as a function of errors in the input. Venkatasubramanian et al. (1990) showed that their neural net with 78 weights and 12 data sets for the fault detection for a reactor–distillation arrangement (net with six inputs + bias/five hidden nodes + bias/six outputs) could classify faults correctly in spite of faulty sensors. This indicates the robustness and fault-tolerance capabilities of neural nets. Similar comments were made by Hoskins and Himmelblau (1988).

Venkatasubramanian et al. (1990) found a two-layer net to be sufficient for their applications, implying that their applications were not complicated enough to justify two hidden layers to be defined. It has, however, been pointed out (Lippmann, 1987) that a two-hidden-layer network is capable of discriminating any complex region. Furthermore, Hornik et al. (1989) proved that NNs have the ability to find any non-linear relationship between inputs and outputs without having a priori knowledge about the system, provided that sufficient hidden nodes and hidden layers are chosen. In this capacity a neural net will be applied to relate process conditions to the adjustment factors \( \alpha \) and \( \beta \) as depicted by Fig. 1, or to relate the adjustment factors to possible process conditions during process identification.

With the architecture given in Fig. 1, it is attempted to place concentration–time predictions, with the aid of a NN for metallurgical reactors, on a more fundamental basis, hence, including first-principles knowledge. During process identification, predictions of process conditions are based on calculated \( \alpha \) and \( \beta \) factors. This implies that the NN can give a reason for its predictions, i.e. it can show how much the process conditions affect the rate and the final recovery. With this approach, it is possible to include the reasoning capabilities of expert systems during process identification, which is an inherent weakness of NNs.

**GENERALIZED KINETIC RATE EQUATION**

The usual method followed during the modelling of the kinetic data given in Fig. 3, for example, is (i) the development of a kinetic model and (ii) the sub-

\[
1 - \alpha (1 - C_n) = k(C)_n (1 - \beta (1 - C_{n+1}))
\]

\[
C_n = \frac{k(C)_n}{1 - \beta (1 - C_{n+1})}
\]

\[
C_{n+1} = \frac{1}{1 - \alpha (1 - C_n)}
\]

\[
(1 - R) = \frac{k(C)_n}{(1 - \alpha (1 - C_n))}
\]

\[
(1 - R) = \frac{1}{1 - \beta (1 - C_{n+1})}
\]

**Fig. 3.** Kinetic data and the derivation of \( k(C)_n \) from these data.
sequent least-squares fitting of the kinetic model with its various parameters to the data.

If it is postulated that the kinetic data in Fig. 3 are described by the simple first-order model, 

\[ C = 1 - R(1 - e^{-kt}) \]

the parameters \( R \) (final recovery) and \( k \) (rate constant, \( s^{-1} \)) would be determined by least-squares fitting.

It is obvious that the above approach is not general as it is a function of the kinetic rate equation, which is derived to describe the specific kinetics under consideration. Since metallurgical and flotation processes may be described by a myriad of different kinetic models, the above approach would not be suitable in a generalized kinetic modelling approach, which is strived for in this paper. The above first-order kinetic model would be capable of describing only a very limited number of metallurgical processes accurately. If, however, \( k \) is not a constant but a variable in concentration, e.g. \( k(C) \), it has been shown (Reuter and van Deventer, 1991, 1992) that the proposed kinetic rate expression can describe the kinetics of a variety of metallurgical and flotation processes accurately. Hence, the normal first-order rate law, \(-r = k'C\), is rewritten as

\[ \frac{dC}{dt} = -k(C)C \]  

(1)

where \( C(0) = 1 \) or \( C_0 \) (if not dimensionless) and \( C(\infty) = 1 - R \).

This definition implies that:

- if the process is first-order in nature then \( k(C) \) is equal to \( k' \),
- for a second-order process, \( k(C) \) is equal to \( k'C \), i.e. a linear function of concentration, and
- if the process is characterized by mixed kinetics, \( k(C) \) is a complex function of \( C \) (and of \( t \) in certain cases; however, not discussed here).

This definition would suggest that \( k(C) \) is reaction-specific and a "fingerprint" of a particular reaction under the measured process conditions (i.e. pulp densities, temperature, reagent levels, pH, mixing conditions, etc.), and could, hence, be used as a basis for simulation and reactor identification activities. Note that this kinetic rate equation is essentially a "one-parameter" expression since \( k(C) \) also contains information regarding the final recovery, i.e. where \( k(C) \) becomes zero \([C(\infty) = 1 - R] \). For this reason, it is, therefore, essential that \( k(C) \) covers the process over the full extent of the reaction, as depicted by Fig. 3.

As the approach discussed here attempts to define a methodology not based on statistics, i.e. \( k(C) \) should not be based on a function of various parameters to be fitted, an alternative procedure is proposed. This is facilitated by defining \( k(C) \) as a discrete function of \( C \), which is a discrete set of average values for \( k(C) \) and their associated validity ranges, calculated directly from smoothed practical kinetic data via eq. (1) by rewriting this equation into a discretized form:

\[ k(C) = \frac{-\delta C}{\delta t C_{av}} \]  

(2)

\( C_{av} \) is taken to be either the arithmetic or geometric average between two successive data points from the smoothed (smoothing is also possible with a neural net trained with the \( C \)-\( t \) data) kinetic data set separated by the time interval \( t_{n+1} - t_n \). For an arithmetic mean, eq. (2) becomes (see Fig. 3)

\[ k(C)_n = \frac{-2(C_{n+1} - C_n)}{(C_{n+1} + C_n)(t_{n+1} - t_n)} \]  

(3)

Substituting these discrete rates for \( k(C) \) in eq. (1),

\[ \frac{dC}{dt} = -k(C)_n C \]  

(4)

where \( k(C)_n \) holds for \( C_{n+1} \leq C \leq C_n \) and \( C(0) = 1 \) or \( C_0 \) (if not dimensionless).

The process conditions for which this set \( \{k(C)_n, C_n, t_n\} \) of rates is determined are termed the pivot process conditions and it is postulated that this set remains independent of the process conditions, i.e. the kinetic mechanism does not change. The pivot conditions are those conditions such as temperature, density, reagent levels, pH, stirring conditions, etc., prevalent during the measurement of the kinetic data. The solution of eq. (4) would follow the algorithm below:

- At \( t = 0 \) and \( C_0 \) the rate \( k(C)_0 \) is selected.
- \( C \) is calculated via numerical integration (e.g. Runge–Kutta) for the next time increment \( \delta t \).
- If \( C_1 \leq C(0 + \delta t) \leq C_0 \) then \( k(C)_0 \) is selected for the next numerical integration step, or if \( C_2 \leq C(0 + \delta t) \leq C_1 \) then \( k(C)_1 \) is selected, etc.
- Return to step 2.

The above algorithm may be replaced by a trained neural net, which relates the concentration \( C \) to the output \( k(C) \).

This definition permits the simulation of processes in which the kinetic data are not a monotonic decreasing function in time, i.e. it could be S-shaped, for example. An example is the induction period in the concentration–mean-residence-time data for the Tennessee copper rougher circuit of flotation cells (Dowing et al., 1986), which cannot really be defined by the available theoretical kinetic models. It is exactly this induction period which could have a decisive influence during the identification of the flotation rougher circuit.

Equation (4) has an analytic solution for the given range, producing the recursive kinetic expression

\[ C = C_n \exp \{-k(C)_n(t - t_n)\} \]  

(5)

where the range of \( t \) is such that \( C \) does not fall outside the validity range of \( k(C)_n \).

As will be seen later on, \( \{k(C)_n, C_n, t_n\} \) is the basis of the kinetic rate equation, from which all other \( C-t \) data may be predicted, provided that the mechanism does not change.

The above kinetic rate equation has been derived for a single species, but may easily be expanded for multispecies systems, as has been shown by Reuter and van Deventer (1989) for the competitive adsorption of Au and Ag on activated carbon.
Generalized kinetic rate equation for a batch reactor

As discussed in the previous section, the generalized kinetic rate equation (4) holds for the pivot conditions. As was pointed out earlier, deviations from the pivot conditions change the rate variable $k(C)$ and the final recovery $R$. This change is quantified as $\alpha$ and $\beta$ adjustment factors.

It is easily shown that the rate equation (4) may be rewritten as

$$\frac{dC}{dt} = -\alpha k(C)(C - C_0(1 - \beta))$$

(6)

which holds for

$$C_0 - \beta(C_0 - C_{s+1}) \leq C(t) \leq C_0 - \beta(C_0 - C_s).$$

The initial concentration is unity (or $C_0$ if not dimensionless) and $C(\infty) = C_0(1 - R)$. $\alpha, \beta > 0$ if $\alpha$ and $\beta$ are not equal to 1. The algorithm for solving eq. (6) is:

- At $t = 0$ and $C_0$ the rate $k(C_0)$ is selected.
- $\alpha$ and $\beta$ are determined by an appropriate trained neural net.
- $C$ is calculated via numerical integration for the next time increment $\delta t$.
- If $C_1 \leq C(0 + \delta t) \leq C_0$ then $k(C_0)$ is selected for the next numerical integration step, or if $C_2 \leq C(0 + \delta t) \leq C_1$ then $k(C_1)$ is selected, etc.
- Return to step 2 and adjust $\alpha$ and $\beta$ if the process conditions have changed within the time increment $\delta t$.

Dowling et al. (1986) used 13 different batch flotation kinetic models and a least-squares method to fit concentration–mean-residence-time data profiles of various rougher flotation circuits, which contain flotation cells in series. This implies that these authors used concentration–residence-time data of mixed-flow reactors in series, and this argumentation is followed, it is clear that eq. (6) may also be applied to predict concentration–mean-residence-time profiles of mixed-flow reactors in series. This means that instead of using $\{k(C)_n, C_n, \tau_n\}$, as for batch reactors, concentration–mean-residence-time data for the mixed-flow reactors in series are used, from which $\{k(C)_n, C_n, \tau_n\}$ may be determined via eq. (3). This set is used in the batch kinetic rate equations (4) and (6), with the time $t$ being substituted by the mean residence time $\tau$.

Simulation of mixed-flow reactors

By including the proposed generalized kinetic rate expression in appropriate mass balance equations for microfluids (Reuter and van Deventer, 1991) and macro fluids, arbitrary mixed-flow reactors may be simulated.

Macrofluid. For a macrofluid with a residence time distribution $E(t)$, the average concentration in the outlet stream of a mixed-flow reactor may be given by (Levenspiel, 1972)

$$C_v = \int_0^\infty E(t) C \, dt$$

(7)

where

$$E(t) = (1/\tau) (V/V_m) \exp \left[-(V/V_m) (t/\tau - V_p/V)\right].$$

$\tau$ is the mean residence time or any other suitable function.

If $C$ in eq. (7) is substituted by eq. (5), eq. (8) is produced:

$$C_v = \sum_{n=0}^\infty \int_{\tau_n}^{\infty} E(t) \exp \left[-k(C)_n(t - \tau_n)\right] dt.$$  

(8)

The algorithm for solving eq. (8) is:

- The set $\{k(C)_n, C_n, \tau_n\}$ is calculated via eq. (3) for a particular batch kinetic data set.
- Initially, $C = C_0$ or 1 (if dimensionless) and $E(0) = 0$ holds and an integration is performed between $t_0 = 0$ and $t_1$ for $k(C)_0$ [method: (i) analytically for simple $E(t)$ functions, (ii) by a Gaussian quadrature (Press et al., 1989) or (iii) by Euler’s method].
- The above step is repeated for all $n$ and summed until the product $E(\tau_n)C_n$ becomes negligible.

If $C$ in eq. (7) is substituted by discrete $C_n$ values produced by eq. (6) for arbitrary $\alpha$ and $\beta$ adjustment factors and the integral sign is substituted by a summation, eq. (9) is produced:

$$C_v = \sum_{n} C_n E(\tau_n) \delta t$$

(9)

where $C_n$ is calculated by eq. (6).

The algorithm for solving eq. (9) is:

- By the application of eq. (6), a set $C_n-\tau_n$ may be calculated for any $\alpha$ and $\beta$.
- The product $C_n E(\tau_n)$ is summed for all $n$ until the product becomes negligible.

In these equations $E(t)$ represents an arbitrary residence time distribution (RTD), which may be a function of dead volume, fraction of plug and/or mixed flow, number of reactors, mean residence time, etc. (Levenspiel, 1972). The parameters in $E(t)$ may be determined by independent tracer tests or may be related to the flow conditions via a neural net. The latter was, however, not done in this paper.

Microfluid. For a single ideal continuous stirred-tank reactor the differential mass balance equation is

$$\frac{dC}{dt} = \frac{1}{\tau} (C_0 - C) - k(C)_n C$$

(10)

where $C_0$ is the concentration in the feed (1 if dimensionless), $C(0) = C_0$ (1 if dimensionless), and $\tau$ is the mean residence time.
A generalized neural-net kinetic rate equation

The linear differential eq. (10) for microfluids may be integrated [since \( k(C) \) is constant for a particular validity range of \( C \)] to produce simple analytic solutions for the respective validity ranges of the generalized kinetic rate equation.

**REACTOR IDENTIFICATION**

In the context of this paper, reactor identification implies establishing by what degree the conditions within the reactor, i.e. reagent levels, \( \text{pH} \), pulp density, temperature, pressure, etc., change the kinetics of the process. The mean residence time and residence time distribution (RTD) are assumed to remain constant. As mentioned previously, the parameters in the RTD can also be related to the flow conditions within the reactors via neural nets.

From Fig. 1 it is clear that the basis of the proposed approach is to determine by what \( \beta \) and \( \alpha \) factors the kinetic data for the reactor to be identified deviate from the pivot kinetic data, for which \( \alpha = 1 \) and \( \beta = 1 \), respectively. By means of a trained neural net these \( \alpha \) and \( \beta \) values are related to the appropriate process conditions within the reactor.

**Batch reactors**

For batch reactors the instantaneous change of the slope \( k(C) \) relative to that of the pivot kinetic data is a continuous indication of \( \alpha \), which can give a continuous indication as to what the process conditions are via a neural net. The factor \( \alpha \) may also be determined from the complete kinetic \( C-t \) data set if this becomes available at the end of the reaction. This could also assist in characterizing the eventual product. The mathematical basis for this approach is given by eq. (11), indicating that \( \alpha \) is an average value over the complete \( C-t \) data set. A time-weighted integration of the slope of a reaction of first order produces the result \( 1/k' \) (min or h) (if \( C \) is dimensionless):

\[
1/k' = \int_0^\infty tk'C dt = \int_0^\infty tk'e^{-k't} dt
\]

which has been generalized to eq. (12) if \( k' \) in eq. (11) is substituted by \( k(C)_n \) and an Euler integration of eq. (11) is performed:

\[
1/k'' = \sum_n \tau_n k(C)_n C_n \delta t.
\]

For \( N \) ideal mixed-flow reactors in series, the above method [eq. (11) or (12)] may also be applied to identify these reactors continuously. In this case, eq. (12) is rewritten as for the set \( \{ k(C)_n, C_n, \tau_n \} \):

\[
1/k'' = \sum_n \tau_n k(C)_n C_n \delta t.
\]

**Mixed-flow reactors**

If the batch simulation approach is used to simulate mixed-flow reactors in series, the above method [eq. (11) or (12)] may also be applied to identify these reactors continuously. In this case, eq. (12) is rewritten as for the set \( \{ k(C)_n, C_n, \tau_n \} \):

\[
1/k'' = \sum_n \tau_n k(C)_n C_n \delta t.
\]

**VALIDATION OF THE GENERALIZED NEURAL-NET KINETIC RATE EQUATION**

The proposed generalized kinetic rate equation will be tested by comparing its performance against that of four kinetic models from the literature.

For the first two examples two kinetic models were taken, viz. (i) a kinetic model for flotation (Dowling et al., 1986) and (ii) a second-order kinetic model (Levenspiel, 1972). These two examples are included to demonstrate the numerical accuracy of the proposed generalized rate equation. For this reason, no neural net was included for these two examples. The second two examples are applications to industrial kinetic data, viz. (i) the leaching of a Zn ore and (ii) the precipitation of Fe\(^{3+} \) ions from Zn-leach liquors.

Note that in spite of each of the four examples representing a totally different kinetic mechanism, the proposed generalized kinetic rate equation could simulate the kinetics very well.
In order to demonstrate the accuracy of the generalized kinetic rate equation, the theoretical results are given in table form. Differences in the theoretical and calculated values are due to errors produced by the approximation.

**Theoretical examples**

Two theoretical examples will be discussed in this section. Note that the accuracy of these examples is a function of $\alpha = 0.1$ and $\beta = 4$ (and $\beta$ is usually estimated by NN).

**Flotation kinetic model.** The flotation kinetic model used to generate the theoretical values for different $R$ and $k'$ (min$^{-1}$) values is (Dowling et al., 1986)

$$ C/C_0 = 1 - R \left( 1 - \left( 1 - e^{-k't}\right) / k't \right). $$

These theoretical values are given in the (a) columns of Table 1 ($C_0 = 1$).

If it is assumed that the data in column 1(a) of Table 1 are the pivot data, i.e. $\beta = 1$ ($R = 1$) and $\alpha = 1$ ($k' = 1$ (min$^{-1}$)), the slopes $k(C)_n$ may be calculated by eq. (3). These results have been summarized in Table 2. These $k(C)_n$ may now be used in conjunction with eq. (6) and the relevant $\alpha$ and $\beta$ values in Table 1 to produce the values given in the columns (b) of Table 1.

The solution algorithm for predicting the data in column 3(b), for example [data in column 3(a) are the theoretical values] is:

**Second-order kinetic model.** The second-order kinetic model which was applied to generate the theoretical values for different $k'$ (min$^{-1}$) and $R = 1$ values is (Levenspiel, 1972)

$$ C/C_0 = 1/(1 + C_0k't). $$

The theoretical values produced by this model are given in columns (a) of Table 3 ($C_0 = 1$). As for the previous example, column 1(a) was selected to be the pivot, from which $k(C)_n$, could be calculated via eq. (3). In this example both an arithmetic as well as a geometric mean for $C_\text{ar}$ was calculated, the resulting set $\{k(C)_n, C_\text{ar}, t_n\}$ being partially summarized in Table 4. $k(C)_n$ could then be used to predict (same algorithm as for the previous example) the values in columns (b) (arithmetic mean) and columns (c) (geometric mean) via eq. (6) and the given $\alpha$ values in Table 3.

In each of the two theoretical examples, a good correspondence is achieved between theoretical kinetic models and the proposed generalized kinetic rate equation.

**Industrial examples**

Both examples discussed in this section were taken from the field of zinc hydrometallurgy. The numerous
complex effects of the process conditions on the kinetics of the described processes make these examples amenable to the simulation approach presented here. A purely theoretical model would not be capable of incorporating all the process effects such as reagent concentrations, temperature, pH, particle size, mixing conditions, etc.

For each of the respective examples a NN is defined, which relates the measured process conditions to the adjustment factors given in Figs 5 and 6, respectively. The testing phase of each trained neural net could not be performed due to the relatively small number of training data sets. This is similar to Venkatasubramanian et al. (1990), who used 12 data sets for a 78 weight neural net. Each neural net in this paper could, however, be trained almost exactly (error \(< 10^{-4}\) ) from the training data set. Graphical representations of the results of the trained neural nets show that plausible generalizations could be produced and that exact predictions can be made for the trained data. However, since the training data sets are small and have not been tested by a test training set, it cannot be stated that the trained neural nets are totally general.

\textbf{Table 4. Summary of the calculated }k(C)\textbf{ values according to eq. (3) and geometric mean). using the data in column }l(a)\textbf{ in Table 3}

<table>
<thead>
<tr>
<th>Column l(a) of Table 3</th>
<th>C range</th>
<th>$k(C)_a$ (arithmetic)</th>
<th>$k(C)_a$ (geometric)</th>
<th>t (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.000-0.833</td>
<td>18.1818</td>
<td>18.2574</td>
<td>0.000-0.010</td>
<td></td>
</tr>
<tr>
<td>0.833-0.714</td>
<td>15.3846</td>
<td>15.4303</td>
<td>0.010-0.020</td>
<td></td>
</tr>
<tr>
<td>0.714-0.625</td>
<td>13.3333</td>
<td>13.3631</td>
<td>0.020-0.030</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>-0.000</td>
<td>0.0000</td>
<td>-\infty</td>
<td></td>
</tr>
</tbody>
</table>
produce the predictions given in Fig. 5 (lines). From Fig. 4 it is clear that the NN permits reasonable extrapolations to other process conditions, although the training temperature lies between 75 and 95°C, and the \([H_2SO_4]\) between 15 and 50 kg/m³.

**Jarosite precipitation.** The example being discussed here is the precipitation of \(Fe^{3+}\) as Na-jarosite (Rastas et al., 1979):

\[
3Fe_2(SO_4)_3 + Na_2SO_4 + 12H_2O = 2NaFe_3(SO_4)_2(OH)_6 + 6H_2SO_4.
\]

The kinetics of this reaction are affected by a variety of factors, which include the temperature, the concentration of the Na-jarosite seed, \([H_2SO_4]\), \([Fe^{3+}]\) and the \([Na^+]\). The modelling of the kinetics of this reaction is complex, a fact which is reflected by the numerous models presented to describe it. Two of these kinetic rate equations are:

\[
dFe^{3+}/dt = -k(T)[Fe^{3+}]^4[Na^+]^b \times [Na-jarosite]^c[H_2SO_4]^d
\]

(Rastas et al., 1979)

or

\[
dFe^{3+}/dt = -\{k_1[Fe^{3+}]^2[Na^+]^{1/2} - k_2[H_2SO_4]^{1/4}\}
\]

(Wang Qian-kun et al., 1985).

The raw kinetic data in Fig. 6 (symbols) were taken from Rastas et al. (1979), who modelled the data with
A generalized neural-net kinetic rate equation

The given semi-empirical kinetic model above (four empirical parameters a–d). In Fig. 6 experimental data of Rastas et al. (1979) and predictions produced by eq. (6) (lines) are compared as a function of the α values produced by the neural net (Fig. 7). The kinetic data for 95°C and [Na-jarosite seed] = 300 kg/m³ are taken to be the pivot conditions. Other pivot conditions are [Fe³⁺] = 30 kg/m³, [Na⁺] = 8 kg/m³ and [Zn²⁺] = 100 kg/m³; the latter two were assumed not to affect the kinetics of the process.

The α and β values in Fig. 7 have been modelled as a function of [Na-jarosite seed] and temperature by using the same feedforward NN (12 weights) given in Fig. 2 ([jarosite seed]/100 replaces [H₂SO₄]/10). As for the previous example the neural net could learn the five data sets {α, [H₂SO₄], [Na-jarosite]} and process values almost exactly (error < 10⁻⁴), although numerous starting weights were used. Figure 7 shows that a good generalization could be achieved, rendering the generalized rate equation general for a wide range of [Na-jarosite] and temperature values.

APPLICATION TO MIXED-FLOW REACTORS

The first two examples discussed in this section are theoretical in nature and demonstrate the application of eqs (8) and (9), respectively. The third and fourth examples demonstrate the application of eq. (6) to a flotation rougher circuit for the recovery of copper sulphide. The application of eq. (10) has already been
demonstrated by Reuter and van Deventer (1991) for an industrial gold carbon-in-pulp circuit and will not be repeated here.

As with the previous examples the theoretical results are given in tabular form in order to illustrate the accuracy of the approximations.

**Theoretical examples**

In this section mixed-flow reactors will be simulated by following a macrofluid approach for a second-order kinetic model and $E(t) = (1/r) \left( \frac{V}{V_m} \right) \exp \left[ - \frac{V}{V_m} (t/t_c - t_p) \right]$. In each case the results of the theoretical solution [eq. (7)] are compared with the results produced by the approximate solution [(8) and (9)].

The given theoretical solution for the second-order process for $E(t) = (1/r) \exp(-t/t_c)$ in Tables 5 and 6 is calculated via $(1/k_r) e^{-1/k_r} E(t)$ (Levenspiel, 1972; $E(t)$ = exponential integral).

For both the theoretical examples an adequate correspondence was obtained. The accuracy of the results is a function of $\delta t = t_{c+1} - t_c$ in eq. (3) and may, hence, be manipulated as required.

**Application of eq. (8).** For this example a set of $k(C)_n$ was calculated for each of the first- and second-order kinetic models given in Table 5 ($C_0 = 1$) by using eq. (3). These sets were then subsequently used in eq. (8) to produce values for $C_n$ for the given $E(t)$ functions. The integration of eqs (7) and (8) was performed by a 15-point Gaussian quadrature (Press et al., 1989).

The method for solving eq. (8) for $C = e^{k(t)}$, $k(C)_n = 1$ (for all $n$), $E(t) = 0.5 e^{-0.5 t}$ ($r = 2$, $V_m = 1$, $V_p = 0$) and $\delta t = 1$ would be

$$C_n = 1 \sum_0^1 0.5 e^{-0.5 t_{e+1}} e^{-1.0 t_{e}} dt$$

$$+ 0.368 \sum_0^2 0.5 e^{-0.5 t_{e+1}} e^{-1.0 t_{e+1}} dt + \ldots$$

$= 1/3$.

The same summation was performed for all the other entries in Table 5, with the only difference being the $k(C)_n$ and $\delta t$ values for each respective application.

**Application of eq. (9).** In contrast to the previous example, the kinetic $C$–$t$ data produced by the kinetic model $1/(1 + 20t)$ were taken to be the pivot data, from which a set of slopes $k(C)_n$ could be calculated by the application of eq. (3). By applying eq. (6) and the given adjustment factors $a$, $C_n$ values could be cal-

<table>
<thead>
<tr>
<th>Equation and parameters</th>
<th>$e^{-t}$</th>
<th>$1/(1 + t)$</th>
<th>$1/(1 + 5t)$</th>
<th>$1/(1 + 10t)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t = 1$ (theoretical)</td>
<td>0.3679</td>
<td>0.5000</td>
<td>0.1667</td>
<td>0.0909</td>
</tr>
<tr>
<td>$t = 1$ (approximate)</td>
<td>0.3679</td>
<td>0.5000</td>
<td>0.1667</td>
<td>0.0909</td>
</tr>
<tr>
<td>Equation (7) $t = 2$, $V_m = 1.0, V_p = 0.0$</td>
<td>0.3333</td>
<td>0.4614</td>
<td>0.2002</td>
<td>0.1261</td>
</tr>
<tr>
<td>Equation (8) $t = 2$, $V_m = 1.0, V_p = 0.0$</td>
<td>0.3333</td>
<td>0.4616</td>
<td>0.2025</td>
<td>0.1320</td>
</tr>
<tr>
<td>Theoretical</td>
<td>0.3333</td>
<td>0.4617</td>
<td>0.2014</td>
<td>0.1297</td>
</tr>
<tr>
<td>Equation (7) $t = 2$, $V_m = 0.8, V_p = 0.2$</td>
<td>0.2578</td>
<td>0.4075</td>
<td>0.1357</td>
<td>0.0748</td>
</tr>
<tr>
<td>Equation (8) $t = 2$, $V_m = 0.8, V_p = 0.2$</td>
<td>0.2578</td>
<td>0.4075</td>
<td>0.1358</td>
<td>0.0749</td>
</tr>
<tr>
<td>Equation (7) $t = 2$, $V_m = 0.5, V_p = 0.5$</td>
<td>0.1839</td>
<td>0.3613</td>
<td>0.1052</td>
<td>0.0559</td>
</tr>
<tr>
<td>Equation (8) $t = 2$, $V_m = 0.5, V_p = 0.5$</td>
<td>0.1839</td>
<td>0.3614</td>
<td>0.1052</td>
<td>0.0559</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Equation and parameters</th>
<th>$1/(1 + 20t)$</th>
<th>$1/(1 + 5t)$</th>
<th>$1/(1 + t)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_0 = 1, \beta = 1, \delta t = 0.01$</td>
<td>$\alpha = 1$ (pivot)</td>
<td>$\alpha = 0.25$</td>
<td>$\alpha = 0.05$</td>
</tr>
<tr>
<td>$t = 1, \text{(theoretical)}$</td>
<td>0.0076</td>
<td>0.1667</td>
<td>0.5000</td>
</tr>
<tr>
<td>$t = 1, \text{(approximate)}$</td>
<td>0.0076</td>
<td>0.1658</td>
<td>0.4999</td>
</tr>
<tr>
<td>Equation (7) $t = 2$, $V_m = 1.0, V_p = 0.0$</td>
<td>0.00786</td>
<td>0.2012</td>
<td>0.4609</td>
</tr>
<tr>
<td>Equation (9) $t = 2$, $V_m = 1.0, V_p = 0.0$</td>
<td>0.00790</td>
<td>0.1986</td>
<td>0.4569</td>
</tr>
<tr>
<td>Theoretical</td>
<td>0.00803</td>
<td>0.2014</td>
<td>0.4617</td>
</tr>
<tr>
<td>Equation (7) $t = 2$, $V_m = 0.8, V_p = 0.2$</td>
<td>0.00395</td>
<td>0.1357</td>
<td>0.4073</td>
</tr>
<tr>
<td>Equation (9) $t = 2$, $V_m = 0.8, V_p = 0.2$</td>
<td>0.00390</td>
<td>0.1349</td>
<td>0.4065</td>
</tr>
<tr>
<td>Equation (7) $t = 2$, $V_m = 0.5, V_p = 0.5$</td>
<td>0.00288</td>
<td>0.1052</td>
<td>0.3613</td>
</tr>
<tr>
<td>Equation (9) $t = 2$, $V_m = 0.5, V_p = 0.5$</td>
<td>0.00286</td>
<td>0.1046</td>
<td>0.3610</td>
</tr>
</tbody>
</table>
culated for the other two second-order kinetic models given in Table 6, which could then be used with the appropriate \( E(t) \) functions to produce \( C_{av} \) values. The integration for eq. (7) was performed by a 15-point Gaussian quadrature (Press et al., 1989) and the integration of eq. (9) by Euler's method.

The method for solving eq. (9) for \( \alpha = 1/(1 + 20t) \) is \( [C_{av} - t_a] \) values from Table 3 column 1(b), \( E(t) = 0.5e^{-0.5t} \) (\( \alpha = 2 \), \( V_m = 1 \), \( V_p = 0 \)) and \( \delta t = 0.01 \)

\[
C_{av} = [E(0).1 + E(0.01) \cdot 0.833 + E(0.02) \cdot 0.714 + E(0.03) \cdot 0.625 + \ldots \cdot 0.01]
= 0.0790
\]
or, for \( \alpha = 0.5 \) \( [C_{av} - t_a] \) values from Table 3 column 2(b), \( E(t) = 0.5e^{-0.5t} \) (\( \alpha = 2 \), \( V_m = 1 \), \( V_p = 0 \)) and \( \delta t = 0.01 \).

\[
C_{av} = [E(0).1 + E(0.01) \cdot 0.913 + E(0.02) \cdot 0.833 + E(0.03) \cdot 0.771 + \ldots \cdot 0.01]
= 0.1986
\]

All the entries in Table 6 were determined on this basis using appropriate \( C_{av} - t_a \) kinetic data determined by eq. (6) as given in Table 3.

**Practical examples**

Two practical examples for mixed-flow reactors will be discussed in this section. For both these examples eq. (6) was used, in which \( t \) was replaced by \( t \). This means that the \( C \)-residence-time data were treated as batch kinetic data similar to Dowling et al. (1986). Data for both these examples were taken from Dowling et al. (1986).

**Tennessee copper rougher circuit.** As may be seen from Fig. 8 the recovery–mean-residence-time data (symbol) for the Tennessee rougher circuit could be simulated exactly (dark line) for which \( \alpha = 1 \) and \( \beta = 1 \). This was not possible by the least-squares fitting of concentration–retention-time data for all 13 tested batch flotation kinetic models as applied by Dowling et al. (1986). Even the induction period could be predicted exactly by the proposed kinetic model, an aspect which could not be modelled by any of the available batch kinetic flotation models. The pivot process conditions for these pivot kinetic data are [hydrated lime] = 0.915 kg/t, [cyanide] = 0.107 kg/t, [sodium silicate] = 0.325 kg/t, [Na ethyl xanthate] = 0.024 kg/t and [Guar gum] = 0.016 kg/t.

A sensitivity analysis was subsequently performed by using eq. (6) for the pivot data, showing the concentration profile for different \( \alpha \) and \( \beta \) factors.

**Nchanga sulphide rougher circuit.** As opposed to the previous example a neural net (Fig. 9) was trained that relates \( \alpha \) and \( \beta \) to particle size, which was subsequently applied to predict the concentration profiles in the Nchanga rougher circuit. Similar to Fig. 2 the neural net has two inputs, three nodes in the hidden layer and one output (12 weights). The net could be trained from any initial weight set.

The reagent pivot conditions, i.e. [sodium isopropyl xanthate], [Frother] and [NaHS], were assumed to remain at a constant level throughout. The pivot particle size was taken to be +150 mesh, which is given the label 1. The other particle size fractions are -150 + 200 mesh: label 2 and -200 + 325 mesh: label 3, respectively.

This net was then applied to predict \( \alpha \) and \( \beta \) adjustment factors for the different particle size fractions, which were then used in eq. (6) to predict the recovery profiles (Fig. 10) for the different particle size fractions for the eight cells in the Nchanga rougher circuit. From Fig. 10 it is clear that a combination of neural–net and generalized kinetic rate equation could predict the recovery profiles for this circuit very well. The

---

**Fig. 8.** Recovery profiles vs mean residence time for copper in the Tennessee rougher circuit.
modelling of the recovery profile as a function of particle size is really only possible by empirical methods, which the proposed neural-net approach could accomplish with ease.

**REACTOR IDENTIFICATION**

Reactor identification in this paper is taken to imply the detection of faults within the respective unit operations, i.e. process conditions such as reagent levels, densities, pressures, particles size, temperature, etc., within the metallurgical reactors. Reactor identification will be discussed under two headings, viz. identification in batch and mixed-flow (e.g. CSTRs) reactors, respectively.

For identification purposes a NN for each process condition as a function of $\alpha$ and $\beta$ is defined. This implies that a parallel architecture of NNs is proposed, as depicted by Fig. 11. This architecture makes it possible to expand the neural net without affecting the other neural nets (Watanabe et al., 1989).

Identification of the process conditions within a batch reactor

In order to illustrate the application of the proposed identification algorithm, a theoretical as well as a practical example will be discussed.

Theoretical example. Consider the pivot kinetic data in column 1(a) and the data to be identified, in...
The time-weighted slope of 1.0781 (min) is calculated via eq. (13) for the pivot process conditions (+150 mesh) in Fig. (10) (bottom curve). For the top two curves β is equal to 1.23 (0.86/0.7) and 1.37 (0.96/0.7), respectively.

Identification of the process conditions within mixed-flow reactors

Two methods will be discussed in this section. In both the cases the same trained neural net (two inputs, three hidden nodes, one output: 12 weights) was used (applied) which relates particle size to α and β. The three sets for [particle size, α, β] could be learnt exactly by the neural net.

Identification of the particle size in the Nchanga rougher circuit: method 1. This method applies eq. (13) (as for batch processes) to describe the kinetics of the flotation circuit as a whole, i.e. C(r) vs τ data are applied. Hence the identification procedure is identical to the one discussed for the batch reactors above.

The algorithm for identifying the particle size for the top two profiles in Fig. 10 is:

---

In this example it is demonstrated how k(C) values may be used to determine α directly from the batch data. For the pivot data k(C) = 0.435 and for the data to be identified k(C) = 0.32, which implies α = 0.74. From Fig. 12 it is clear that the neural net would suggest a temperature of approximately 85°C and [H$_2$SO$_4$] = 30 kg/m$^3$, which are the correct process conditions.

As may be seen from Fig. 12, α could be related to the learnt process conditions over a wide range of α values (β = 1 throughout), pointing to the generalization possible with this trained neural net.

Practical example. In order to illustrate the proposed identification procedure on a practical example, the zinc ferrite process conditions and associated α and β adjustment factors in Fig. 5 were used as training sets for the proposed identification NNs. A NN with two hidden layers, each containing three nodes, could be trained almost exactly (error < 10$^{-4}$) from any initial weight set to relate the five data sets {[H$_2$SO$_4$], temperature, α}. The number of weights for this example are 18 for each NN.
The iterative application of eq. (9) this value may be calculated for each of the respective profiles, which are 1.38 and 1.53.

The learnt neural net establishes what particle size produces the sets \{1.38, 1.23\} and \{1.53, 1.37\} for \{\alpha, \beta\}, according to the architecture given in Fig. 11.

From Fig. 13 it is clear that the first set corresponds to a particle size fraction 2 (150 + 200 mesh) and the second to the size fraction 3 (200 + 325 mesh). For identification purposes this example would suggest that if the overall recovery decreases, the particle size distribution of the feed is too coarse. It is interesting to note that, in spite of only three training sets, the neural net could predict that for finer particle sizes the recovery increases and that in order to maintain a recovery the rate has to increase if the particle size becomes coarser.

Identification of the particle size in the Nchanga rougher circuit: method 2. Whereas the batch identification procedure as applied to the previous example applies eq. (13) to determine an estimate of \(\alpha\), eqs (9) and (14) are applied as the basis of identification procedure for method 2.

If the particle size for the top curve in Fig. (10) is to be identified the following procedure is followed:

- The final recoveries for the pivot concentration profile and profile to be identified are easily determined to be 70 and 96%, respectively (\(\beta = 1.37\)).
- Equation (14) may be written as an approximation for these profiles:

\[
C_{av,\, pivot} = 0.70 - 0.70/[1 + (kt)_{pivot}]^\gamma
\]

\[
C_{av,\, identify} = 0.96 - 0.96/[1 + (kt)_{identify}]^\gamma
\]

from which \((kt)_{pivot} = 0.972\) and \((kt)_{identify} = 1.449\) may be calculated from the \(C_{av}\) values, which are 0.52 and 0.8 at cell no. 2, respectively.

- Assuming \(\tau\) to be the same for both the profiles, an \(\alpha\) value of 1.49 may be produced.
- By the iterative application of eq. (9) this value may 1.38 be refined, if the batch kinetic data for this process were available.
- From \(\beta = 1.37\) and the refined \(\alpha\) value, an estimate may be made for the particle size. For \(\alpha = 1.49\) (first estimate) and \(\beta = 1.37\) a particle size fraction 3 could be suggested by the appropriate neural net (Fig. 13) as a first estimate.

**DISCUSSION AND SIGNIFICANCE**

A simple generalized approach was presented for the simulation and identification of batch and mixed-flow mineral processing and metallurgical reactors.

The proposed generalized kinetic rate equation is process-independent and has been applied to simulate adsorption and reduction processes (Reuter and van Deventer, 1991), and flotation, leaching and precipitation processes, as shown in this paper. The parameters of the rate equation are related to the process conditions of the process under consideration by a trained neural net.

It is also shown how the kinetic rate equation and a trained neural net may be implemented to perform process-independent reactor identification using kinetic data from mineral processing and metallurgical processes and associated process data such as pH, reagent levels, temperature, pulp density, particle size, etc.

The generality, compact nature and simplicity of the proposed kinetic rate equation cum trained-
A generalized neural-net kinetic rate equation

A neural-net approach permits the development of a generalized process-independent and industrial-orientated reactor simulation and identification tool.

**NOTATION**

- \( C, C_n \): Concentration in a batch or mixed-flow reactor and discrete concentration at \( t_0 \), dimensionless, ppm
- \( C_{sv} \): Average concentration from a mixed-flow reactor, dimensionless, ppm
- \( C_0 \): Initial concentration, dimensionless, ppm
- \( E(t) \): Residence time distribution \( f(V_p, V_m, V_d, \tau, N) \), h\(^{-1}\)
- \( k' \): Rate constant in theoretical kinetic models, min\(^{-1}\), h\(^{-1}\)
- \( k'' \): Mean rate constant calculated by eqs (13) and (14), min\(^{-1}\), h\(^{-1}\)
- \( k(C)_n \): Discrete rate variable as a function of \( C \), min\(^{-1}\), h\(^{-1}\). Valid for \( C_{n+1} < C \leq C_n \)
- \( N \): Number of stages for mixed-flow reactors in series, dimensionless
- \( R \): Final recovery
- \( R_{piv} \): Final recovery of pivot kinetic data
- \( t \): Time, min, h
- \( t_0 \): Time corresponding to \( C_n \), min, h
- \( V, V_m, V_p, V_d \): Total, mixed flow, plug flow and dead volume, respectively, used in \( E(t) \), m\(^3\)

**Greek letters**

- \( \alpha, \beta \): Adjustment factors > 0: function of neural net, dimensionless
- \( \delta t \): Time increment, min, h
- \( \tau \): Mean residence time ( = reactor volume/volumetric flow rate), min, h

**REFERENCES**


THE DYNAMIC MODELLING OF ILL-DEFINED PROCESSING OPERATIONS USING CONNECTIONIST NETWORKS

T. J. VAN DER WALT and J. S. J. VAN DEVENTER
Department of Metallurgical Engineering, University of Stellenbosch, Stellenbosch 7600, South Africa

and

E. BARNARD
Department of Computer & Electronics Engineering, University of Pretoria, Pretoria 0001, South Africa

(First received 15 December 1991; accepted in revised form 14 October 1992)

Abstract—This paper proposes a new methodology to model ill-defined processing operations using neural nets (NNs). A process with many variables (large dimensionality) cannot be modelled adequately if limited process data are available. This problem of multidimensionality is addressed and an approach suggested to reduce the dimensionality using NNs. An NN is trained on process data for the global variable space, whereafter the first-order partial derivatives of the process are estimated with the NN and used to perform perturbation analyses. As a result, the variable space can be subdivided into subspaces with reduced dimensionality. The final product of this modelling methodology is a combined model of phenomenological expressions and NNs. The model can be incorporated successfully in a dynamic simulator of the process. A new approach to conduct modelling on the basis of continuous data collected directly from an industrial processing unit is also proposed. The modelling methodology is applied to a typical processing operation, i.e. the carbon-in-leach (CIL) process.

INTRODUCTION

In the chemical and metallurgical industry, a dynamic process model is usually developed by performing material, momentum and energy balances on the system, and differential equations are the standard language for expressing such models. Such equations can be formulated only if the chemistry and mechanics of the process are reasonably well-understood and the model parameters are known. This implies that a "deep" knowledge of the physical principles of the process should be available. Although such phenomenological models have the ability to extrapolate beyond the information associated with the modelling data, it is in most cases not possible to formulate such models, owing to a lack of understanding of the process. This places an obligation on the modelling engineer to construct an adequate model to be fully representative of an entire ill-defined system for which a phenomenological model cannot be formulated yet.

The search for a modelling technique with the ability to describe adequately an arbitrary poorly understood process is still in progress. Such a technique should be able to construct a mapping between process variables and functions so that these ill-defined relational mappings can be implemented as reliable models. Most importantly, it should have the ability to learn relations within those process domains which cannot readily be modelled by means of conventional techniques. In the search for modelling techniques to address this problem, various techniques have been investigated such as empirical modelling, traditional regression analysis, as well as knowledge based systems (KBS) or expert systems.

Some recent work by Reuter et al. (1991) showed how KBS can be applied to simulate CIP (carbon-in-pulp) and CIL (carbon-in-leach) circuits. From batch reactor simulations, they constructed system variable profiles (such as gold concentration profiles) which could be coupled with the dynamics of the adsorption and leaching processes through a data base. Although KBS can be powerful modelling tools, constructing them is usually time-consuming and expensive.

Connectionist networks (neural nets) are mathematical regression tools which seem to be very promising for a wide range of modelling problems. Although the field is still in its infancy, encouraging results of various applications have been published to date. Most research in neural nets (NNs) has focused on developing classifiers and tools for recognition. NNs have also been used in the process industry in this regard, and substantial work is in progress to use NNs for process control, fault diagnosis and optical classifiers.

Although process modelling forms an integral part of process engineering, little work has been done to employ NNs for modelling purposes. The main reason for this is the problem caused by high-dimensional variable spaces (many process variables determine the dynamics of a process), which are characteristic of process operations in general. This "problem of dimensionality" arises with all modelling techniques,
but becomes even worse if an "intelligent interpolator" such as a NN is to be used. In the present paper this phenomenon is defined and a practical methodology proposed to alleviate it.

It will be shown how process identification can be conducted by means of an NN by analysing data which are representative of a process. The NN sorts the data and forms a "data base" that can be employed to identify some features of a process. Firstly, the NN is employed to perform simple perturbation analyses in order to quantify the influences of the separate variables within different domains of the variable space. By doing so, less influential variables can be eliminated from a specific domain of the variable space and the dimensionality reduced accordingly. In a next step, the "data base" NN identifies mathematical relations, which in turn can be described through mathematical expressions, thus further reducing the "ill-defined" dimensionality. This results in a number of separate variable spaces or subspaces, each one containing a function surface within the subspace of reduced dimensionality. Some variables within a specific subspace will be related to the function surface through mathematical expressions. The relations between the function and the remaining dimensions are ill-defined and can be mapped by using an NN. The final product is a combination of variable subspaces (with an NN to model the process within each), as well as a "global" NN which indicates which submodel should be used.

The CIL process will be used here to illustrate the proposed methodology for NN modelling. This process is a typical mineral processing operation used for gold recovery and is still poorly understood and, therefore, ill-defined. If an adequate simulation model for a weakly-defined process such as the CIL process can be formulated, it can be used in off-line control, where effects of changes in the control strategy of the plant can be predicted. A new approach to employ continuous operating data in order to conduct process identification is also proposed.

THE PROBLEM OF LARGE DIMENSIONALITY

The phenomenon of large dimensionality is an obstacle in the way of effective modelling and is often referred to as the curse of dimensionality (Bellman, 1961). It can be defined synoptically as follows: consider a function \( f \) which is defined on an \( m \)-dimensional variable space \( R^m \). Let \( y_j \) be the average number of data points necessary within the \( j \)th dimension to represent the whole system sufficiently. Then the minimum number of data points needed \( (P) \) will be in the order of

\[
P = \prod_{j=1}^{m} y_j.
\]  

(1)

This shows that the minimum amount of information (such as process data points) required to model a process adequately increases exponentially with a rise in the number of process variables. Inversely formulated, the addition of every single dimension will cause an exponential decay in information contained by a specific set of information for a process. In some cases it is costly and time-consuming to collect data from an industrial system or process. If a modelling technique which successfully addresses the problem of high dimensionality can be developed, the required number of measurements can be reduced accordingly. In other cases it may even be impossible to expand an existing data set, so that the optimal amount of information should be gained from such a set of data.

Several methods have been suggested recently by different authors to improve function approximation in high-dimensional variable spaces. The theory of regularization networks which includes task-dependent clustering has been described by Poggio and Girosi (1990). Mavrovouniotis (1990) proposed alternative NN structures that contain much fewer connection weights than a three-layered perceptron and are organized in a hierarchical fashion. Cherkassky et al. (1991) developed a modification of Kohonen's self-organizing maps which they called constrained topological mapping (CTM). They compared the CTM algorithm to backpropagation NNs and conventional approaches to regression, such as projection pursuit. Their comparisons demonstrated general superiority of the proposed CTM algorithm, both in terms of generalization (interpolation) and computational speed. Although the advantages gained in computation were quite remarkable, this approach offers only a slight improvement over backpropagation NNs if both regression models are implemented in a straightforward fashion. The CTM algorithm will not be able to perform adequate function approximation in variable spaces of most chemical and metallurgical processes where available process data are limited and dimensionality is usually quite high, if implemented without some external form of dimensionality reduction.

Cherkassky et al. (1991) stated that there are mainly three approaches to non-parametric multivariate regression analysis used by statisticians. The first approach, additive models, assumes that functions can be approximated as a linear combination of several univariate functions. However, this approach may neglect some non-linear interactions among the variables. Projection pursuit (PP) regression as a second approach, approximates a function by a sum of non-linear functions of linear combinations of the variables. Although PP performs well in high-dimensional space, it is poorly suited to deal with highly non-linear function surfaces. A third approach, recursive partitioning (RP), partitions a variable space of independent variables into disjoint subspaces. The function is then approximated through spline fitting in each subspace. The performance of this technique for functions with complex non-linear interactions among many variables is not known.

The technique of local variable selection provides a powerful tool for function approximation in high-
Dynamic modelling of ill-defined processing operations

dimensional spaces. Several statistical methods make use of this approach, including CART (Breiman et al., 1984), MARS (Friedman, 1991a, b) ID3 (Quinlin, 1983; Schlittwimmer and Fisher, 1986; Sun et al., 1988) and C4 (Quinlin, 1987). Algorithms such as GMDH (Ivakhnenko, 1971; Ikeda et al., 1976; Barron et al., 1984) and SONN (Tenorio and Lee, 1989) use similar principles to approximate functions. Most of these algorithms use tree structures to portray the selective and sequential incorporation of the input variables. Sanger (1991a, b) proposed a tree-structured adaptive network. The learning procedure of this network compiles a tree, the structure of which depends upon both the input data and the function to be approximated. On a related note, various authors in the NN field such as Mozer and Smolensky (1989) and Kamin (1990) proposed methods of clustering by pruning less significant hidden nodes in order to simplify the NN structure with resulting improvement in generalization properties.

A further trend in the development of input–output models for non-linear systems is the use of non-linear time-series techniques, such as the non-linear auto-regressive moving average with exogenous inputs model (NARMAX) of Leontaritis and Billings (1985). In general, NARMAX models consist of polynomials which include linear and non-linear terms combining the inputs, outputs and past errors. A special case of the NARMAX model, the non-linear autoregressive model (NAR) was recently proposed by Raich et al. (1991).

van der Walt et al. (1991) proposed a new modelling methodology which also addresses the problem of dimensionality in a practical way. This paper extends and refines this methodology further. It should be emphasized that this approach deals with external and localized reduction of dimensionality, while most other authors have addressed the problem in an internal fashion by reducing the complexity of the model itself without investigating the external features of the function to be approximated.

MODELLING ON THE BASIS OF CONTINUOUS DATA

In practice, a model compiled on batch data needs, almost without exception, to be adjusted considerably if it is to be incorporated in a simulator for a continuous process. Hence, process identification should preferably be conducted on the basis of continuous data.

The dynamics of a continuous reactor can be described by performing material, momentum and energy balances over the reactor. Equation (2) represents a typical material balance equation for a substance $X$ in a CSTR:

$$\frac{dX}{dt} = X_{in} - X_{out} + f. \quad (2)$$

The $f$ term represents the reaction(s) which take place within the reactor and, if $f$ can be modelled, it is possible to describe the dynamics of the reactor completely. This reaction term can be expressed in terms of the other three terms of the latter equation:

$$f = \tau \frac{dX}{dt} - X_{in} + X_{out}. \quad (3)$$

The terms on the right-hand side of eq. (3) should be measurable so that $f$ can be calculated under various circumstances with this expression. At the same instant when $f$ is determined, other process variables which may have an effect on $f$ are evaluated as well and combined with $f$ to form a process data point. These data points can be used by the process engineer to develop a model for $f$. In the case of an NN modelling approach, the data set consisting of the above-mentioned data points is employed as a "training set" by an NN training program to learn the relations between the different process variables and the reaction function.

NEURAL NETS

NNs are highly flexible non-linear regression analysis equations originating from attempts to capture a number of facts related to brain function and structure. They have certain attractive features such as a high degree of parallelism and powerful associative memory properties, and can in some cases rapidly compute near-optimal solutions to highly constrained optimization problems. The influential universal approximation theorem of Hornik et al. (1989) states that multilayered feedforward networks with only one hidden layer are capable of approximating any measurable, bounded function with a finite number of discontinuities, from one finite dimensional space to another with any degree of accuracy, provided sufficient hidden nodes are available. This implies that an NN can fit any curve between process variables and functions. NNs thus seem to be promising tools for modelling.

As yet, little research has been done on applications of NNs in the chemical and metallurgical environment. Research in this regard has focused mainly on relatively simple applications where little information is required. The reason for this can be the "curse of dimensionality". Applications already investigated include fault diagnosis in chemical processes, with work done amongst others by Watanabe et al. (1989) and Hoskins et al. (1991). Kramer and Leonard (1990) used NNs for fault diagnosis purposes and preferred traditional classifiers to NNs.

The application of NNs for modelling purposes still has to be exploited to a large extent. Bhat et al. (1990) used a three-layered NN to model non-linear chemical processes. They considered whole chemical systems as "black-boxes" without distinguishing between known features of the system and obscure areas within the system. It should be more effective to use NNs only within those areas where a mathematical model is not suitable. The problems caused by high dimensionalities have not been addressed either.
Architecture of a typical NN

Various NN topologies, neuron transfer functions and training methodologies have been proposed. Lippmann (1987) gave an introduction to some of these structures, such as Hopfield nets, Hamming nets, Carpenter/Grossberg classifiers, perceptrons, multilayered perceptrons and Kohonen self-organizing feature maps. Recent success in connectionist network research is mainly attributed to the achievements in multilayered perceptrons with backpropagation training algorithms. The NN used for the purposes of this paper is a three-layered perceptron, the architecture of which will now be discussed.

The three-layered perceptron consists of three layers of nodes, viz. (1) an input layer containing $m_1$ nodes (including a bias node), (2) a single hidden layer ($m_2$ nodes) and (3) an output layer with $m_3$ nodes. This NN is illustrated in Fig. 1. Nodes of adjacent layers are connected and these connections quantified with weights. The weight matrix of a trained NN contains the information about a process under investigation. Each node is characterized by an activation function $\phi$. During the first step of a feedforward calculation of the NN, the values of the process variables as contained in a training data point, are fed to the input nodes. Each process variable is associated with a different input node. An additional input node, the bias node with a constant value of 1, provides additional degrees of freedom which enable a fitted curve to be moved up or down in the variable space. Each node $n$ in a layer $z$ has two important features, namely the node input $x_{z,n}$ and its activity $y_{z,n}$. The activity $y_{z,n}$ of a specific node is a function of its input $x_{z,n}$. Equation (4) gives a notation of such an activation function:

$$y_{z,n} = \phi_{z}(x_{z,n}).$$

![Fig. 1. Architecture and nomenclature of a three-layered perceptron.](https://scholar.sun.ac.za)

The input to each hidden node ($z = 2$) is calculated with eq. (5). It equals the sum of the products $w_{ij}y_{1,i}$, where $w_{ij}$ is the connection weight between input node $i$ and the hidden node $j$, and $y_{1,i}$ is the activity of input node $i$. The bias node activity and its corresponding weight are also taken into account.

$$x_{2,j} = \sum_{i=1}^{m_1} w_{ij}y_{1,i} + T_j.$$

The input to node $k$ in the output layer ($z = 3$) has a similar form, namely,

$$x_{3,k} = \sum_{j=1}^{m_2} w_{jk}y_{2,j}.$$

Each node in the output layer is associated with a function to be approximated. The output of the NN is simply the activities of the nodes in this layer.

The activation functions of nodes are chosen according to the requirements of the application. The three-layered perceptron used for the purposes of this paper contained linear nodes [eq. (7)] in the input and output layers, as well as sigmoidal hidden nodes [eq. (8)]:

$$\phi_{l}(x_{z,n}) = x_{z,n}$$

$$\phi_{s}(x_{z,n}) = \frac{1}{1 + e^{-x_{z,n}}}.$$

Training an NN

The training process of NNs is an important part of NN research. It usually involves optimizing a weight matrix of an NN so that the NN will imitate the mapping between a number of system variables and functions. The backpropagation algorithm, as described by Rumelhart et al. (1986), is used by our NN training program, which incorporates the conjugate gradient (CG) optimization algorithm with restart procedures of Powell (1977) as optimization method. During each training iteration all weights of the NN are adjusted in such a way as to decrease the value of an objective function. Most NN training programs employ the LMS error function as an objective function and the net used here is no exception.

Two main steps can be identified during each iteration of training, viz.

1. determining the gradients of the error function with respect to each weight in the network by presenting all training examples once to the net, and
2. adjusting the weight matrix by means of an optimization method.

The first step in calculating the error-weight gradients (step 1) is performed in the following substeps, for each training example, respectively:

(i) A training example is fed to the input layer. The activities of all nodes in the net are calculated during a feedforward step as described in previous paragraphs.
(ii) The value of the error function at each output node is calculated.
(iii) During a backward sweep the gradients with respect to all weights are determined via the procedure of backpropagation.

It is very important that an effective optimization method be used during step 2. The CG method is usually able to locate a minimum of a multivariate function much faster than the steepest descent algorithm with momentum which is customarily employed with backpropagation. Furthermore, its memory usage is of the order of \( N \) (number of weights) locations. Also important to note is that the CG technique eliminates the choice of critical parameters, such as the learning rate and momentum parameters of the steepest descent algorithm. On the other hand, like all gradient descent optimization techniques, the CG algorithm can converge into local minima, which is a major drawback.

CONSTRUCTING AN NN PHENOMENOLOGICAL MODEL

The procedure followed to compile an NN phenomenological model is conducted through a number of steps. In the first instance, a global NN is trained with all process data points. This global NN is used to divide the process variable space into subspaces. This division is done by performing a perturbation analysis on the training data set, whereafter these same perturbation results are used to identify less-influential variables within every individual subspace. These non-significant variables can then be eliminated. The dimensionality of the various subspaces is, thus, reduced and the population density of the training data points increased accordingly. During the next step, an NN is trained for each subspace on data points containing only the values for the significant variables within a specific subspace. Such NNs have fewer input nodes and should, therefore, be able to perform improved curve fittings due to a reduction in the degrees of freedom. Hence, the generalization or interpolation properties of the NNs should be better as well. The trained NN of each subspace is now employed to identify simple mathematical relations between a function and certain variables within a subspace. The remaining dimensions with unknown relations to the function can now be mapped during a further step by an even simpler NN. The result will be an empirical model consisting of a combination of phenomenological expressions and small NNs.

Perturbation analysis

The approach to perturbation analysis proposed here is similar to conventional sensitivity analysis [examples Wong and Rabitz (1991) and O’Sullivan (1991)], but goes further than those methods since sensitivities local to specific areas are eventually used, as will be explained below.

The partial derivative of a function with respect to a specific dimension is indicative of the relative influence of the corresponding variable on the function. If all the first-order partial derivatives of a multivariate function are known at a specific variable space coordinate, these derivatives can be used to compare the relative influences of the different variables on the function at that location in variable space. Less influential variables can then be eliminated locally from the variable space, resulting in subregions with reduced dimensionality.

A description follows which shows how the first-order partial derivatives of a three-layered perceptron can be calculated analytically.

The first-order partial derivatives of an NN. The notation used below has been defined in Fig. 1. The first-order partial derivative of NN output \( y_{2,k} \) with respect to NN input \( x_{1,i} \) is

\[
\frac{\partial y_{2,k}}{\partial x_{1,i}} = \sum_{j=1}^{n} \frac{\partial y_{2,k}}{\partial y_{2,j}} \frac{\partial y_{2,j}}{\partial y_{1,i}} \frac{\partial y_{1,i}}{\partial x_{1,i}}. \tag{9}
\]

Substituting \( x_{2,n} \) in eq. (4) with \( x_{2,j} \) of eq. (5), the activity of a hidden node \( j \) can be expressed in the form

\[
y_{2,j} = \phi_{2}[x_{2,j}(y_{1,1}, \ldots, y_{1,i}, \ldots)], \tag{10}
\]

Applying the chain rule of differentiation to eq. (10) leads to the expression

\[
\frac{\partial y_{2,j}}{\partial y_{1,i}} = \frac{\partial \phi_{2}}{\partial x_{2,j}} \frac{\partial x_{2,j}}{\partial y_{1,i}}. \tag{11}
\]

From eq. (5) it follows that:

\[
\frac{\partial x_{2,j}}{\partial y_{1,i}} = w_{ij}. \tag{12}
\]

If the latter two equations are combined, eq. (11) can be written as

\[
\frac{\partial y_{2,j}}{\partial y_{1,i}} = \frac{\partial \phi_{2}}{\partial x_{2,j}} w_{ij}. \tag{13}
\]

The same procedure can be conducted to compute the partial first-order derivatives of the output node activities with respect to the activities of the hidden nodes. The result is

\[
\frac{\partial y_{3,k}}{\partial y_{2,j}} = \frac{\partial \phi_{3}}{\partial x_{3,k}} w_{jk}. \tag{14}
\]

The inputs to the nodes in the input layer \( (x_{1,i}) \) are simply the input variable values, so that

\[
\frac{\partial y_{1,i}}{\partial x_{1,i}} = \frac{\partial \phi_{1}}{\partial x_{1,i}}. \tag{15}
\]

The first-order partial derivative of a three-layered perceptron can now be finalized by combining eqs (13)–(15) with eq. (9):

\[
\frac{\partial y_{2,k}}{\partial x_{1,i}} = \sum_{j=1}^{n} \frac{\partial \phi_{3}}{\partial x_{3,k}} \frac{\partial \phi_{2}}{\partial x_{2,j}} \frac{\partial \phi_{1}}{\partial x_{1,i}} w_{j} w_{jk}. \tag{16}
\]

The superscript \( s \) in the generalized equation (16) indicates that the first-order partial derivatives of
a **scaled** function \( y_{3,k} \), approximated in a variable space with **scaled** variables \( x_{1,i} \), is described by eq. (16).

It is very important to scale the values of the variables and functions (input–output data pairs) appropriately. There are three principal reasons for this. Firstly, the input node activities of the NN should be small enough that the hidden nodes are not saturated, i.e. their absolute values should not be too large. Otherwise the network will end up in an unacceptable local minimum. Secondly, it is imperative that the input values to the net are of the same absolute order. During training, each weight is adjusted according to the overall smallest error-weight gradient. The error-weight gradient is proportional to the magnitude of the input node according to backpropagation. A very small input node will, thus, prevent the weight connecting it to the hidden layer to make a significant contribution to the objective function and will slow down the training process. Thirdly, whilst learning is complicated if highly curved function surfaces are required, NNs can easily learn smooth relations. Thus, scaling which smooths out highly convoluted dimensions is desirable.

The NN-predicted function values should be scaled back to their true values. The same principle is valid for the NN gradients. The first-order partial derivative of the true function \( y_{3,k} \) in relation to the true values of a variable \( x_{1,i} \), can be computed after transforming eq. (16), as will be explained.

Consider \( y_{3,k} \) and \( x_{1,i} \) to be scaled according to the scaling functions \( \alpha \) and \( \beta \), i.e.

\[
y_{3,k} = \alpha(y_{3,k})
\]

\[
x_{1,i} = \beta(x_{1,i}).
\]

The inverse scaling functions \( \sigma \) and \( \delta \) are

\[
y_{3,k} = \sigma(y_{3,k})
\]

\[
x_{1,i} = \delta(x_{1,i}).
\]

The first-order derivatives of eqs (18a) and (18b) can now be used to transform eq. (16) to the form

\[
\frac{\partial y_{3,k}}{\partial x_{1,i}} = \left( \frac{\partial y_{3,k}}{\partial y_{1,i}} \right) \left( \frac{\partial y_{1,i}}{\partial x_{1,i}} \right) \left( \frac{\partial y_{1,i}}{\partial x_{1,i}} \right)^{-1} \frac{\partial y_{1,i}}{\partial x_{1,i}}.
\]

An analytical equation has been derived, which allows us to compute the partial gradients of a function within an unscaled variable space, if the function is approximated by an NN trained on scaled data.

It was mentioned earlier that the NN used in this paper contains sigmoidal hidden nodes, as well as linear input and output nodes. The derivative of the linear activation function of eq. (7) is unity. The sigmoidal activation function of eq. (8) has a derivative which can be reduced to the form

\[
\frac{\partial f_x}{\partial x_{z,n}} = y_{z,n}(1 - y_{z,n}).
\]

For the relevant NN, eq. (19) can be simplified to the following expression:

\[
\frac{\partial y_{3,k}}{\partial x_{1,i}} = \left( \frac{\partial y_{1,i}}{\partial x_{1,i}} \right)^{-1} \left( \frac{\partial y_{3,k}}{\partial y_{1,i}} \right) \sum_{j=1}^{m} y_{2,j} \times (1 - y_{2,j}) w_{ij} w_{jk}.
\]

With eq. (21) available, it is possible to compute the estimated partial gradients at different coordinates of a multivariate function with an NN, which was trained on scaled data representing the relationship between the function and its variables.

**Procedure for perturbation analysis.** It has been explained that perturbation analyses are performed at various coordinates within the variable space. Prior to each analysis, the NN state (activities of the NN nodes at the specific coordinate) is calculated. During the next step, eq. (21) is used to determine all the partial derivatives of the function (corresponding to an NN output node) at the specific coordinate in variable space. Each partial derivative corresponds to a specific input node.

In many cases, the partial derivatives at a coordinate will differ by orders of magnitude owing to an equally large difference between the values of the variables. It is important to adjust the partial derivatives at each coordinate so that the magnitudes of the modified gradients can be compared mutually. This is done by multiplying eq. (21) with the appropriate variable value

\[
ppv_{p.i} = x_{1,i} \frac{\partial y_{3}}{\partial x_{1,i}}
\]

where \( ppv_{p.i} \) is the **preliminary perturbation value** for the \( i \)th dimension or variable at location \( p \). (From now on it is assumed that only one function needs to be estimated and, thus, the subscript \( k \) on the output activities will be eliminated.)

A suitable criterion has been developed which represents the quantified influences of the different variables on the function so that they can be compared mutually.

The next criterion of analysis is the **relative perturbation value** \( (\text{r pv}) \), which is a scaled version of the \( ppv \) and emphasizes the relative importance of each of the variables. The \( \text{rpv} \) of the \( i \)th dimension at the \( p \)th location can be determined using eq. (23) as

\[
\text{rpv}_{p,i} = \frac{ppv_{p,i}}{ppv_{\text{max},p}} \times 100\%.
\]

Here \( ppv_{\text{max},p} \) is the largest \( ppv \) at the \( p \)th location. After determining the perturbation criteria values at all locations to be investigated, the influences of the different variables on the function are now quantified throughout the variable space so that the magnitudes of these average perturbed values can be compared directly.

**Dividing the variable space into subregions.** The perturbation results can now be used to identify boundaries between subspaces on the basis of eliminating...
different non-significant variables from neighbouring subspaces. A "cut-off" value for each rpv can be specified in order to divide the global variable space into subregions. A subregional boundary is identified within variable space at the points where the rpv of a specific dimension becomes smaller than the rpv cut-off. The dimensionality of a specific subregion is reduced by eliminating the dimension of which the rpv is smaller than the rpv cut-off. In practice, boundaries which are too close to the edges of the training set are also disallowed, because the global network tends to be inaccurate in these regions.

The resulting subregions are similar to the subregions of the MARS algorithm (Friedman, 1991a, b), but, whereas MARS creates hyperrectangular subregions, this method can create more general geometries. Further differences between this approach and MARS centre on the use of sigmoidal relations rather than splines, as well as linear combinations of the input variables instead of products.

DYNAMIC SIMULATION OF A TYPICAL PROCESSING OPERATION

The NN modelling methodology proposed in this paper is illustrated using a typical metallurgical processing operation as example. The CIL process is a gold-recovery process which still holds numerous secrets concerning its reaction mechanism. Various phenomenological models have been developed for this processing operation. Although some of these models can be used successfully to predict the dynamics of the process within certain operating domains, their domain specificity is a major drawback for full-scale industrial implementation.

In order to illustrate the mentioned methodology, artificial data points were generated by using process-variable profiles together with the material balance equations (25)-(27). This procedure of data generation will be described in greater detail below.

In the next few paragraphs the CIL processing plant layout, the material balance equations representing the process dynamics, as well as the phenomenological model describing the process kinetics, are described.

The CIL gold-recovery process

CIL plant layout. In Fig. 2 the CIL cascade is illustrated schematically. It consists of N adsorption reactors in series. Countercurrent flow of gold ore slurry, containing gold cyanide in solution, takes place through the cascade. There is a constant slurry stream down the cascade. Carbon is transferred periodically upstream, during which an upstream slurry flow also takes place. \( Q_f \) is the volumetric flow rate of the slurry stream entering the cascade from the leaching section into the first reactor. The volumetric flow rate of the slurry, which is transferred upstream during carbon transfer periods, is expressed by \( Q_t \). The volumetric flow rate of the total downflow slurry stream \( (Q_{fs}) \) from the ith reactor can be calculated as follows:

\[
Q_{fs} = Q_f + Q_s, \quad i = 1, \ldots, N. \quad (24)
\]

During the intervals when no carbon transfer takes place, \( Q_s \) is non-existent and \( Q_{fs} \) equals \( Q_f \) for all reactors.

Material balance equations. The dynamic behaviour of the CIL cascade can be simulated through a number of differential equations representing material balances for gold within the different phases. Equations (25)-(27) describe the gold mass balances over each reactor in the liquid, carbon and ore phases. The gold concentration in the liquid phase is represented by \( C \), while \( q \) and \( G \) are the average gold loading on

![Fig. 2. Flow sheet of the CIL process.](https://scholar.sun.ac.za)
the carbon and the average gold grade in the ore, respectively. The mass of ore and carbon within a specific reactor are \( M \) and \( W \), respectively. The void fractions and densities of liquid \( (l) \), ore \( (o) \) and carbon \( (c) \) are expressed by \( \epsilon \) and \( \rho \), respectively.

**Gold balance in liquid phase for stage 1:**

\[
\frac{dC_1}{dt} = \frac{\epsilon_{12}}{(\epsilon_{11} + \epsilon_{12})} Q_{s2} C_2 + \frac{\epsilon_{10}}{(\epsilon_{10} + \epsilon_{11})} Q_{fso} C_0 - \frac{\epsilon_{11}}{(\epsilon_{11} + \epsilon_{12})} Q_{s1} C_1 + f_1. \tag{25a}
\]

Gold balance in liquid phase for stages \( i = 2, \ldots, N \):

\[
\frac{dC_i}{dt} = \frac{\epsilon_{i+1}}{(\epsilon_{i+1} + \epsilon_{i+2})} Q_{s1} C_{i-1} + \frac{\epsilon_{i-1}}{(\epsilon_{i-1} + \epsilon_{i-2})} Q_{f3} C_{i-1} - \frac{\epsilon_{i-1}}{(\epsilon_{i-1} + \epsilon_{i+1})} Q_{s1} C_{i+1} + f_i. \tag{25b}
\]

Gold balance on carbon for stages \( i = 1, \ldots, N \):

\[
W_i \frac{dq_i}{dt} = \frac{\epsilon_{i+1}}{(\epsilon_{i+1} + \epsilon_{i+2})} Q_{s1} \rho G_i + f_1 - \frac{\epsilon_{i+1}}{(\epsilon_{i+1} + \epsilon_{i+2})} Q_{s1} \rho G_i + f_2. \tag{26}
\]

Gold balance on ore for stage 1:

\[
M_1 \frac{dG_1}{dt} = \frac{\epsilon_{12}}{(\epsilon_{11} + \epsilon_{12})} Q_{s2} \rho G_2 + \frac{\epsilon_{10}}{(\epsilon_{10} + \epsilon_{11})} Q_{fso} \rho G_0 - \frac{\epsilon_{11}}{(\epsilon_{11} + \epsilon_{12})} Q_{s1} \rho G_1 + f_3. \tag{27a}
\]

Gold balance on ore for stages \( i = 2, \ldots, N \):

\[
M_i \frac{dG_i}{dt} = \frac{\epsilon_{i+1}}{(\epsilon_{i+1} + \epsilon_{i+2})} Q_{s1} \rho G_i + \frac{\epsilon_{i-1}}{(\epsilon_{i-1} + \epsilon_{i-2})} Q_{f3} \rho G_i - \frac{\epsilon_{i-1}}{(\epsilon_{i-1} + \epsilon_{i+1})} Q_{s1} \rho G_i + f_3. \tag{27b}
\]

If \( f_2 \) and \( f_3 \) are known, these differential equations can be solved by numerical techniques such as the fourth-order Runge–Kutta integrator. These three reaction terms can be expressed in numerous ways through different modelling techniques. If a data base which captures the knowledge of the process kinetics (KBS) is available for \( f_1, f_2 \) and \( f_3 \), their values can be determined at each instant in time during a simulation run. The same holds for phenomenological, regression analysis or NN models for the reaction functions.

**Kinetic and equilibrium models.** The model employed by a phenomenological model simulator (dynamic simulator incorporating a phenomenological model) was used to generate artificial data points for the CIL process. \( f_1, f_2 \) and \( f_3 \) in eqs (25)–(27) are the kinetic reaction terms and can be described by the following equations according to the film diffusion model for adsorption onto the carbon surface and an empirical expression for the process of leaching occurring within the CIL cascade:

\[
f_2 = \frac{6k_f W_i}{\rho_i dt} (C_i - C_{\infty}) \tag{28}
\]

\[
f_3 = -k_i M_i (G_i - G_{i-1})^2 \tag{29}
\]

\[
f_1 = -f_2 + f_3. \tag{30}
\]

The equilibrium at the carbon surface is described by the Freundlich isotherm as

\[
C_i = \left( \frac{q}{A} \right)^{1/n} \tag{31}
\]

where \( A \) and \( n \) are the equilibrium parameters, specified to be 10 and 0.15, respectively, for the specific carbon. Further properties of the carbon are a density \( \rho_c \) of 900 kg m\(^{-3} \) and an average particle diameter \( d_p \) of 1.0 mm. The adsorptive and leaching coefficients \( (k_f \) and \( k_i \) are assumed to remain constant at \( 10^{-13} \) m s\(^{-1} \) and 1.2 k\( \text{g}_{\text{ore}} \) m\(^{-3} \) s\(^{-1} \), respectively. The grade of gold in the ore after infinite leaching time \( (G_{\infty}) \) was specified to be \( 8 \times 10^{-3} \) g\( \text{Au} \) k\( \text{g}_{\text{ore}} \).

**Constructing an artificial data set**

During simulation runs with the phenomenological model simulator for the CIL process, variable profiles were constructed which covered the process domain in which the simulation runs were completed as portrayed in the final part of this paper. Using these profiles with the balance equations (25)–(27), \( f_1, f_2 \) and \( f_3 \) were calculated and the values of \( C, q, G, M \) and \( W \) registered together with the three reaction terms. This was done in order to illustrate how continuous data can be gathered for modelling purposes, as explained earlier. The values for \( M \) and \( W \) were varied between 400–500 tons and 12–22 tons, respectively. All simulations were done at fixed reactor volumes of 750 m\(^3 \) for all stages. The values of \( f_1, f_2 \) and \( f_3 \) of these data points (variable–function pairs with 5 input variables and 3 output functions) were then randomly corrupted with 15% Gaussian noise. The first training set, thus, contained 800 noisy data points which is typical of an industrial system. After scaling the values of the data points, this training set was employed to train an NN which in turn was used to perform perturbation analyses on the training data set.

**Scaling the training data**

The input–output data pairs for the CIL example were scaled as illustrated in Fig. 3. The variables \( C, q \) and \( G \) varied over a few orders of magnitude and were therefore scaled logarithmically. \( M \) and \( W \) were linearly scaled. All scaled values were transformed into the range \(-3.5–3.5\). Figure 4 illustrates the distribution of the scaled variables. It shows that the num-
number of scaled values for all variables $C$, $q$, $G$, $M$ and $W$ are more or less symmetrically distributed around zero. This could be expected for $M$ and $W$ which are scaled linearly, and also shows a favourable situation for training with the logarithmically scaled values for $C$, $q$ and $G$.

Training an NN

The search for an adequate NN for a specific application is very important. A number of factors should be considered prior to training an NN. One of these factors is the number of hidden nodes to be specified. Although techniques have been proposed by Huang and Huang (1991) as well as Baum and Haussler (1989) to determine an optimal NN configuration for NNs used as binary classifiers, no effective procedure exists to estimate this optimal architecture for NNs employed as function approximators. An excessive number of hidden nodes will cause bad generalization properties, so that the NN would not be able to interpolate effectively between adjacent training data points. As opposed to this, too few hidden nodes will limit the competence of the NN to locate an adequate mapping between functions and variables. An iterative approach is, therefore, employed to determine the appropriate number of hidden nodes: various networks with different numbers of hidden nodes are trained, and the network with optimal test set performance is chosen.

A number of NNs were trained with the five-dimensional training data set containing 800 data points. The different training runs were performed with different initial weight matrices and differing numbers of hidden nodes. The generalization properties of each
net were evaluated with a test data set composed of 400 data points not used during training. The NN which performed the best on the test data set was found to have 10 hidden nodes in this case. It has been proved that NN training algorithms with gradient descent optimization procedures converge more easily to a satisfactory weight matrix if noisy data are used. This was also the case for the NN trained here. This trained NN was used to perform perturbation analyses on the training data.

**Perturbation analysis and elimination of less significant variables**

Perturbation analyses were performed at each of the 800 training data points. In this process, the orders of magnitude of the variables $C$ and $q$ are highly correlated, and the other three variables are of secondary importance. Consequently, the perturbation analysis is performed as a function of $C$ only. Thus, the rpvs of the reaction functions $f_1$, $f_2$ and $f_3$ as a function of this variable (ignoring all others) are shown in Figs 5–7. It is impossible to plot the rpvs of all variables in a five-dimensional variable space and it will be unsuitable to show a complete set of graphs for the rpvs within single dimensions.

Figure 5 portrays the perturbation results of $f_1$ and shows that all five variables play significant roles as variables of $f_1$. Notable is the strong influence of $q$ on $f_1$ within the high-$C$ region and the drop in significance of $q$ towards the region of lower $C$-values within the variable space. This corresponds to the Freundlich isotherm of eq. (31). Although it should be possible to determine an rpv-cut-off in an automated way, a suitable cut-off (rpv < 10%) was selected by hand in order to illustrate this technique. According to this, the variable space is divided into two subspaces at approximately $C = 0.2$ g m$^{-3}$. It can also be noted that $G$ plays an increasingly important role at lower $C$-values, confirming the role played by the leaching term in the phenomenological model for $f_1$.

It can be observed from Fig. 6 that $C$ and $W$ play relatively strong roles in determining $f_2$ throughout the variable space. Also notable is the strong influence of $q$ on $f_2$ within the high-$C$ region and the drop in significance of $q$ towards the region of lower $C$-values within the variable space. The rpvs for $G$ and $M$ remain relatively small through the CIL variable space, so that these two variables can be completely eliminated as variables for $f_2$. These observations correspond to the phenomenological expression for $f_2$ [eq. (28)].

The perturbation results in Fig. 7 confirm the dependence of $f_3$ on $G$ and $M$. It shows that the dimensions for $C$, $q$ and $W$ can be eliminated (rpvs < 10%), thus reducing the dimensionality of the variable space for $f_3$ drastically. The higher rpvs for $G$, if compared with those of $M$, are attributed to the quadratic relation between $f_3$ and $G$. The variable space for $f_3$ is not subdivided.

The variable spaces of $f_1$ and $f_2$ have now been subdivided into two subspaces each. The first subspace of $f_1$ (for $C > 0.2$ g m$^{-3}$) is five-dimensional, while the dimensionality of the second subspace ($C < 0.2$ g m$^{-3}$) has been reduced to four. In the case of $f_2$, its first subspace contains three dimensions (for $C$, $q$ and $W$). This subspace is adjacent to the second subspace with dimensions for $C$ and $W$ only. The undivided variable space for $f_3$ has two dimensions ($G$ and $M$).

**Identifying mathematical relations**

During the following step, different simplified NNs are trained for the different subspaces of each function. These NNs (2 for $f_1$, 2 for $f_2$ and a single one for $f_3$) are employed to identify mathematically simple relations between the functions and their variables. The procedure is explained with reference to the NN for $f_3$. At different domains within the variable space, the value of $M$ is varied, while $G$ is kept constant. $f_3$ is calculated at these different coordinates. Figure 8 exhibits the predicted curves at three distinct values of $G$. It confirms the direct proportionality between $f_3$ and $M$. The curves can be forced through the origin, so that $f_3$ can be expressed as follows:

$$f_3 = Mr_3$$  \hspace{1cm} (32)

where $r_3 = z(G)$. 

**Fig. 5.** Rpvs of the five variables of the CIL example—perturbation results for $f_1$ through the global variable space.

**Fig. 6.** Rpvs perturbation results for reaction function $f_2$. 

Stellenbosch University https://scholar.sun.ac.za
Dynamic modelling of ill-defined processing operations

The same procedure was performed for \( f_2 \) and \( W \), and eq. (33) displays the simplified functions:

\[
\begin{align*}
f_3 &= r_2 + \beta(C, q) \\
r_2 &= \beta(C) \quad \text{if } C < 0.2 \text{ g m}^{-3}
\end{align*}
\]

(33)

where

\[
r_2 = \beta(C, q) \quad \text{if } C > 0.2 \text{ g m}^{-3}
\]

and

Although the relevant NNs for \( f_1 \) predicted linear relations between \( f_1 \) and \( W \), as well as \( f_1 \) and \( M \), the mathematical relations could not be isolated from the "ill-defined" dimensional space. This is due to the presence of two terms in eq. (30).

If the assumption can be made that some \textit{a priori} knowledge about the CIL process is available, such knowledge can be applied to simplify the modelling of \( f_1 \). It should be emphasized that this NN approach to modelling is not intended to replace existing modelling techniques, but should contribute to the modelling tool kit. If it is known, for example, that the gold mass balance in the liquid phase is dependent on an adsorption and leaching process, eqs (32) and (33) can be combined to describe \( f_1 \) as

\[
f_1 = Mr_3 - Wr_2.
\]

(34)

The result: an NN-phenomenological model

According to the relations of the model equations (32)–(34), an NN-phenomenological model with five NNs (two nets for each C-interval for \( r_2 \) and a net for \( r_3 \)) was compiled. These NNs were trained on exactly the same training data as the NNs trained earlier. The NNs of \( r_3 \) and the lower C-range subspace network model of \( r_2 \) had only one input node in addition to the bias node, while the NN for \( r_2 \) within the higher C-range contained two input nodes only. The dimen-
sionality of the "ill-defined" parts of the system has, thus, been reduced considerably from a five-dimensional one to an ill-defined dimensionality of only 2 (function $r_2$ for $C > 0.2 \text{ g m}^{-3}$) and 1 (functions $r_2$ for $C < 0.2 \text{ g m}^{-3}$ and $r_3$) by means of the techniques described above. This new model was used to replace the phenomenological model in a dynamic CIL-cascade simulator. At each time step, $f_1$, $f_2$ and $f_3$ could now be calculated by eqs (32)-(34) during a simulation run.

A second NN model was constructed for $f_1$, $f_2$ and $f_3$ using the five-dimensional variable space. This model consisted of three NNs (one net for each of $f_1$, $f_2$ and $f_3$) which were trained with the same data points used to develop the NN phenomenological model. An NN with only one output node will more easily learn the mapping between this single function and its variables as opposed to an NN with more output nodes. This was the reason for training separate NNs for the different reaction functions. Each one of these NNs had five input nodes for the five variables involved. This NN model was developed to show the inadequacy of implementing an NN model of high dimensionality in a straightforward fashion, as compared to the effectiveness of the model proposed in this paper.

COMPARING SIMULATION RESULTS

A dynamic simulator was developed which solved the differential mass balance equations (25)-(27) by incorporating the three different models for $f_1$, $f_2$ and $f_3$. The predictions of the phenomenological model simulator can be used as a basis of comparison. Figure 9 illustrates some results of simulation runs completed with the three different models under the same starting conditions. Fifty per cent of the carbon contents of each reactor is transferred every 12 h over a transfer period of half an hour. It should be apparent from this figure that the NN phenomenological model is much more accurate than the NN model of high dimensionality. Although the concentration profiles of Fig. 9 represent only a small part of the simulation results, the same conclusions can be drawn from the simulated variable profiles of each CIL reactor.

The superiority of the simulation results with the NN-phenomenological simulator (reduced dimensionality) to those of the NN-model simulator is mainly due to the large reduction in ill-defined dimensionality and an associated improvement of the model.

CONCLUSIONS

It has been explained how process data can be collected directly from an industrial operating unit, and a new technique to conduct process identification on the basis of such continuous data was proposed. A novel modelling approach using NNs has been developed and a typical metallurgical process, the CIL process for gold recovery, was used to demonstrate how modelling can be conducted along this line.

In chemical processing, process identification usually deals with problems caused by high dimensionality. This "curse of dimensionality" becomes even worse in the case of NN modelling. A simple perturbation analysis method was proposed, which uses an NN to estimate the first-order partial derivatives of a multivariate function. These partial gradients in turn were employed to quantify the relative influences of different variables on a function. This forms part of the broader methodology for modelling on the basis of connectionist networks. The perturbation results can be used to subdivide a variable space into subspaces, after which less significant variables could be eliminated, thus reducing the dimensionality of a subspace accordingly. NNs were then employed to identify mathematically definable relations between
a function and certain variables. During the final step, NNS with low dimensionalities could be trained to capture the ill-defined relations between a function and the remaining variables for each subspace. This model (a combination of phenomenological expressions and NNS) was incorporated in a dynamic simulator for the CIL process. The performance of this process was successfully predicted by the simulator.

REFERENCES


Quinlan, J. R., 1983, Learning efficient classification procedures and their application to chess and games, in Machine Learning: an Artificial Intelligence Approach (Edited by

NOTATION

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>Freundlich isotherm parameter</td>
</tr>
<tr>
<td>$C$</td>
<td>gold concentration, g m$^{-3}$</td>
</tr>
<tr>
<td>$d$</td>
<td>average diameter of solid particles, m</td>
</tr>
<tr>
<td>$f$</td>
<td>reaction term in mass balances</td>
</tr>
<tr>
<td>$G$</td>
<td>grade of gold in the ore, g kg$^{-1}$</td>
</tr>
<tr>
<td>$k_f$</td>
<td>leaching parameter, kg g$^{-1}$ s$^{-1}$</td>
</tr>
<tr>
<td>$M$</td>
<td>mass of gold ore in a single column, kg</td>
</tr>
<tr>
<td>$N$</td>
<td>number of adsorption stages or reactors in cascade</td>
</tr>
<tr>
<td>$n$</td>
<td>Freundlich isotherm parameter</td>
</tr>
<tr>
<td>$ppv$</td>
<td>preliminary perturbation value</td>
</tr>
<tr>
<td>$q$</td>
<td>gold loading on the resin</td>
</tr>
<tr>
<td>$Q$</td>
<td>volumetric flow rate of slurry, m$^{3}$ s$^{-1}$</td>
</tr>
<tr>
<td>$rpv$</td>
<td>relative perturbation value</td>
</tr>
<tr>
<td>$t$</td>
<td>time, s</td>
</tr>
<tr>
<td>$V$</td>
<td>volume of RIP column, m$^{3}$</td>
</tr>
<tr>
<td>$w$</td>
<td>NN connection weight</td>
</tr>
<tr>
<td>$W$</td>
<td>resin mass in a single column, kg</td>
</tr>
<tr>
<td>$x$</td>
<td>input to a node of an NN</td>
</tr>
<tr>
<td>$y$</td>
<td>activity of a node in NN</td>
</tr>
</tbody>
</table>

Greek letters

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\epsilon$</td>
<td>void fractions</td>
</tr>
<tr>
<td>$f$</td>
<td>function symbol</td>
</tr>
<tr>
<td>$\rho$</td>
<td>density, kg m$^{-3}$</td>
</tr>
<tr>
<td>$\phi$</td>
<td>activation function of a node in NN</td>
</tr>
<tr>
<td>$\tau$</td>
<td>residence time of process stream, s</td>
</tr>
</tbody>
</table>

Subscripts

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c$</td>
<td>carbon</td>
</tr>
<tr>
<td>$f$</td>
<td>$i$, $s$, $fs$ different countercurrent slurry streams in RIP cascade</td>
</tr>
<tr>
<td>$i$</td>
<td>$i$th column, reactor or stage</td>
</tr>
<tr>
<td>$i$</td>
<td>$i$th node in the NN input layer</td>
</tr>
<tr>
<td>$in$</td>
<td>in processing unit inlet stream</td>
</tr>
<tr>
<td>$j$</td>
<td>$j$th node in the NN hidden layer</td>
</tr>
<tr>
<td>$k$</td>
<td>$k$th node in the NN output layer</td>
</tr>
<tr>
<td>$l$</td>
<td>liquid phase</td>
</tr>
<tr>
<td>$n$</td>
<td>$n$th node in NN layer</td>
</tr>
<tr>
<td>$o$</td>
<td>gold ore</td>
</tr>
<tr>
<td>$out$</td>
<td>processing unit outlet stream</td>
</tr>
<tr>
<td>$p$</td>
<td>coordinate label in variable space</td>
</tr>
<tr>
<td>$s$</td>
<td>surface of solids</td>
</tr>
<tr>
<td>$z$</td>
<td>$z$th NN layer</td>
</tr>
</tbody>
</table>

Superscripts

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s$</td>
<td>scaled function or variable values</td>
</tr>
<tr>
<td>$t$</td>
<td>true function or variable values</td>
</tr>
</tbody>
</table>
NEURAL NETS FOR THE SIMULATION OF MINERAL PROCESSING OPERATIONS: PART I. THEORETICAL PRINCIPLES

T.J. VAN DER WALT§, J.S.J. VAN DEVENTER§ and E. BARNARD†

§ Department of Metallurgical Engineering, University of Stellenbosch, Stellenbosch 7600, South Africa
† Department of Computer & Electronics Engineering, University of Pretoria, Pretoria 0001, South Africa

(Received 29 March 1993; accepted 3 May 1993)

ABSTRACT

The ill-defined nature of processes in the metallurgical industry necessitates the quest for new modelling techniques to emulate features of processes which are poorly understood from a fundamental point of view. For this reason nonparametric regression techniques such as neural nets offer an appealing alternative to fundamental modelling. The robust associative and computational properties of neural networks make these regression tools ideally suited for the modelling of ill-defined systems.

Being the most commonly-used connectionist network, sigmoidal backpropagation neural networks (SBNN's) have been shown to model metallurgical and chemical systems satisfactorily without any a priori knowledge about the system provided sufficient data are available. This paper introduces the field of connectionist networks to the metallurgical process engineer and describes the fundamentals of an SBNN.

Keywords
Sigmoidal backpropagation neural nets, connectionist networks, problem of dimensionality.

INTRODUCTION

With the rapid development of new technology in the field of process engineering, the modern engineering environment expands continually and becomes more complex. To assimilate engineering theory and practice in order to apply new knowledge to a myriad of situations in the process environment is a daunting task. It becomes more and more important to take swift, correct and consistent action based upon quick and effective decision making during plant operation. Due to the enhanced level of sophistication in the processing industry, there is a growing need to estimate process behaviour more accurately. The construction of accurate approximations for predicting the behaviour of processing operations is a necessity to maintain optimal productivity through improved process control. Adequate models must be used in the optimal design of more sophisticated processing equipment. Furthermore, the location of deviations from acceptable processing conditions well in advance is important in order to implement corrective measures which can prevent major loss in productivity.

Although extended fundamental research has been done on metallurgical processes, most unit operations in the processing industry are still ill-defined in some way. This is normally attributed to an inadequate understanding of the intrinsic physics of the process. For such processes it is difficult to construct an adequate fundamental model. Whereas different approaches to modelling have been proposed and applied in industry, the ill-defined nature of processes necessitates the quest for new modelling techniques to overcome this problem. For this reason there is a growing interest in nonparametric regression techniques as an alternative to fundamental modelling.
Connectionist networks or neural nets have caught the imagination of many researchers from psychologists to engineers due to some exceptional features of these techniques. All connectionist networks are inherently related to other nonlinear and nonparametric regression techniques and also form part of the artificial intelligence (AI) collection.

Although the principal fundamentals of neural nets had already been formulated as early as a century ago, the renaissance of connectionist networks started only a decade ago [1]. Many aspects of this field of research remain largely unexplored despite the tremendous interest that the field has recently attracted. This applies especially to the use of neural nets for the purpose of modelling. Over recent years neural nets have shown exceptional performance as regression tools, especially if used for the purpose of pattern recognition and classification. However, if neural nets are employed as function approximators in the processing industry, vast numbers of model parameters (connection weights) are normally required to memorise satisfactorily information as contained by process data of low population density. As for other nonparametric regression techniques, this can result in weak generalisation properties of the relevant neural net model. Bellman [2] referred to this obstacle in the way of effective modelling as the "curse of dimensionality".

Various connectionist network topologies have been proposed to date. Lippmann [3] gave an introduction to some of these structures, such as Hopfield nets, Hamming nets, Carpenter/Grossberg classifiers, perceptrons, multilayer perceptrons and Kohonen self-organising feature maps. Recent success in connectionist network research is mainly attributed to the achievements of multilayer perceptrons with backpropagation training algorithms.

The "three-layer backpropagation neural network" with sigmoidal hidden nodes and feedforward configuration is the most widely studied and applied connectionist network. This connectionist network topology is commonly referred to as artificial neural nets (ANN's) in the literature. Since the latter terminology can also be used as a collective term that also includes other connectionist networks, the relevant topology will be called sigmoidal backpropagation neural nets (SBNN’s) further in the text.

**NEURAL NETWORKS**

Neural nets are mathematical regression tools which appear to be very promising in respect of a wide range of modelling problems. Although this area of study is still in its infancy, encouraging results with various applications have been published to date. Connectionist networks are used for optical recognition purposes (military, medical and criminological applications), in speech recognition, system identification and fault diagnosis, process control, etc.

Research Developments in the use of Neural Nets in Chemical and Metallurgical Applications

A substantial amount of work has already been reported on and is in progress to employ connectionist networks for process control, fault diagnosis and optical classification in the chemical and metallurgical industry. A number of authors reported on applications and novel neural net control strategies in the processing industry [4-8]. The detection and diagnosis of process faults via neural nets have also been studied widely [9-15].

Neural nets generally perform excellently in pattern recognition and classification problems. Similarly, the application of these networks for the purpose of modelling holds great promise, but still has to be exploited to a large extent. The reason for this is the problems experienced if a function is approximated with this nonparametric regression technique in a variable space containing sparse data (i.e. the "curse of dimensionality" [2]). Hence, most research in this regard has been conducted on relatively simple applications [16-22].
Little work has been done with a view to employing connectionist networks for modelling purposes in the metallurgical industry. Reuter et al. [23] used a three-layered neural net to model activities in metals and slags, the distribution of species between metal and slag, and slag viscosity on the basis of published data. During another study, Reuter et al. [24] proposed the use of neural nets in the construction of generalised connectionist network kinetic rate equations. It was shown how such equations can be employed for the simulation and identification of batch and mixed flow mineral processing and metallurgical reactors.

Architecture of Sigmoidal Backpropagation Neural Nets (SBNN's)

The basic architecture of an SBNN is exhibited in Figure 1. The nomenclature used to explain the mathematics of the SBNN is also illustrated in this figure. The SBNN consists of three layers of nodes, viz. (i) an input layer containing \( m_1 \) nodes (including a bias node), (ii) a single hidden layer \( (m_2 \) nodes) and (iii) an output layer with \( m_3 \) nodes. Nodes of adjacent layers are interconnected and these connections quantified with weights. A weight matrix describes the relative strength of the interconnections, and therefore the relative importance of variables. Each node is characterised by an activation function \( \phi \).

During the first step of a feedforward calculation of the net, the values of the predictor variables as contained in a training data point are fed to the input nodes. Each predictor variable is associated with a different input node. Each node \((n)\) in a layer \((z)\) has two important features, namely the node input \((x_{z,n})\) and its activity \((y_{z,n})\). The activity \((y_{z,n})\) of a specific node is a function of its input \((x_{z,n})\).

Equation (1) gives a notation of such an activation function.

\[
y_{z,n} = \phi_n(x_{z,n})
\]

An additional input node, the bias node with a constant value of \( y_{1,m1} = 1 \), introduces additional degrees of freedom which enable a curve (hyperplane) to be fitted in such a way that the intersections of the hyperplane with the weight axes would not necessarily be at the origin. The input to each hidden node \((z = 2)\) is calculated with Equation (2) and is the additive product of the weights \((v_{ij})\) of connections.
between each input node \( i \) and a specific hidden node \( j \), and the activities of each corresponding input node \( (y_{i1}) \).

\[
x_{ij} = \sum_{i=1}^{m_1} v_{ij} y_{i1} ; \quad y_{i1,m_1} = 1
\]  

(2)

The input to node \( k \) in the output layer \( (z = 3) \) has a similar form, as illustrated in the following relationship:

\[
x_{jk} = \sum_{j=1}^{m_2} w_{jk} y_{j2}
\]  

(3)

Each node in the output layer is associated with a response variable to be approximated. The output of the SBNN is simply the activities of the nodes in this layer.

The activation functions of nodes are normally squashing functions, which can either be linear (Equation 4a), sigmoidal (Equation 4b) or hard-limiter functions:

\[
\phi_z(x_{z,n}) = x_{z,n}
\]  

(4a)

\[
\phi_z(x_{z,n}) = \frac{1}{1 + e^{-x_{z,n}}}
\]  

(4b)

It should be noted that any functional relation may be represented in connectionist structure. To illustrate this principle, consider the following simple linear functional relation:

\[
f = V \circ X
\]  

(5)

In this relation \( V = [v_{1,1}, v_{2,1}, \ldots, v_{m_1,1}] \) is the parameter matrix and \( X = [x_1, x_2, \ldots, x_{m_1-1}, 1]^T \) is the matrix of independent variables and the bias of unity. This relation can be represented in a connectionist configuration by a two-layer network containing \( m_1-1 \) linear input nodes for each of the variables \( x_1 \) to \( x_{m_1-1} \) and a bias node of unity in the input layer, as well as a single linear output node. The connection weights between the nodes of the input layer and the output node, \( i.e. \ v_{1,1} \) to \( v_{m_1,1} \), represent the parameters of the linear model.

Although the mentioned activation functions are the most commonly used, it is also possible to employ any other function such as exponential or harmonic functions as nodal relation. Van der Walt [25] developed a connectionist network topology called a regression network that can represent any functional relationship in a connectionist configuration by use of different combinations of activation functions and either additive or product inputs to the different nodes. In specifying the network structure to be either layered or nonlayered, it should be clear that the relational representability of such an approach is theoretically unlimited.

Model Selection

The selection of the network model \( i.e. \ the choice of the number of hidden nodes or network weights in the case of nonparametric networks such as SBNN's) is an important factor in connectionist network
applications. Although techniques have been proposed [26-27] to determine an optimal configuration for SBNN's used as binary classifiers, no effective procedure has yet been published to estimate this optimal architecture for SBNN's prior to training of the network. An excessive number of hidden nodes will cause bad generalisation properties, so that the neural network would not be able to interpolate effectively between adjacent training data points. As opposed to this, too few hidden nodes will limit the competence of the network to locate an adequate mapping between response and predictor variables. An iterative approach is generally employed to determine the appropriate number of hidden nodes. Various networks with different numbers of hidden nodes are trained, and the network which performs best on a test data set (i.e. a data set containing samples not considered during training) is selected as the most adequate model.

Training or Optimisation

The preferable method of knowledge acquisition by a neural network is to train the net with examples presented to the training algorithm by the environment. Training or learning plays an important role in neural network models. Learning in connectionist networks is mostly understood as supervised learning, i.e. training the connectionist network with predefined input-output pairs.

The training process is a procedure of minimising an objective function in the weight space. Although various error functions have already been used, the most commonly used objective function is the least mean square (LMS) error function. The best known training algorithm is the backpropagation algorithm [28].

When backpropagation was introduced originally, it was proposed that the error function should be optimised using gradient descent. Most connectionist networks use the momentum algorithm [29] which is characterised by two parameters (the learning rate and momentum constant), whose values are critical to the success of the optimisation process. The conjugate-gradient (CG) optimisation method gained popularity recently over techniques such as steepest descent with momentum due to its ability to locate a minimum of a multivariate function much faster. Also important to note is that the CG-technique eliminates the choice of critical parameters. On the other hand, like all gradient descent optimisation techniques, the CG-algorithm can experience trouble to locate the global minimum if the error surface of the LMS fitted to the data of a specific system contains local minima.

Scaling of the Training Data

It is important to adequately scale the training data prior to training. There are three principal reasons which can be emphasised. Firstly the input node activities of the neural net should be small enough that the sigmoidal hidden nodes are not saturated, i.e. their absolute values should not be too large. Otherwise the network will end up in an unacceptable local minimum. Secondly, it is imperative that the input values to the net are of the same absolute order: During training each weight is adjusted according to the overall error gradient (with respect to the weights), which is directly proportional to the magnitude of the input node. A very small input node will thus prevent the weight that connects it to the hidden layer from making a significant contribution to the derivative of the objective function and will slow down the training process. Thirdly, whilst learning is complicated if highly curved function surfaces are required, neural nets can easily learn smooth relations. Thus, scaling which smooths out highly convoluted dimensions is desirable.

Scaling is performed in two ways in Part II of this paper series, viz. linearly or logarithmically. Logarithmic scaling is beneficial if the values of a specific variable or function vary over different orders of magnitude in the same direction. The scaling expressions for logarithmic and linear scaling are given by Equations (6a) and (6b) respectively.
In these equations $x_{\text{sca}}$ and $x_{\text{act}}$ represent the scaled and actual values of a specific predictor or response variable, while $c_1$ to $c_6$ are the scaling parameters. Note that scaling parameter $c_3$ can be used to transform the values of a variable which contains both positive and negative values into the positive range during logarithmic scaling, so that $(c_3 x_{\text{act}} + c_5) > 0$. This can be done effectively only if the norms of all negative values $c_2 x_{\text{act}}$ are fairly small owing to the sensitivity properties of the logarithmic function. The logarithmically and linearly scaled values are mean-centred around zero with the scaling parameters $c_4$ and $c_6$ respectively.

The neural network training algorithm used for the purposes of Part II of this paper series was found to perform satisfactorily if the values of the input-output data pairs were scaled between approximately -3 and 3.

**CONCLUSIONS**

In this paper a broad overview of connectionist networks was presented and a reference base was outlined for the process engineer who is not familiar with this field of study. The basic architecture of the most widely used connectionist network, namely the sigmoidal backpropagation neural network (SBNN), as well as a procedure of model selection was described. This was followed by a discussion about optimisation research of connectionist networks, while a procedure was proposed to preprocess the training data.

In Part II of this paper series it is illustrated how neural net models are developed for metallurgical processes.

**NOMENCLATURE**

- $c_1$ to $c_6$: Scaling parameters of the scaling expressions defined by Equations (6a) and (6b)
- $m_i$: Number of nodes in network layer $i$
- $V$: Model parameter matrix
- $v$: Weights of connections between input and hidden nodes of SBNN
- $w$: Weights of connections between hidden and output nodes of SBNN
- $x$: Input to a neural network node
- $x$: A process variable (predictor or response)
- $\phi$: Activity of a neural network node
- $y$: Response variable or function
- $\hat{y}$: Model-predicted value of function or response variable

**Greek**

- $\phi$: Activation function of network node
- $X$: Matrix of independent variables
- $x_i$: Independent variable - $i^{th}$ element of matrix $X$
Subscripts and Superscripts

<table>
<thead>
<tr>
<th>Subscript</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>act</td>
<td>Actual or true value of a variable</td>
</tr>
<tr>
<td>i</td>
<td>$i^{th}$ node in the input layer of the SBNN</td>
</tr>
<tr>
<td>j</td>
<td>$j^{th}$ node in the single hidden layer of the SBNN</td>
</tr>
<tr>
<td>k</td>
<td>$k^{th}$ node in the output layer of the SBNN</td>
</tr>
<tr>
<td>a</td>
<td>Node label in the $z^{th}$ layer of an SBNN</td>
</tr>
<tr>
<td>sca</td>
<td>Scaled value of variable</td>
</tr>
<tr>
<td>z</td>
<td>Label of SBNN layer</td>
</tr>
</tbody>
</table>

REFERENCES


The use of neural nets to detect systematic errors in process systems

C. Aldrich and J.S.J. van Deventer
Department of Metallurgical Engineering, University of Stellenbosch, Stellenbosch 7600, South Africa
(Received 15 September 1992; accepted after revision 17 March 1993)

ABSTRACT

The monitoring of plants and the verification of process models depend crucially on reliable sets of steady state component and total flow rate data. These measurement data are generally subject to random noise (and possibly systematic errors) and typically violate the process constraints of the system. It is consequently necessary to adjust the data, and also to account for systematic or gross errors in the data prior to this reconciliation procedure, or as part of it, in order to avoid severe impairment of the adjustment process. This can be accomplished by using a back propagation neural net to form an internal representation of the relationship between the distributions of the measurement residuals and the residuals of the process constraints. The major advantage of using neural nets instead of conventional statistical methods, is that neural nets can also be used to detect systematic errors in process systems subject to non-linear process constraints (a common situation in the chemical and mineral industry that is not accommodated satisfactorily by traditional statistical procedures).

INTRODUCTION

The acquisition of reliable plant data is fundamental to a clear understanding of the operational behaviour of a plant, the modelling and optimization of process circuits, as well as the identification of other phenomena peculiar to the process. These data are generally subject to random noise, or even gross errors, owing to inadequate instrumentation, failure or miscalibration of measuring instruments, the departure of the process from steady state due to malfunctioning process equipment, or significant changes in the environment (Hlavacek, 1977). Any set of process data will consequently violate mass and energy conservation requirements, as well as other physical constraints pertaining to the process and will have to be adjusted in order to satisfy these constraints. Under these circumstances it is essential that gross errors are detected and accounted for prior to, or during the reconciliation process, since failure to do so could result in a severely distorted picture of the process (Hodouin and Coelho, 1987). Methods for the detection and isolation of gross errors can be grouped into two major categories, depending on whether they are based on the use of a plant model or not (Gertler, 1988). Model-based
methods exploit the concept of analytical redundancy, and virtually all these methods involve statistical tests associated with the characteristics of the constraint residuals generated by measurement errors (Mah, 1989). Unfortunately these tests are generally only useful as far as systems subject to linear constraints are concerned (Serth et al., 1987; Tjoa and Biegler, 1991). In the chemical and mineral industries this is a major drawback, since most process systems in these industries are non-linear. In this paper a new method is consequently proposed for the detection of systematic errors in constrained measurement data. This method is based on the powerful capability of neural nets to classify measurement errors, and is not hindered by the nature of the system constraints.

Problem statement

A popular technique for the generation of residual from plant measurements in the chemical engineering literature is based on so-called static balance equations (Gertler and Luo, 1989). These equations contain a static system matrix $C$ and variables $y$, which are usually restricted to mass and energy flows (Romagnoli and Stephanopoulos, 1981). For the purpose of this paper Crowe's (1989) description of the process constraints of a typical linear system serves to clarify the gross error detection problem. For more general representations of these types of problems the reader is referred to the literature (Gertler, 1988).

$$C \cdot y = 0 \quad (1)$$

where $C$ is an $(m \times n)$ constraint matrix of full row rank $m$ ($n > m$) and $y$ is the $(n \times 1)$ vector of true values of the state variables. If

$$y' = y + e \quad (2)$$

where $y'$ constitutes the $(n \times 1)$ vector of measurements of the true values $y$, with an $(n \times 1)$ error vector $e$, and covariance matrix $Q$, then the measured values of the process variables generally violate the process constraints

$$C \cdot y' = y'' \neq 0$$

or in terms of the true values and error components

$$C \cdot (y + e) = y'' \quad (3)$$

and assuming the constraints to be linear

$$C \cdot (y + e) = C \cdot y + C \cdot e = y''$$

i.e.

$$C \cdot e = z'' \quad (4)$$
If it is assumed that the error vector $e$ has a Gaussian distribution and that no systematic errors are present (the null hypothesis), $z''$ is a multivariate normal with a zero mean and known variance (Madron et al., 1977; Romagnoli and Stephanopoulos, 1981; Mah and Tamhane, 1982; Tamhane and Mah, 1985), i.e.

$$E(z'') = E(C - e) = C - E(e) = 0$$

Otherwise the expected value of $z''$ is not zero (the alternative hypothesis), i.e.

$$E(z'') = b \neq 0$$

which indicates the presence of a systematic error with a bias of magnitude $b$. By making use of standard statistical criteria, or variants of these statistics, the two hypotheses can subsequently be evaluated and rejected or accepted, and the presence of gross errors be determined.

**Existing methods**

Early methods were based on iterative data adjustment procedures, whereby measurements were successively deleted from the measurement data set. Gross errors could subsequently be identified through association with the maximum effect of such a deletion on a least squares objective function, but the method was cumbersome, especially when applied to large sets of measurements (Romagnoli and Stephanopoulos, 1981). These methods for detecting the presence of systematic errors were later validated statistically, based on the relation between the residuals of the constraints and the measurement errors. Further advances followed with the proposal of univariate and multivariate statistical criteria for detection not only of the presence of gross errors in the data set as a whole, but also of the locations of these errors (Romagnoli and Stephanopoulos, 1981; Crowe et al., 1983; Knepper and Gorman, 1980; Madron et al., 1977). These methods were only applicable to measurement data subject to linear constraints and non-linear constraints had to be linearized, typically by retaining first order terms in a Taylor series expansion (Crowe et al., 1986; Romagnoli and Stephanopoulos, 1980, 1981; Stephenson and Shewchuck, 1986). Linearization procedures such as these are not always successful however, especially where highly non-linear systems are concerned (Kim et al., 1990).

Although the principle on which all these tests was based remained essentially the same, many refinements to these tests were proposed in subsequent years. Serth and Heenan (1986) for example, proposed a screened combinatorial test, as well as a modified iterative measurement test which they applied to measurement data subject to bilinear constraints. Iordache et al.
(1985) similarly proposed a modified test for the identification of multiple gross errors. Narasimhan and Mah (1987, 1989) recommended the use of a generalized likelihood ratio test, which could accommodate errors not only attributable to erroneous measurements, but also to actual deviations in the process itself, while Rollins and Davies (1992) suggested the use of an unbiased method to detect systematic errors. The sophistication of current statistical procedures notwithstanding, these methods all suffer from a serious shortcoming. They are all inherently limited in their applicability to data restricted by linear process constraints. That is not to say these techniques cannot be applied to data subject to non-linear process constraints at all. Serth and Heenan (1986) and others have proved that under certain circumstances (such as where the process constraints can be linearized successfully) the application of these statistical methods yields reasonably satisfactory results (Crowe et al., 1986; Romagnoli and Stephanopoulos, 1980, 1981).

In a new approach Kramer (1992) has recently shown that autoassociative neural networks can be implemented to detect and eliminate gross errors in measurement data subject to non-linear constraints. The disadvantage of these nets is that they depend on a large degree of redundancy in the measurement data, and are therefore not suitable for the detection and elimination of gross errors in singular variable measurements, or measurements characterized by small sample sizes, such as are frequently encountered in the metallurgical industry, where the independent measurement of process variables is often difficult and expensive. Error classification by autoassociative neural nets also depends on the relative distribution of errors in the samples. If for example two out of three variable measurements contain biased or gross errors, the autoassociative net will incorrectly characterize the unbiased error as biased, since it does not have an absolute reference regarding the features of a gross error.

It is consequently the aim of this paper to highlight the use of alternative neural net methods to detect gross errors in measurement data. These nets make use of the constraint residuals of the process system and like autoassociative neural nets (and contrary to classical statistical methods), they also have a powerful ability to detect gross errors in the presence of non-linear constraints. Redundant variable measurements are not a prerequisite for the use of these nets, however, and they are better suited for the classification of multiple errors of various types.

THE DETECTION OF SYSTEMATIC ERRORS BY MEANS OF BACK PROPAGATION NEURAL NETS

Artificial neural net models or parallel processing models are characterized by the dense interconnection of primitive computational elements or artificial neurons, and have enjoyed remarkable success as classifiers. Comprehen-
sive in-depth discussions on neural nets abound in the literature and only a very brief overview of the basics is provided in this paper (Hecht-Nielsen, 1990; Lippman, 1987, 1989; Rumelhart et al., 1986; Wasserman, 1989).

A back propagation neural net consists of a number of primitive process units arranged in sequential layers. In general the net is comprised of an input and an output layer, which may be separated by one or more hidden layers. Units in different layers are interconnected and the strengths of these connections are described by the weight matrix of the neural net. Each processing element (shown diagrammatically in Fig. 1) can have an arbitrary number of input connections, but only one output connection (that can branch out to form a multiple output connection). The input layer does not process the data, but merely serves to distribute the data to the next layer. In all other layers succeeding the input layer, the inputs are weighted, summed and transformed by means of an activation function particular to a process unit. These activation functions may take any form, but are typically non-linear functions that map or squash the input space to a smaller output space, such as \([0; 1]\) or \([-1; 1]\). In principle this means that neural nets can be used to map any continuous input space to any related continuous output space to an arbitrary degree of accuracy (Hornik et al., 1989).

Back propagation nets can be trained by repeatedly presenting them with examples of inputs and desired outputs (Bhat et al., 1990; Bhat and McAvoy, 1990; Hecht-Nielsen, 1990; Hinton, 1989; Hornik et al., 1989; Leonard and Kramer, 1990; Lippman, 1987, 1989; Rumelhart et al., 1986; Wasserman, 1989). Training, which entails the modification of the elements of the weight matrix of the net, occurs by means of learning algorithms designed to minimize the mean square error between the desired and the actual output of the net (Bhat and McAvoy, 1990). The net learns by adjusting its connection

\[
OUTPUT = F\left( \sum_{i=1}^{n} w_i \cdot INPUT_i \right)
\]

\[
F( x ) = \frac{1}{1 + e^{-x}}
\]

Fig. 1. A typical process element in a neural net.
weights, based on the discrepancy between the net’s computed output and the desired output. In this way the net forms an internal representation of the relationship between the input and the output data presented to it. If these data are sufficiently representative of the relationship between input and output, the net also learns to generalize this relationship.

Computation in back propagation neural nets is feedforward and synchronous, i.e. the states of the process units in lower levels or layers of the net are updated before units in layers further down in the net. The activation rule of a process unit \( i \) in layer \( l \) of the neural net is typically of the form

\[
z_{i,l}(t+1) = g \left[ \sum_j w_{ij} \cdot z_{j,l-1}(t) - \Theta_i \right]
\]

with \( 1 \leq i \leq N_i \) and \( 1 \leq j \leq N_{l-1} \). The form of the transfer function \( g \) may vary, but it could be a linear, step or sigmoidal transfer function, among others, with a domain typically much smaller than that of the potential (i.e. the sum of the weighted inputs) of the process unit.

The training of back propagation neural nets is an iterative process and can be accomplished by any suitable procedure (such as a gradient descent method), in order to minimize an error criterion, that is

\[
w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}
\]

where

\[
\Delta w_{ij} = -\tau \cdot \partial\epsilon / \partial w_{ij}
\]

with \( 1 \leq i \leq N_i \) and \( 1 \leq j \leq N_{l-1} \), where \( \tau \) is the learning rate and \( \epsilon \) the error criterion, i.e.

\[
\epsilon = \frac{1}{2} \sum_j (d_j - z_j)^2
\]

based on the difference between the desired \( (d_j) \) and the actual outputs \( (z_j) \) of the unit.

The neural nets used in this investigation were simulated on a 486DX 33 MHz IBM compatible personal computer. Although the programming of the nets can be done without too much effort, it is often convenient to make use of commercial software packages, such as NeuralWorks Professional II, which provides a flexible and sophisticated framework for the development of neural net applications.

**Strategy for the detection of systematic errors**

By presenting a back propagation neural net with examples of process variable measurements and process constraint residuals as input, and appropriate categories denoting the classification of the residuals, the net can be trained...
to generalize the relationship between the measurement errors and the constraint residuals. By training the net with exemplars consisting of these measurement errors and constraint residuals, as well as criteria or labels for the classification of these types of errors, it learns to generalize this relationship. When presented with a test vector, it is thus able to classify the residuals into the categories it had been trained to recognize (analogous to the statistical hypothesis tests traditionally used to categorize errors). In the simplest case the classification of the errors can only consist of distinguishing between the collective presence of random and gross errors, without an indication of the sources of these errors. In more sophisticated variations, the classification scheme can be designed to discriminate not only between the locations and types of various errors, but also between gross errors of various magnitudes.

**Specification of the neural net structure**

The number of processing elements of the input layer of the net is equal to the sum of process variables and their associated process constraints. Since the elements in the input layer do not actually operate on the input data, but merely distribute them to the following layers in the net, the activation functions of these layers are linear.

The number of hidden layers, the number of processing elements in each hidden layer, as well as the specific activation functions associated with each of these elements generally have to be determined by trial and error during training of the net. The number of process elements in the output layer is equal to the number of process or flow variables, and correspond to the input elements in the input layer which represent process or flow variables. In the authors’ experience, it is usually best to start off with as simple a structure as possible, i.e. zero hidden layers or only one hidden layer with as few elements as possible, and to gradually expand the structure by adding more elements until the net performs satisfactorily. During this trial and error procedure, various activation functions and training strategies should also be evaluated.

**Generation of training data**

For the purpose of this investigation, the nets were trained by means of synthetic data, but experimental data, if available, could also be used. Training data were generated in spreadsheet files. Spreadsheet macros first generated sets of consistent measurement vectors, which were consequently corrupted by random and systematic errors, while appropriate adjustments were also made to the error labels (zero for random and one for systematic errors). These corrupted data files were then exported to ASCII data files which could be linked to the neural net simulators.
Training of the neural net

The neural nets were trained by presenting them repeatedly with the exemplars in the training data files. The performance of the net was continuously monitored in terms of the average root-mean-square (RMS) output of the net. When the RMS values of the net complied with a suitable convergence criterion, or appeared to have reached minimum values, training was terminated. The nets were then tested against exemplars not previously presented to them.

Use of the neural net to detect gross errors

When provided with input data consisting of plant measurements, the trained nets generated output data which labelled the input measurements according to the type of error assigned to these inputs, i.e. zero for negligible or random errors, and one for systematic errors.

EXAMPLES

Example 1: location of multiple gross errors in an industrial flotation circuit

The flotation circuit depicted in Fig. 2 has previously been described in the literature (Cutting, 1976) and consists of 12 process units, viz. 6 flotation banks \((R_1, R_2, \ldots, R_6)\), 5 hydrocyclones and a mill. Since only the flow rates of the process streams, \(W_1, W_2, \ldots, W_{19}\) are considered, the effect of the mill can be ignored. The circuit is thus subject to 11 linear process constraints (Eqs. 10–20) and since measurements of the flow variables generally violate these constraints, they have to be adjusted prior to further use. As part of the reconciliation procedure, it is necessary to detect and eliminate gross errors.

Fig. 2. A flotation circuit with linear process constraints (example 1).
in the flow variable measurements, as the presence of these errors can lead to large distortions in the reconciled values of the variables. Knowledge of the variances of these measurements is furthermore a prerequisite to the detection of systematic errors, as it is used to differentiate between the different classes of errors. Because the variances of the measurements have not been available, and the purpose of the example is to demonstrate the use of an artificial neural net to detect systematic errors in a set of constrained measurements, arbitrary variances have been assumed for the measurement errors.

PROCESS CONSTRAINTS \((C_n)\)

**Flotation banks**

\[ C_1: \quad W_1 - W_2 - W_3 = 0 \]  
\[ C_2: \quad W_4 - W_5 - W_6 = 0 \]  
\[ C_3: \quad W_9 - W_{10} - W_{11} = 0 \]  
\[ C_4: \quad W_{12} - W_{13} - W_{14} = 0 \]  
\[ C_5: \quad W_{15} - W_{16} - W_{17} = 0 \]  
\[ C_6: \quad W_{16} - W_{18} - W_{19} = 0 \]  

**Hydrocyclones**

\[ C_7: \quad W_3 + W_{11} - W_4 = 0 \]  
\[ C_8: \quad W_6 - W_7 - W_8 = 0 \]  
\[ C_9: \quad W_5 + W_{14} - W_9 = 0 \]  
\[ C_{10}: \quad W_{10} + W_{17} - W_{12} = 0 \]  
\[ C_{11}: \quad W_{13} + W_{18} - W_{15} = 0 \]

The adjusted data, shown in Table 1, were consequently used as a basis for

**TABLE 1**

<table>
<thead>
<tr>
<th>(W_1)</th>
<th>(W_2)</th>
<th>(W_3)</th>
<th>(W_4)</th>
<th>(W_5)</th>
<th>(W_6)</th>
<th>(W_7)</th>
<th>(W_8)</th>
<th>(W_9)</th>
<th>(W_{10})</th>
<th>(W_{11})</th>
<th>(W_{12})</th>
<th>(W_{13})</th>
<th>(W_{14})</th>
<th>(W_{15})</th>
<th>(W_{16})</th>
<th>(W_{17})</th>
<th>(W_{18})</th>
<th>(W_{19})</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.418</td>
<td>2.950</td>
<td>0.468</td>
<td>3.492</td>
<td>1.808</td>
<td>1.684</td>
<td>0.134</td>
<td>1.549</td>
<td>4.882</td>
<td>3.407</td>
<td>1.475</td>
<td>3.660</td>
<td>3.557</td>
<td>0.123</td>
<td>4.523</td>
<td>4.251</td>
<td>0.273</td>
<td>3.284</td>
<td>0.966</td>
</tr>
</tbody>
</table>
generating artificial measurements, by corrupting the consistent set of measurements by various random and systematic errors. In this investigation all errors had a relative variance of 0.0134, so that random and systematic errors were differentiated solely in terms of their expected values, as shown in Fig. 3. (Corruption of a measurement with an expected value of $a$, with an error with a relative bias of 1.4, would mean that the expected value of the biased measurement would be $1.4a$. The relative variance also referred to in Table 2, is the variance associated with a measurement with a unitary value, so that the actual variance of a measurement with an expected value of $a$ would be $0.0134 \cdot a^2$.)

Since the constraints are linear, a traditional statistical method, such as the popular measurement test could also have been used to determine the presence of gross errors in the process variables (Iordache et al., 1985).

![Graph showing frequency and variance](image)

**Fig. 3. Artificial errors introduced into the training and test sets of exemplars.**

**TABLE 2**

<table>
<thead>
<tr>
<th>Case</th>
<th>$E(\text{UM})/E(\text{CM})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>1.225</td>
</tr>
<tr>
<td>C</td>
<td>1.25</td>
</tr>
<tr>
<td>D</td>
<td>1.3</td>
</tr>
<tr>
<td>E</td>
<td>1.4</td>
</tr>
<tr>
<td>F</td>
<td>1.5</td>
</tr>
<tr>
<td>G</td>
<td>1.7</td>
</tr>
<tr>
<td>H</td>
<td>2</td>
</tr>
</tbody>
</table>

$^a$All measurements had relative variances of 0.0134.
A single layer back propagation net (shown diagrammatically in Fig. 4) consisting of an input layer with 30 computational elements (corresponding to the 19 process variables and the 11 process constraints) and an output layer consisting of 19 computational elements (corresponding to the 19 process variables) was constructed to identify gross errors in the measured values of the flow streams $W_1, W_2, \ldots, W_{19}$. The states of the computational elements in the output layer of the net (one element for each measured variable) indicate the presence (output value = 1) or the absence (output value = 0) of a gross error. It is in principle also possible to distinguish between systematic errors with different biases or expected values, by expanding the domain of the states of these elements. Irrelevant connections were eliminated (such as those between variables and constraint equations not comprised of these variables) to make the net more efficient. The pruning of these connections is not essential to the successful implementation of smaller nets such as the net used in this example, since the net automatically assigns negligibly small values to connections between unrelated variables and constraints after sufficient training. This may not necessarily be the case for larger nets, and can result in suboptimal weight matrices in these nets, as well as an inferior capability to detect systematic errors in variable measurements.

The set of exemplars consists of feature vectors of the type

$$V_k = \{W_i, C_p | E_i\}$$
where
\[ i=1,2, \ldots, 11 \quad \text{and} \quad p=1,2, \ldots, 6 \]
i.e. the inputs consist of the measurement values of the flow variables in the system \((W_i)\), the values of the constraint residuals associated with the values of these flow variables \((C_p)\), as well as an indication of the type of error associated with a particular measurement value \((E_i)\).

Approximately 100 artificially generated exemplars were needed to adequately train the neural net, as indicated in Fig. 5. After approximately 10000 training cycles, which involved the presentation of each vector in the training set to the net, very little improvement in the root mean square error (difference between actual and the desired output of the net) occurred, as shown in Fig. 6. The performance of the net could consequently be evaluated against the test set of vectors and is depicted graphically in Fig. 7. The labels A–G shown in Fig. 7 denote the corruption of the measurement data with errors with different biases as explained in Table 2. Biases are shown relative to the measurements. In this figure it can be seen that the net classified all gross errors correctly when the relative bias of the systematic error was larger than approximately 70%. These values are not absolute, since the performance of the net is also determined by the variance of the errors (which was fixed.

![PERCENTAGE OF SYSTEMATIC ERRORS IDENTIFIED](https://scholar.sun.ac.za)

**Fig. 5.** Relation between the performance of the neural net and the size of the training set.
THE USE OF NEURAL NETS TO DETECT SYSTEMATIC ERRORS IN PROCESS SYSTEMS

RMS ERROR OF OUTPUTS

0.3
0.25
0.2
0.15
0.1
0.05
0.0
0
0 1 2 3 4 5 6 7 8 9 10

NUMBER OF TRAINING CYCLES (Thousands)

Fig. 6. Performance of net during training.

PERCENTAGE ERRORS IDENTIFIED

100
80
60
40
20
0

% BIAS IN SYSTEMATIC ERROR

0 10 20 30 40 50 60 70 80 90 100

Fig. 7. Ability of net to detect systematic errors with different biases.
Table 3
Weights of sparsely connected trained neural net used for the detection of gross errors in flow variables of flotation circuit (see Fig. 4)

<table>
<thead>
<tr>
<th>Connection(^a)</th>
<th>(W_{1}-EW_{1})</th>
<th>(DW_{1}-EW_{1})</th>
<th>(B-EW_{2})</th>
<th>(W_{2}-EW_{2})</th>
<th>(DW_{1}-EW_{2})</th>
<th>(DW_{2}-EW_{2})</th>
<th>(B-EW_{3})</th>
<th>(W_{3}-EW_{3})</th>
<th>(DW_{1}-EW_{3})</th>
</tr>
</thead>
<tbody>
<tr>
<td>(-2.4388)</td>
<td>(3.2736)</td>
<td>(0.9215)</td>
<td>(-1.3816)</td>
<td>(2.8042)</td>
<td>(-1.4825)</td>
<td>(0.4902)</td>
<td>(-1.9175)</td>
<td>(3.6914)</td>
<td>(-0.2169)</td>
</tr>
<tr>
<td>(DW_{2}-EW_{3})</td>
<td>(B-EW_{4})</td>
<td>(W_{4}-EW_{4})</td>
<td>(DW_{2}-EW_{4})</td>
<td>(DW_{2}-EW_{4})</td>
<td>(B-EW_{5})</td>
<td>(W_{5}-EW_{5})</td>
<td>(DW_{4}-EW_{5})</td>
<td>(B-EW_{6})</td>
<td></td>
</tr>
<tr>
<td>(-0.0491)</td>
<td>(-1.1540)</td>
<td>(2.2959)</td>
<td>(-1.6824)</td>
<td>(0.7057)</td>
<td>(-1.6658)</td>
<td>(3.3520)</td>
<td>(-0.9344)</td>
<td>(0.1855)</td>
<td>(-1.7126)</td>
</tr>
<tr>
<td>(W_{6}-EW_{6})</td>
<td>(DW_{3}-EW_{6})</td>
<td>(DW_{4}-EW_{6})</td>
<td>(B-EW_{7})</td>
<td>(W_{7}-EW_{7})</td>
<td>(DW_{3}-EW_{7})</td>
<td>(B-EW_{8})</td>
<td>(W_{8}-EW_{8})</td>
<td>(DW_{3}-EW_{8})</td>
<td></td>
</tr>
<tr>
<td>(3.1374)</td>
<td>(0.5339)</td>
<td>(-1.0080)</td>
<td>(-1.9375)</td>
<td>(3.5752)</td>
<td>(-0.2845)</td>
<td>(-1.5842)</td>
<td>(3.3627)</td>
<td>(0.1706)</td>
<td>(-1.0864)</td>
</tr>
<tr>
<td>(B-EW_{9})</td>
<td>(W_{9}-EW_{9})</td>
<td>(DW_{5}-EW_{9})</td>
<td>(DW_{6}-EW_{9})</td>
<td>(B-EW_{10})</td>
<td>(W_{10}-EW_{10})</td>
<td>(DW_{6}-EW_{10})</td>
<td>(DW_{7}-EW_{10})</td>
<td>(W_{11}-EW_{11})</td>
<td></td>
</tr>
<tr>
<td>(-1.0775)</td>
<td>(2.4308)</td>
<td>(-1.8186)</td>
<td>(0.7896)</td>
<td>(-1.5646)</td>
<td>(3.1763)</td>
<td>(-1.2289)</td>
<td>(0.4387)</td>
<td>(-1.7214)</td>
<td>(3.4500)</td>
</tr>
<tr>
<td>(DW_{2}-EW_{11})</td>
<td>(DW_{6}-EW_{11})</td>
<td>(B-EW_{12})</td>
<td>(W_{12}-EW_{12})</td>
<td>(DW_{2}-EW_{12})</td>
<td>(DW_{8}-EW_{12})</td>
<td>(B-EW_{13})</td>
<td>(W_{13}-EW_{13})</td>
<td>(DW_{8}-EW_{13})</td>
<td></td>
</tr>
<tr>
<td>(0.1135)</td>
<td>(-0.7770)</td>
<td>(-1.5393)</td>
<td>(2.6934)</td>
<td>(-1.3527)</td>
<td>(0.7025)</td>
<td>(-1.5235)</td>
<td>(3.0520)</td>
<td>(-1.3367)</td>
<td>(0.3939)</td>
</tr>
<tr>
<td>(B-EW_{14})</td>
<td>(W_{14}-EW_{14})</td>
<td>(DW_{5}-EW_{14})</td>
<td>(DW_{8}-EW_{14})</td>
<td>(B-EW_{15})</td>
<td>(W_{15}-EW_{15})</td>
<td>(DW_{5}-EW_{15})</td>
<td>(DW_{10}-EW_{15})</td>
<td>(B-EW_{16})</td>
<td></td>
</tr>
<tr>
<td>(-1.8867)</td>
<td>(3.7894)</td>
<td>(-0.2376)</td>
<td>(-0.1518)</td>
<td>(-1.2379)</td>
<td>(2.5626)</td>
<td>(-1.5544)</td>
<td>(0.6490)</td>
<td>(-1.4959)</td>
<td>(2.8072)</td>
</tr>
<tr>
<td>(DW_{10}-EW_{16})</td>
<td>(DW_{11}-EW_{16})</td>
<td>(B-EW_{17})</td>
<td>(W_{17}-EW_{17})</td>
<td>(DW_{7}-EW_{17})</td>
<td>(DW_{10}-EW_{17})</td>
<td>(B-EW_{18})</td>
<td>(W_{18}-EW_{18})</td>
<td>(B-EW_{19})</td>
<td></td>
</tr>
<tr>
<td>(-1.5080)</td>
<td>(0.8062)</td>
<td>(-1.6237)</td>
<td>(3.4103)</td>
<td>(-0.1807)</td>
<td>(-0.6249)</td>
<td>(-1.3155)</td>
<td>(3.2951)</td>
<td>(-1.3006)</td>
<td>(-1.5420)</td>
</tr>
<tr>
<td>(W_{19}-EW_{19})</td>
<td>(DW_{9}-EW_{19})</td>
<td>(DW_{11}-EW_{19})</td>
<td>(-0.2980)</td>
<td>(-0.6374)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\(^a\) \(B\) — bias element connected to all computational elements in the output layer. \(W_{1}-W_{19}\) and \(DW_{1}-DW_{11}\) — computational elements in the input layer. \(EW_{1}-EW_{19}\) — computational elements in the output layer.
Throughout this investigation at approximately 0.0134). For systematic errors with expected values not much different from those of the actual measurement themselves (less than 40%), the discriminatory power of the net dropped progressively, as could be expected. The weights of a typical net used to detect gross errors in the flotation circuit with a relative bias of 100% are shown in Table 3. In Table 3, \( B \) refers to the bias element which is fully connected to the output layer, while \( E_i \) refers to the process elements in the input layer associated with flow variable measurements, \( D_W \) refers to the process elements in the input layer associated with constraint residuals, and \( E_W \) refers to the process elements in the output layer. Although the value of any particular weight is not significant as far as the net’s ability to identify gross errors is concerned, it is an indication of the effects of input variables on output variables. The net forms an internal distributed representation of the relationship between the errors in the measurements and the constraint residuals, and it is the collective action of all the process nodes and their associated weights which allows the net to identify the different errors. As can be seen from the dominant diagonal of weight elements in the trained net (shown in Table 3), the net has learnt to ignore the relationships between unrelated variables.
Example 2: location of multiple gross errors in the measurement data of a three-stage backfill circuit subject to non-linear process constraints

Example 2 is based on a backfill circuit which consists of three hydrocyclones connected as indicated in Fig. 8 and which is used for the preparation of backfill material in a South African mine (Woollacott et al., 1992). Although the plant data both before and after reconciliation are provided in Table 4, the measured data could only be tested for gross errors in an arbitrary way, since no knowledge of the covariation matrices of these measurements was available. In order to elucidate the use of neural nets for the detection of gross errors, an example is given below.

### Table 4

<table>
<thead>
<tr>
<th>Cyclone</th>
<th>Size or % solids</th>
<th>( W^9 )</th>
<th>( W^2 )</th>
<th>( W^3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Exp</td>
<td>Calc</td>
<td>Exp</td>
<td>Calc</td>
</tr>
<tr>
<td>I</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>150</td>
<td>5.4</td>
<td>5.6</td>
<td>15.9</td>
<td>16.5</td>
</tr>
<tr>
<td>106</td>
<td>16.4</td>
<td>15.4</td>
<td>33.5</td>
<td>32.5</td>
</tr>
<tr>
<td>75</td>
<td>14.8</td>
<td>13.9</td>
<td>16.4</td>
<td>17.1</td>
</tr>
<tr>
<td>53</td>
<td>9.1</td>
<td>9.7</td>
<td>9.2</td>
<td>8.2</td>
</tr>
<tr>
<td>38</td>
<td>4.0</td>
<td>5.9</td>
<td>3.8</td>
<td>3.9</td>
</tr>
<tr>
<td>0</td>
<td>50.3</td>
<td>50.5</td>
<td>21.2</td>
<td>21.8</td>
</tr>
<tr>
<td>solids</td>
<td>49.3</td>
<td>47.5</td>
<td>61.8</td>
<td>62.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cyclone</th>
<th>Size or % solids</th>
<th>( W_2 )</th>
<th>( W_4 )</th>
<th>( W_5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Exp</td>
<td>Calc</td>
<td>Exp</td>
<td>Calc</td>
</tr>
<tr>
<td>II</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>150</td>
<td>15.9</td>
<td>16.5</td>
<td>18.6</td>
<td>17.2</td>
</tr>
<tr>
<td>106</td>
<td>33.5</td>
<td>32.5</td>
<td>33.5</td>
<td>34.7</td>
</tr>
<tr>
<td>75</td>
<td>16.4</td>
<td>17.1</td>
<td>15.2</td>
<td>15.2</td>
</tr>
<tr>
<td>53</td>
<td>9.2</td>
<td>8.2</td>
<td>8.2</td>
<td>8.5</td>
</tr>
<tr>
<td>38</td>
<td>3.8</td>
<td>3.9</td>
<td>2.5</td>
<td>2.3</td>
</tr>
<tr>
<td>0</td>
<td>21.2</td>
<td>21.8</td>
<td>22.0</td>
<td>22.0</td>
</tr>
<tr>
<td>solids</td>
<td>61.8</td>
<td>62.0</td>
<td>61.7</td>
<td>62.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cyclone</th>
<th>Size or % solids</th>
<th>( W_3 )</th>
<th>( W_7 )</th>
<th>( W_8 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Exp</td>
<td>Calc</td>
<td>Exp</td>
<td>Calc</td>
</tr>
<tr>
<td>III</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>150</td>
<td>1.0</td>
<td>1.4</td>
<td>12.4</td>
<td>12.0</td>
</tr>
<tr>
<td>106</td>
<td>8.7</td>
<td>8.8</td>
<td>35.9</td>
<td>36.2</td>
</tr>
<tr>
<td>75</td>
<td>12.7</td>
<td>12.7</td>
<td>17.0</td>
<td>17.1</td>
</tr>
<tr>
<td>53</td>
<td>12.4</td>
<td>10.2</td>
<td>9.5</td>
<td>9.6</td>
</tr>
<tr>
<td>38</td>
<td>6.2</td>
<td>6.7</td>
<td>3.7</td>
<td>3.5</td>
</tr>
<tr>
<td>0</td>
<td>59.0</td>
<td>39.8</td>
<td>21.5</td>
<td>21.6</td>
</tr>
<tr>
<td>solids</td>
<td>42.3</td>
<td>43.5</td>
<td>23.0</td>
<td>23.0</td>
</tr>
</tbody>
</table>
gross errors in processes subject to non-linear constraints, the adjusted data have been corrupted in stead, so that the performance of the nets could be tested accurately against the known errors.

The material balance of the circuit is expressed by Eqs. (16–25). These equations constitute the constraints on the process system \((C_1-C_{35})\), the residuals of which are incorporated in the training data set of the neural net.

**PROCESS CONSTRAINTS**

\[
\begin{align*}
C_1: & \quad W_1 + W_5 - W_6 = 0 \quad (16) \\
C_2: & \quad W_6 + W_7 - W_9 = 0 \quad (17) \\
C_3: & \quad W_9 - W_2 - W_3 = 0 \quad (18) \\
C_4: & \quad W_3 - W_8 - W_7 = 0 \quad (19) \\
C_5: & \quad W_2 - W_4 - W_5 = 0 \quad (20)
\end{align*}
\]

and for \(i=1\) to \(6\)

\[
\begin{align*}
C_6 - C_{11}: & \quad W_1 \cdot X_{1i} + W_5 \cdot X_{5i} - W_6 \cdot X_{6i} = 0 \quad (21) \\
C_{12} - C_{17}: & \quad W_6 \cdot X_{6i} + W_7 \cdot X_{7i} - W_9 \cdot X_{9i} = 0 \quad (22) \\
C_{18} - C_{23}: & \quad W_9 \cdot X_{9i} - W_2 \cdot X_{2i} - W_3 \cdot X_{3i} = 0 \quad (23) \\
C_{24} - C_{29}: & \quad W_3 \cdot X_{3i} - W_8 \cdot X_{8i} - W_7 \cdot X_{7i} = 0 \quad (24) \\
C_{30} - C_{35}: & \quad W_2 \cdot X_{2i} - W_4 \cdot X_{4i} - W_5 \cdot X_{5i} = 0 \quad (25)
\end{align*}
\]

where

\[\sum_j \sum_i X_{ij} = 1, \quad i=1,2,\ldots,5, \quad j=1,2,3\]

As in the previous example, the reconciled values of all the flow variables \(W_i\) and \(X_{ij}\) are used as a basis for the demonstration of gross error detection by means of neural nets. In order not to unduly complicate computational procedures, the reconciled data have been corrupted artificially with only two classes of errors (i.e. random and systematic). In principle the number of different error classes that a net would be able to handle are limited only by the capabilities of the computational device.

In order to determine the presence of systematic errors, a series of five back propagation neural nets, one for each nodal point in the hydrocyclone circuit (nodes A and B and cyclones I, II and III shown in Fig. 8) is used to categorize the two types of errors, having the same (Gaussian) distributions and differing only with regard to their expected values or biases, i.e. either zero or
non-zero. The sign of the bias is arbitrary and was assumed to be positive in this investigation.

Although the use of five smaller nets reduces the computational load drastically, the suppression of the interactions of the between the five nets limits the capacity of the nets to isolate systematic errors. In order to overcome this drawback, the series of nets can be supervised by a control subroutine designed to interpret the outputs of all five nets simultaneously. In this way the interactions between the nets can be exploited fully, without having to deal with the computational burden presented by a single large net covering all the circuit nodes. Experiments indicated that at least 7 to 10 times as many exemplars were needed to train the larger net to perform with the same accuracy as the system of five smaller connected nets, and even then the larger net did not perform as consistently as the smaller nets.

The nets are structured to form internal representations of the relationships between all the flow variables and process constraints at each node in the process circuit. Since each of the five nodes is situated at the confluence of three process streams, and each stream \( W_i \) is in turn composed of six particle size fractions \( X_{i1}-X_{i6} \), the structures of the five nets used to detect gross errors in the circuit are identical. The input layers of the five nets are thus each comprised of 28 input units, corresponding to each of the 21 state variables and the 7 process constraints at each circuit node, while the output layers consist of 21 units each, corresponding to the 21 flow variables. This constitutes a more efficient design than that of a single large net, since a large number of irrelevant connections are eliminated prior to training of the neural net system.

Two variants of the series of nets have been investigated. In the first, all connections associated with unrelated flow variables and process constraints were eliminated. The performance of this series of sparsely connected nets was subsequently compared to that of a similar series in which all input and output elements were fully connected.

The use of different transfer functions was investigated, and although reasonably good results were also obtained with a hyperbolic tangent transfer function, a sigmoidal transfer function (Eq. 26) was found to be the most effective, i.e.

\[
g(u_i) = \frac{1}{1 + \exp(-u_i)}
\]  

(26)

Prior to the detection of gross errors in the circuit, the nets were trained with data approximating the actual state of the process system. If the actual value of a system or state variable \( S \) had been \( s \), the net was trained with data in a domain enclosing \( s \), such as \( s \pm R \), where \( R \) could be an arbitrary interval (perhaps equal to three times the standard deviation of the measurement er-
error, or simply an interval deemed large enough so as to ensure that it encompasses the actual value of the variable). Since $s$ was not known exactly in this case, it had to be estimated (an exact estimate is not required) and it was consequently convenient to use the adjusted value as an estimate.

Training exemplars were thus generated artificially by the addition of random noise to the measured values of the system variables. Some of these variables were also provided with a bias, in order to simulate systematic errors in the data, as shown in Fig. 3. Each of the measurements in each of the input vectors were randomly corrupted and approximately 15–20% of the errors in each vector were biased. The output portion of the exemplar consisted of convenient numeric labels to classify the error represented by the input data. Since only two types of errors were simulated in this case, binary values (0 and 1) were used to indicate the absence or presence of a gross error.

That is

$$V_k = \{ W_i, X_{ij}, C_p \mid E_q \}$$

(27)

where $i = 1,2,\ldots,5$, $j = 1,2,3$, $p = 1,2,\ldots,12$ and $q = 1,2,\ldots,i \times (j+1)$. The process constraints constituted by the set of equations (27) did not form part of the exemplars, after a preliminary investigation had indicated that their explicit incorporation contributed only marginally to the performance of the net.

After convergence the performance of the nets was evaluated against differ-

---

**Fig. 9.** Ability of neural net system to detect systematic errors with different biases.
ent test sets, containing the same types of errors the nets had been trained to recognize. These vectors consisted of a set of input values, as well as a set of actual output values which could be compared with the output values generated by the net. The performance of the system of five smaller sparsely connected nets is shown in Fig. 9. As is the case with Fig. 7, the labels A–F shown in Fig. 9 denote the biases in the errors with which the measurement data were corrupted. These biases are explained in more detail in Table 2.

The error classification process could fail in two different ways. First of all a gross error could pass unrecognized and be classified as a random error with no bias (analogous to a statistical type I error), and secondly a random error could be mistaken for a gross error (analogous to a statistical type II error). Since very few errors of type II occurred during evaluation, these results are not incorporated in Fig. 9.

The Hinton diagram of the sparsely connected net associated with the primary cyclone is shown in Fig. 10. A Hinton diagram is a convenient means of depicting the weight matrix of the neural net graphically. The coordinates indicate the weights or connection strengths between the processing elements.
solid squares indicate positive values, with magnitudes proportional to the size of the square
empty squares indicate negative values, with magnitudes proportional to the size of the square

Fig. 11. Hinton diagram of the fully connected neural net used to classify the measurement errors in the flow variables of the primary hydrocyclone.

marked on the abscissa and the ordinate. Positive values are denoted by solid squares, while negative values are denoted by empty squares. The sizes of the squares are an indication of the magnitudes of the connection strengths or weight values of the net.

The diagram shown in Fig. 10 is typical and is distinguished by the formation of a dominant diagonal of large weights, which relate the flow variables with their corresponding locations in the output layer. In Fig. 11 a Hinton diagram of the same net, this time fully connected, is shown. A comparison of the diagrams in Figs. 10 and 11 reveals a similar weight distribution in both the fully and the sparsely connected nets, since negligibly small weights have generally been assigned to irrelevant connections in the fully connected net.

DISCUSSION OF RESULTS

Besides computational efficiency, the single most important advantage of the use of neural net systems in the detection of systematic measurement er-
rors in process systems is their ability to identify these errors in process systems bound by non-linear constraints. Their success can be attributed to the fact that during training of such systems, no assumptions are made with regard to the underlying distributions of the errors. Conventional techniques on the other hand, depend on explicit assumptions concerning the distributions of these errors, and can consequently only be applied effectively to linear systems.

The two examples presented in this paper were intended to demonstrate the use of back propagation neural nets to detect gross errors in measurement data subject to process constraints. In practice more sophisticated training procedures could be adopted, which could be used in conjunction with more sophisticated error classification schemes. In addition to these schemes, provision could also be made for the identification of errors of which the distribution can not be positively classified (owing to the stochastic nature of measurement data, not all data can be classified outright).

It would also be desirable to incorporate additional information in the net (such as equipment and instrument failure histories, previous knowledge about measurement covariances, etc.), either through direct modification of the architecture of the net, or by using the net in conjunction with a knowledge base or another neural system.

The computational burden presented by a large system can be circumvented to a large degree through reduction of the input space of the net, either by elimination of irrelevant connections in the net (such as connection paths between variable measurement residuals and the residuals of constraints not associated with the particular variables), or through reduction of the input space of the net through other methods (Van der Walt et al., 1993).

Owing to their massive parallelism, and the latest advances in analog VLSI implementation techniques (Lippman, 1987), neural nets pose a very attractive means of on-line detection and classification of measurement errors or related deviations in the values of the state variables of process systems.

CONCLUSIONS

The vast majority of schemes for the detection of systematic errors in process circuits is based on statistical hypothesis tests. These techniques are inadequate in that they can not be applied directly to non-linear systems.

Neural nets are not impeded by non-linear constraints, or particular assumptions about the underlying distributions of measurement errors and can be used on linear and non-linear systems alike with great success.

The examples discussed in this paper indicate that

- Single layer back propagation neural nets provide an attractive alternative to traditional statistical procedures for the detection of systematic errors in measured data subject to constraints;

- Other more sophisticated training procedures could be adopted, which could be used in conjunction with more sophisticated error classification schemes.

- Provision could also be made for the identification of errors of which the distribution can not be positively classified (owing to the stochastic nature of measurement data, not all data can be classified outright).

- It would also be desirable to incorporate additional information in the net (such as equipment and instrument failure histories, previous knowledge about measurement covariances, etc.), either through direct modification of the architecture of the net, or by using the net in conjunction with a knowledge base or another neural system.

- The computational burden presented by a large system can be circumvented to a large degree through reduction of the input space of the net, either by elimination of irrelevant connections in the net (such as connection paths between variable measurement residuals and the residuals of constraints not associated with the particular variables), or through reduction of the input space of the net through other methods (Van der Walt et al., 1993).

- Owing to their massive parallelism, and the latest advances in analog VLSI implementation techniques (Lippman, 1987), neural nets pose a very attractive means of on-line detection and classification of measurement errors or related deviations in the values of the state variables of process systems.
- In contrast to autoassociative neural nets, the nets discussed in this paper do not require multiple measurements of a data variable, and their performance is furthermore not affected by the presence of multiple errors;
- Due to the fact that the training algorithms of neural nets are not dependent on assumptions made about the distributions of the measurement errors, these nets can be used as an efficient means for the detection of systematic errors in non-linear systems;
- Neural nets pose an attractive option for the on-line detection of gross errors in process systems.

SYMBOLS

\( a \) expected value of an arbitrary variable
\( b \) bias of gross error
\( C \) matrix of process constraint coefficients
\( CM \) corrupted measurement; gross error
\( C_p \) \( p \)th constraint of a process system
\( DW_i \) Process constraint: \( W_i - W_{i+1} - W_{i+2} \)
\( DWX_{ij} \) Process constraint: \( W_i \cdot X_{ij} - W_{i+1} \cdot X_{i+1,j} - W_{i+2} \cdot X_{i+2,j} \)
\( d \) desired output value of process unit \( j \)
\( e \) errors in measurements of process variables
\( E(\cdot) \) expected value
\( E_i \) process element in output layer of neural net trained to classify errors in measurement datum \( i \)
\( EW_i \) process element in output layer of neural net trained to classify errors in measurement of flow stream \( W_i \)
\( EX_{ij} \) process element in output layer of neural net trained to classify errors in measurement of component \( X_{ij} \) in flow stream \( W_i \)
\( g \) transfer function of neural net process element
\( N_l \) the number of process elements in the \( l \)th layer of a neural net
\( Q \) value of generalized state variable
\( R \) arbitrary interval length
\( S \) a state variable in general
\( s \) actual value of a state variable \( S \)
\( t \) time
\( UM \) uncorrupted measurement; unbiased error
\( u_i \) potential of process unit \( i \) in neural net
\( V_k \) \( k \)th training vector
\( W_i \) flow rate of stream \( i \)
\( w_{ij} \) connection strength or weight between process units \( i \) and \( j \) in neural net
\( X_{ij} \) mass fraction of component \( j \) in flow stream \( i \)
true values of process variables
measured values of process variables
residuals of process constraints
state of process unit \( i \) in layer \( l \) of a neural net

Greek symbols

error criterion for adjustment of weight matrix of neural net
bias of process unit \( i \) in neural net
learning rate of process unit

REFERENCES


IDENTIFICATION OF GROSS ERRORS IN MATERIAL BALANCE MEASUREMENTS BY MEANS OF NEURAL NETS

C. ALDRICH and J. S. J. VAN DEVENTER
Department of Metallurgical Engineering, University of Stellenbosch, Stellenbosch, 7600, South Africa

(First received 14 October 1992; accepted in revised form 11 October 1993)

Abstract—Reliable sets of steady-state component and total flow rate data form the cornerstone for the monitoring of plant performance. The detection and isolation of gross errors in these data constitute an essential part of the process of reconciliation of the measurement data, which are generally inconsistent with process constraints. By using a neural net to classify measurement or constraint residuals, gross errors in the data can be identified accurately and efficiently. Gross error detection and isolation with artificial neural nets do not require explicit knowledge of the distribution of random errors in measurement values and can be applied to processes with arbitrary constraints.

INTRODUCTION
A clear understanding of the operational behaviour of a plant is essential for the identification of process phenomena, as well as the optimization and control of the plant or process circuit. The collection and analysis of data from processes, whether they are operated in steady state or not, are therefore an important means for evaluating the performance of a plant or an individual process unit. These data are as a rule subject to random noise, or even gross errors, which can be attributed to failure or miscalibration of measuring instruments, inadequate instrumentation, the departure of the process from steady state owing to malfunctioning process equipment, or significant changes in the environment (Hlavacek, 1977).

As a result of these erroneous measurements, any set of process data will generally violate mass and energy conservation requirements, as well as other physical constraints pertaining to the process. Since most of the methods used to reconcile such inconsistent data distribute the error among the measurements, the presence of systematic errors can lead to a severely distorted picture of the actual process (Hodouin and Vaz Coelho, 1987). It is thus crucial that biased errors are identified and eliminated from the process data prior to final reconciliation of variable measurements, and as a result most data reconciliation algorithms contain procedures for the identification of gross errors in the measurement data.

Following detection and identification of their sources, these errors are usually removed or compensated for by some or other means (Serth et al., 1987). Since repeated measurements of a variable would not allow the detection of a systematic error, virtually all gross error detection schemes are based on statistical tests involving the constraint residuals of the measurement errors. These tests are generally valid for systems subject to linear constraints and very few attempts appear to have been made to accommodate nonlinear systems explicitly (Serth et al., 1987; Tjoa and Biegler, 1991).

In this paper a new method is proposed for the detection of gross errors in constrained measurement data. This method makes use of neural nets to classify measurement errors and is not impeded by the nature of the system constraints.

PROBLEM STATEMENT
Typically, the process constraints or conservation equations are described by Crowe (1989):

\[ A \cdot y = 0 \]  

where \( y \) is the \((n \times 1)\) vector of true values of the state variables and \( A \) an \((m \times n)\) constraint matrix of full row rank \( m (n > m) \). If:

\[ y' = y + e \]  

constitutes the \((n \times 1)\) vector of measurements of the true values \( y \), with \((n \times 1)\) error vector \( e \), and covariance matrix \( Q \); then the measured values of the process variables generally violate the process constraints

\[ A \cdot y' = r \neq 0 \]  

or in terms of the true values and error components

\[ A \cdot (y + e) = r \]  

and, if the constraints are linear,

\[ A \cdot (y + e) = A \cdot y + A \cdot e = r, \]  

i.e.

\[ A \cdot e = r. \]

Under the null hypothesis that no systematic errors are present, \( r \) is a multivariate normal with a zero mean (Madron et al., 1977; Romagnoli and Stephanopoulos, 1981; Mah and Tamhane, 1982; Tamhane and Mah, 1985), i.e.

\[ E(r) = E(A \cdot e) = A \cdot E(e) = 0 \]  

and a covariance matrix \( V = \text{cov}(r) \) of the constraint

1Author to whom correspondence should be addressed.
residuals (Crowe, 1988), where
\[
V = E(ee^T) = A \cdot E(ee^T) \cdot A^T = A \cdot Q \cdot A^T. \tag{6}
\]
The alternative hypothesis is that the expected value of \( r \) is not zero, i.e.
\[
E(r) = b \neq 0
\]
which indicates the presence of an error with a bias of magnitude \( b \). These two hypotheses are subsequently evaluated and rejected or accepted on the basis of standard statistical criteria. Note that the data need not be reconciled in order to calculate \( r \) and its derived statistics.

**EXISTING METHODS**

The earliest methods of handling systematic errors in inconsistent measurement data consisted among others of the successive application of data adjustment procedures on subsets of the measurement data set in order to determine the presence and source of gross errors (Romagnoli and Stephanopoulos, 1981). Measurements not contained in the subset which minimized the least-squares objective function were associated with gross errors. These procedures were cumbersome and inefficient, especially when large complex data sets had to be analysed. The use of such methods was placed on a formal statistical foundation soon after, based on the relation between the distribution of the measurement errors and the constraint residuals.

Since this global test does not identify the sources of gross errors, however, its presence, an univariate statistical criterion to identify biased errors in sets of variable measurements containing more than one gross error was proposed, but this method was based on an iterative test permitting the possible cancellation of gross errors (Romagnoli and Stephanopoulos, 1981). This notion was later extended to multivariate chi-square tests and the calculation of covariance matrices for all estimated data (Madron et al., 1977; Knepper and Gorman, 1980; Crowe et al., 1983). Since these methods are only applicable to linearly constrained data, non-linear constraints had to be linearized (Romagnoli and Stephanopoulos, 1980, 1981; Crowe et al., 1986; Stephenson and Shewchuck, 1986). Refinements to these tests were made by Iordache et al. (1985), who modified the test to accommodate the identification of multiple gross errors. More sophisticated procedures were developed when Serth and Heenan (1986) and Serth et al. (1987, 1989) proposed a modified iterative measurement test (MIMT) and a screened combinatorial (SC) test, and compared these with the measurement test proposed by Mah and Tamhane (1982), as well as the method of pseudo-nodes (Crowe et al., 1983).

The disadvantage of all these methods is that they do not distinguish between different types of gross errors, as stated by Narasimhan and Mah (1987, 1989). These authors proposed a generalized likelihood ratio (GLR) method for the identification of gross errors, by modifying the process model to facilitate the detection of gross errors not occurring due to impaired measurement, but to actual deviations in the process itself (Narasimhan and Mah, 1987, 1989; Narasimhan, 1990). The generalized likelihood ratio (GLR) method of Narasimhan and Mah leads to statistical test values which are exactly equal to the square of the corresponding maximum power test. The modified process model is constituted by
\[
A \cdot y = 0 \quad \text{for measurement errors} \tag{1}
\]
\[
A \cdot y - b \cdot m = 0 \quad \text{for leaks,} \tag{7}
\]
These authors derived the statistical properties of the constraint residuals in the presence of both a measurement bias and a leak, and used a generalized likelihood ratio method to identify gross errors associated with these conditions. They showed that generalized likelihood ratio methods provide a framework for the identification of any type of gross error that can be modelled and can also provide estimates of the magnitude of the bias and its consequent impact on the process data. Narasimhan and Mah also introduced a serial compensation strategy to detect multiple gross errors. This approach attempted the successive detection of gross errors from the largest to the smallest, but suffered from the drawback that it could still be subject to large type I errors and low power (Rollins and Davis, 1992). An improvement on this method was consequently proposed by Rollins and Davis (1992), who used a method to make unbiased estimates of gross errors in measurement data.

It has long been realized that all these methods were limited by the restriction of linear process constraints. Methods involving the linearization of non-linear constraints are often only partly successful and do not always capture the essential characteristics of a process (Kim et al., 1990). Serth et al. (1987, 1989) were the first to propose an explicit procedure to deal with non-linear constraints in gross error detection methods, through an extension of their MIMT procedure. It has long since been shown that the distortion caused by gross errors in data reconciliation procedures can be obviated effectively by making use of maximum likelihood ratio methods to rectify these inconsistent data (Britt and Leucke, 1973; Fariss and Law, 1979). These methods do not indicate the presence or sources of gross errors, however. The method was applied to data subject to bilinear constraints (a mild type of non-linearity) and entailed an iterative procedure whereby the data reconciliation problem had to be solved at each iteration step. Although the method was applied successfully to several examples, the computational procedure is cumbersome when large systems are involved, and its effectiveness remains to be demonstrated in the presence of less benevolent types of non-linearities.

In practice, the presence of non-linear constraints impose a severe restriction on traditional methods for the identification of gross errors, and it is the purpose of this paper to highlight the use of neural net methods to detect gross errors in measurement data.
It will be shown that neural nets have both a powerful ability to detect data errors, and an unparalleled ability to detect these errors in measurement data subject to non-linear constraints.

GROSS ERROR DETECTION BY MEANS OF NEURAL NETS

Background

Excellent in-depth discussions on neural nets can be found in the literature and only a very brief overview is provided in this paper (Rumelhart et al., 1986; Lippmann, 1987, 1989; Wasserman, 1989; Hecht-Nielsen, 1990).

A neural net is a parallel distributed information processing structure, consisting of an arrangement of interconnected primitive processing elements. Each processing element can have an arbitrary number of input connections, but only one output connection (that can branch or fan out to form a multiple output connection). Processing elements or artificial neurons can have local memory and also possess transfer functions that can use or alter this memory, process input signals and produce the output signals of the elements, as shown in Fig. 1.

These processing elements are typically divided into disjoint subsets, called layers, in which all the process units in essence possess the same computational characteristics. The layers comprising a neural net are usually categorized as either input, hidden or output layers, to denote the way in which they interact with the information environment of the net.

The back propagation nets used in this study were feedforward networks which could be trained by repeatedly presenting them with examples of inputs and desired outputs (Rumelhart et al., 1986; Lippmann, 1987, 1989; Hinton, 1989; Hornik et al., 1989; Wasserman, 1989; Bhat et al., 1990; Bhat and McAvoy, 1990; Hecht-Nielsen, 1990). Training, which entails the adjustment of the weight matrix of the net, occurs by means of learning algorithms designed to minimize the mean square error between the desired and the actual output of the net (Bhat and McAvoy, 1990). During the learning process, information is propagated back through the net in order to update the connection weights of the net. Through training, the net forms an internal representation of the relationship between the inputs and the outputs presented to it.

Computation in back propagation neural nets is feedforward and synchronous, i.e. information is distributed by the input layer to the first hidden layer (if present) from where it moves through successive layers to the final or output layer. During training, however, the weights of the net are updated in reverse order, i.e. the errors between the outputs of the net and the desired outputs are propagated back through the net so that the weights of the output layer are updated first, followed by the weights of the layer preceding the output layer, down to the weights between the input and the first hidden layer (if present). The activation rules of process units are typically of the form

\[ z_i(t + 1) = g(u_i(t)) \]  

where \( u_i(t) \) designates the potential of a process unit at time \( t \), i.e. the difference between the weighted sum of all the inputs to the unit and the unit bias

\[ u_i(t) = \sum_j w_{ij} z_j(t) - \Theta_i. \]

The form of the transfer function \( g \) may vary, but could be a linear, step or sigmoidal transfer function, among others, with a domain typically much smaller than that of the potential of the process unit, such as \([0; 1]\) or \([-1; 1]\), for example.

The training of back propagation neural nets is an iterative process involving the changing of the weights of the net, typically by means of a gradient descent method, in order to minimize an error criterion, i.e.

\[ w_{ij}(t + 1) = w_{ij}(t) + \Delta w_{ij}, \]

where

\[ \Delta w_{ij} = -\tau \frac{\partial e}{\partial w_{ij}} \]

in which \( \tau \) is the learning rate and \( e \) the error criterion, i.e.

\[ e = \frac{1}{2} \sum (d_{o,j} - z_{o,j})^2 \]

based on the difference between the desired and the actual outputs of the unit.

The use of neural nets for the classification of measurement errors

By presenting a back propagation neural net with examples of measurement and constraint residuals as input, and appropriate categories denoting the classification of the residuals, the net can be trained to generalize the relationship between these residuals and the types of errors present in the data. When presented with test exemplars, it is then able to classify the residuals into the categories it had been trained to recognize (analogous to the statistical hypothesis tests traditionally used to categorize errors). In the simplest case, the classification of the errors can consist of distinguishing between the collective presence of random and gross errors only, without an indication of the sources of these errors, by relating the constraint residuals of the system to the presence of gross errors.

Fig. 1. Processing element of a back propagation neural net.
Alternatively, the detection and isolation of gross errors can be based on an analysis of the measurement residuals as is explained in the following examples.

Type I and type II errors. When statistical hypotheses of populations are tested, two types of errors (referred to in the statistical literature as type I or type II errors) are possible (Walpole and Myers, 1978). A type I error is committed when the null hypothesis is valid, but erroneously rejected (i.e. when a random error is incorrectly identified as a systematic error), and a type II error is committed when the null hypothesis is accepted when it is false (i.e. when a systematic error is not detected). The probability of committing a type I error is known as the level of significance or the size of the critical region of the test, and is usually denoted by \( \alpha \), while the probability of a type II error being committed is usually denoted by \( \beta \). In efficient measurement error detection schemes, the probability of both these errors occurring should be as small as possible.

Similar to statistical tests, the performance of a neural net error detection scheme can also be constructed to minimize the probability of type I or type II errors occurring through the appropriate labelling of training exemplars. To reduce the occurrence of type I errors, only the residuals in region BC in Fig. 2 are designated as gross errors. When the distributions of random and systematic errors overlap (shown schematically in Fig. 2), one critical region can only be adjusted at the expense of another. Stated differently, to ensure that the probability of type II errors occurring is as low as possible, the probability of type I errors occurring has to be increased, as is shown diagrammatically in Fig. 2. In the neural net error detection schemes discussed in this paper, all the nets were trained to significance levels of less than 2%.

### EXAMPLES

**Example 1: measurement data from a two-product classifier containing gross errors of various magnitudes**

Consider a two-product classifier, such as a hydrocyclone or a high-tension roll separator, in which the feed stream \( W_1 \), with component mass fractions \( x_{11} \) and \( x_{12} \), is classified into two product streams, \( W_2 \) with component mass fractions \( x_{21} \) and \( x_{22} \), and \( W_3 \) with component mass fractions \( x_{31} \) and \( x_{32} \). All flow variables \( W_i \) and \( x_{ij} \) are measured and the following mass balance constraints have to be satisfied:

\[
W_1 - W_2 - W_3 = 0 \tag{13}
\]

\[
W_1 x_{11} - W_2 x_{21} - W_3 x_{31} = 0 \tag{14}
\]

\[
W_1 x_{12} - W_2 x_{22} - W_3 x_{32} = 0. \tag{15}
\]

The above system consisting of nine state variables and three bilinear process constraints is typical of measurement data reconciliation problems in chemical or metallurgical process engineering, in which total flow streams and component flows have to be reconciled. In order to demonstrate the detection and isolation of gross errors in the system, an arbitrary but consistent set of values (in effect assumed to be the true values of the variables) is corrupted by both random and gross errors. The corruption of the vectors is allowed to occur on a random basis, so that measurement vectors can contain more than one gross error. All errors have Gaussian distributions with standard deviations of 5%, and differ only with regard to their expected values. Two types of gross errors are considered; one with a bias 20% larger than the expected value of the random variable and one with a bias 50% larger than the expected value of the variable. These errors are depicted in Fig. 3, where it can be seen that the larger systematic error is completely discernible from the random error, while the smaller systematic error is only partially discernible.

**General structure of neural net.** The general structure of the net used to detect or isolate the systematic errors in the data is depicted in Fig. 4. The net consists of an input layer and an output layer, sparsely connected in a feedforward mode (i.e. only corresponding process units in the input and output layers are connected). State changes in the process units are effected through a hyperbolic tangent transfer function, i.e.

\[
g(u_i) = \frac{e^{u_i} - e^{-u_i}}{e^{u_i} + e^{-u_i}}, \tag{16}
\]

which maps input to the \([-1:1]\) domain. Sigmoidal
transfer functions, which are commonly used in back propagation nets, were also investigated, but did not perform as well as nets operating with hyperbolic tangent activation functions. The net was trained by means of the generalized delta rule (Rumelhart et al., 1986)

\[ w_{ij}(t + 1) = w_{ij}(t) + k_1 e_i z_j + k_2 m_{ij} \]  
\[ m_{ij} = w_{ij}(t + 1) - w_{ij}(t) \]

so that weight changes were proportional to the error \( e_i \) and the input to the particular connection \( z_j \). A momentum term \( m_{ij} \) equal to the change in the weight of the process element \( [eq. (18)] \) was used to facilitate the smoothing of weight changes, which resulted in better convergence of the net.

### Training data

The sets of training exemplars generally consisted on \( n \) training vectors \( V_k \), each of which could be partitioned into a subvector \( V_k^\text{in} \) containing all the inputs to the net, and a subvector \( V_k^\text{out} \) which contained the desired outputs associated with the corresponding inputs, i.e.

\[ V_k = [V_k^\text{in} | V_k^\text{out}] \]

The inputs to the net were based on either the constraint residuals \( r_{pi} \), the measurement residuals \( W_i - W_i^p \) or \( x_{ij} - x_{ij}^p \), or the measurements of the variables \( W_i \) or \( x_{ij} \). Each of these approaches is considered in turn.

### Detection of the presence of gross errors in variables based on the constraint residuals of the system

In the first approach, the inputs to the net consist of the magnitudes of the normalized constraint residuals generated by the errors in the observed values of the variables, i.e.

\[ e_{pi} = |r_{pi}|/\sigma_{pi} \; p = 1, 2, \ldots, n + 1 \]  
\[ e_{pi}^\text{out} = 0 \; \text{if } e_{pi}^\text{in} \text{ is associated with a random error} \]  
\[ e_{pi}^\text{out} = 1 \; \text{if } e_{pi}^\text{in} \text{ is associated with a biased error} \]

A training (100 vectors) and a test set (50 vectors) were constructed from constraint residuals generated by corrupting a single set of arbitrary consistent measurements \( \{W_1, W_2, W_3, x_{i1}, x_{i2}, x_{i3}, x_{i4}, x_{i5}, x_{i6}\} \) with the previously mentioned random and gross errors.

After convergence, the performance of the net was evaluated against a test set consisting of 50 test vectors, with a composition similar to that of the exemplars in the training set. These vectors consisted of a set of input values, as well as a set of actual output values which could be compared with the output values generated by the net. The weights of the trained net are shown in Table 1.

On average, the net managed to classify correctly 94% of the large errors with a 50% bias in the data. Although virtually no overlap occurred between the gross error and random error distributions, it should be borne in mind that based on an analysis of the constraint residuals of the system, gross errors in relatively small variables might still be indistinguishable owing to random noise in relatively large variables. As could be expected, the net had somewhat less success (91%) in discriminating between the smaller errors with a 20% bias which were less discernible from the random errors in the data. (It is not difficult to generalize this trend, with results similar to those of other error detection tests, i.e. the smaller the bias of the gross error, the smaller the chance of the error being detected.)

In order to evaluate the robustness of the net, the true values of the variables were changed completely and corrupted at random with the same errors used previously (standard deviations of 5% and biases of 20 and 50%). The previously trained net was subsequently used to detect the presence of gross errors in the new system without retraining. The ability of the net to detect gross errors was found to be essentially the same as in the previous case (within 5%). Stated differently, as long as the magnitudes of the constraint residuals are larger than a certain threshold value, the net will interpret this as an indication of the presence

### Table 1. Weights of net used to detect gross errors in two-product process system based on constraint residuals

<table>
<thead>
<tr>
<th>Output unit 1</th>
<th>Output unit 2</th>
<th>Output unit 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bias</td>
<td>7.2773</td>
<td>-0.9801</td>
</tr>
<tr>
<td>Input unit 1</td>
<td>9.1197</td>
<td>-2.6379</td>
</tr>
<tr>
<td>Input unit 2</td>
<td>-0.2414</td>
<td>2.7840</td>
</tr>
<tr>
<td>Input unit 3</td>
<td>0.0637</td>
<td>1.3763</td>
</tr>
</tbody>
</table>
of a gross error, regardless of the actual or observed values of the variables used to generate the constraint residuals.

Note that, by using a method such as this, gross errors cannot be isolated, though these errors can be confined to subsets of three variables at a time. That is if a gross error is present in any of the flow streams \( W_i \), all three constraint residuals \( r_1, r_2 \) and \( r_3 \) will test positive for the presence of a gross error. If the gross error occurs in any of the variables \( x_{11}, x_{21} \) or \( x_{31} \), only \( r_1 \) will test positive for a gross error, while only \( r_3 \) will test positive if \( x_{21}, x_{22} \) or \( x_{32} \) contains a gross error. In order to locate gross errors more precisely, procedures using the measurements or measurement residuals themselves are necessary.

Detection of the presence of gross errors in variables based on the measurement residuals of the system. In this approach the same procedure is followed as in the previous approach, except that the inputs to the net consist of the normalized magnitudes of the measurement residuals of the variables, instead of the constraint residuals:

\[
v_i^N = |W_i^o - W_i|/\sigma_{wi} \quad (i = 1, 2, 3)
\]

\[
v_{ij+j+1}^Ni = |x_{ij} - x_{ij}'|/\sigma_{xij} \quad (i = 1, 2, 3; j = 1, 2)
\]

\[
v_k^O = 0 \text{ if } v_k^N \text{ is associated with a random error} \quad (k = 1, 2, \ldots, 9)
\]

\[
v_k^O = 1 \text{ if } v_k^N \text{ is associated with a biased error} \quad (k = 1, 2, \ldots, 9).
\]

Note that these residuals are regarded as the differences between the true and observed values of the variables. In practice, the true values of the variables are not known and have to be estimated by an appropriate reconciliation procedure.

A net with the same general structure as the net used with the constraint residuals of the system was defined, i.e. a single input and output layer consisting of hyperbolic tangent process units. The net was trained with a set of 200 exemplars and after convergence, the net was evaluated against a set of test data as before. The weights of the trained net are shown in Table 2. The net was able to classify all the gross errors with a 50% bias (equivalent to 10 standard deviations) correctly, but could only classify approximately 84% of the errors with a 20% bias (equivalent to four standard deviations) correctly.

The ability of the net to detect gross errors associated with variable values, not similar to those the net had been trained on, remained essentially the same, as shown in Fig. 5. The net managed to detect all gross errors with a 50% bias regardless of the specific values of the variables with which these errors were associated, while the net’s ability to detect gross errors with a magnitude of 20% in variable sets it had not been trained on, deteriorated only marginally, typically as shown in Fig. 5.

Some experiments indicated that the actual (positive or negative) values of the residuals can also be
Gross errors in material balance measurements by means of neural nets

Errors correctly classified [%]

![Classification Chart]

Fig. 5. Detection of gross errors based on measurement residuals two-product classifier

used in more sophisticated error classification schemes, but more complicated neural net structures (typically characterized by a single hidden layer) would then be required.

Detection of the presence of gross errors in variables based on the measurements of the system variables. This third approach is essentially the same as the first two described above, except that instead of using the measurement residuals, the actual measurements themselves are used, i.e. the inputs to the net consist of the measurements of the variables \( w_i \) and \( x_{ij} \):

\[
\begin{align*}
\nu_i^\text{IN} &= W_i; & (i = 1, 2, 3) \\
\nu_{2i+j+1}^\text{IN} &= x_{ij}; & (i = 1, 2, 3; \ j = 1, 2) \\
\nu_k^\text{OUT} &= 0 & \text{if } \nu_k^\text{IN} \text{ is associated with a random error } (k = 1, 2, \ldots, 9) \\
\nu_k^\text{OUT} &= 1 & \text{if } \nu_k^\text{IN} \text{ is associated with a biased error } (k = 1, 2, \ldots, 9).
\end{align*}
\]

The advantage of this approach is that no estimates of the true values of the variables are required, since the net constructs its own representation of these true values. It can be shown that the results obtained with this approach are much the same as with the previous method. However, when different true values are used, the net has to be retrained to classify the errors in the variables. Despite the advantage of not having to estimate the true values of the variables, this approach is consequently limited to very stable steady-state systems under strictly controlled operating conditions and is thus seen to be of limited value in practice.

Example 2: a metallurgical grinding circuit

In the previous example, the ability of a neural net to identify gross errors in data subject to non-linear constraints has been demonstrated. In this example, the performance of a neural net is compared to that of a statistical method described in the literature. The specific method has been selected because it is one of the few reported to deal explicitly with non-linear constraints, and is furthermore reported to be an efficient means of detecting multiple gross errors in measured data.

The example chosen from the literature concerns the use of a modified iterative measurement procedure to detect gross errors in the measurement data and the results are consequently compared with those obtained by a neural net (Serth et al., 1987, 1989).

The circuit consisted of a ball and rod mill connected to a cyclone classifier as shown in Fig. 6. Based on the constraints that the mass fractions have to sum to unity, the mass fractions of water in streams 5–9 have been eliminated, and the following set of equations was obtained to describe mass flow in the circuit (Serth et al., 1987):

Rod mill (node 1):

\[
R_1 - R_3 + R_5 \sum_j W_{5j} = 0 \quad (20)
\]

\[
R_4 - R_5 \sum_j W_{5j} = 0. \quad (21)
\]

Pump (node 2):

\[
R_2 + R_3 + R_6 - R_7 - R_3 \sum_j W_{7j} - R_6 \sum_j W_{6j} = 0 \quad (22)
\]

\[
R_4 W_{7j} + R_6 W_{6j} - R_7 W_{7j} = 0 \quad (j = 1, 2, 3). \quad (23)
\]

Ball mill (node 3):

\[
R_3 + R_8 - R_6 + R_6 \sum_j W_{6j} - R_8 \sum_j W_{8j} = 0 \quad (24)
\]

\[
R_8 W_{6j} - R_6 \sum_j W_{6j} = 0. \quad (25)
\]

Cyclone (node 4):

\[
R_7 - R_8 - R_7 \sum_j W_{7j} + R_6 \sum_j W_{6j} + R_9 \sum_j W_{9j} = 0 \quad (26)
\]

\[
R_7 W_{7j} - R_8 W_{6j} = 0 \quad (j = 1, 2, 3). \quad (27)
\]

The corresponding neural net used to identify gross errors in the measurement variables consequently consisted of 24 input nodes (24 measurement residuals) or process units, connected to an output layer. The output layer similarly consisted of 24 process units, corresponding to the 24 measurement variables. The net was subsequently trained on various data sets.
containing one to several gross errors in each training vector. As in example 1, the training exemplars consisted of vectors $V_k$, viz.

$$V_k = \{V_{k1}, V_{k2}\}$$

where

$$V_{k1} = [R_i - R_i'] \quad (i = 1, 2, \ldots, 24)$$

$$V_{k2} = [w_{ij} - w_{ij}'] \quad (i = 5, 6, \ldots, 9, j = 1, 2, 3)$$

and

$$V_{k1}^{\text{OUT}} = 0 \quad \text{if } v_i \text{ is associated with a random error}$$

$$V_{k1}^{\text{OUT}} = 1 \quad \text{if } v_i \text{ is associated with a biased error}$$

The data were corrupted by the same errors used by Serth et al., as shown in Table 3 and depicted in Fig. 7. As can be seen from Fig. 7 where a random error is compared with two of the gross errors (with biases of 10 and 20%), only gross errors with a bias smaller than approximately 20% overlap with random errors, and it can be shown that except for these few errors, all other errors can be detected by a suitable test method such as the so-called statistical measurement test (Mah and Tamhane, 1982). As a result a very high success rate can be expected when gross errors with biases larger than 20% occurs in the data. Since these residuals are generally not available for testing (the true values of the variables are unknown), the data are reconciled and the adjustments are then subjected to testing for gross errors, on the premise that adjustments in measurement values subject to (large) gross errors will be relatively large (i.e. the ratio of the expected value of the adjusted measurement to that of the true value of the measurement will exceed unity). It should thus be noted that Serth et al. (1987) did not use the residuals $R_i' - R_i$ and $W_{ij}' - W_{ij}$, but used to train the neural net, the adjustments $R_i' - R_i$ and $W_{ij}' - W_{ij}$.

After training the net, it was presented with a previously unseen set of test data, corrupted by the same types of errors the net had been trained to recognize. Since only the distribution of errors with a bias of less than 10% overlapped with those of the random errors in the data, the net was able to categorize virtually all errors in the measurement data correctly. The results of the classification process are depicted in Tables 4–6. The weights of the trained neural net are shown

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard deviation (%)</td>
<td>2.5</td>
</tr>
<tr>
<td>Lower bound on systematic error magnitude</td>
<td>0.1</td>
</tr>
<tr>
<td>Upper bound on systematic error magnitude</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Fig. 6. Metallurgical grinding circuit (Serth and Heenan, 1987).

Fig. 7. Normal distribution of random and gross errors. Standard deviation of all errors: 2.5% (Serth et al., 1987).

Table 3. Parameter specifications for test cases

![Diagram](https://scholar.sun.ac.za)
Gross errors in material balance measurements by means of neural nets

schematically in Fig. 8, where the sizes of the squares are indicative of the values of the weights (solid and empty squares represent positive and negative values, respectively). In this figure, the first input unit is designated as the bias unit, while the 24 input are numbered consecutively from 2 to 25 (shown on the horizontal axis) and the 24 output units are number consecutively from 26 to 49 (shown on the vertical axis).

The neural net clearly performed considerably better than the MIMT procedure. From the work done by Serth et al. (1987), it also appears as if the performance of the MIMT algorithm was influenced by the magnitudes of the systematic errors in the data sets. The MIMT algorithm detected only 8% of the gross errors in the smallest flow variable (R₁) and zero in the smallest composition variable (mass fraction 1 in stream 9). This can probably be attributed to the relatively poor estimates of the true values of the process variables during reconciliation of the data, and should not be interpreted as the result of an inadequate gross error test as such. The neural net

Table 4. Comparison of neural net with MIMT method to detect gross errors in measurement data (one gross error)

<table>
<thead>
<tr>
<th>Test method</th>
<th>MIMT (%)</th>
<th>BPNN (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Systematic errors detected</td>
<td>82</td>
<td>98</td>
</tr>
<tr>
<td>Average number of erroneous identifications</td>
<td>0.29</td>
<td>0</td>
</tr>
</tbody>
</table>

1 Modified iterative measurement test (Serth et al., 1987, 1989).
2 Back propagation neural net.

Table 5. Comparison of neural net with MIMT method to detect gross errors in measurement data (five gross errors)

<table>
<thead>
<tr>
<th>Test method</th>
<th>MIMT (%)</th>
<th>BPNN (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Systematic errors detected</td>
<td>75</td>
<td>97</td>
</tr>
<tr>
<td>Average number of erroneous identifications</td>
<td>1.40</td>
<td>0</td>
</tr>
</tbody>
</table>

1 Modified iterative measurement test.
2 Back propagation neural net.

Table 6. Comparison of neural net with MIMT method to detect gross errors in selected variables

<table>
<thead>
<tr>
<th>Variable number</th>
<th>Description</th>
<th>Number of gross errors</th>
<th>Percentage of gross errors detected</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Second smallest flow</td>
<td>10</td>
<td>22</td>
</tr>
<tr>
<td>3</td>
<td>Smallest flow</td>
<td>12</td>
<td>25</td>
</tr>
<tr>
<td>7</td>
<td>Largest flow</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>13</td>
<td>Second smallest composition</td>
<td>11</td>
<td>16</td>
</tr>
<tr>
<td>20</td>
<td>Largest composition</td>
<td>14</td>
<td>20</td>
</tr>
<tr>
<td>22</td>
<td>Smallest composition</td>
<td>15</td>
<td>23</td>
</tr>
</tbody>
</table>

1 Modified iterative measurement test.
2 Back propagation neural net.

Fig. 8. Graphic representation of the weights of the neural net used to detect gross errors in example 2. [Units in the input layer include the BIAS unit and units labelled from 2 to 25, while units in the output layer are labelled from 26 to 49. Large and small squares represent large and small weights respectively, while solid squares (+) and empty squares (−) indicate the signs of the weights.]

would probably have yielded similar results if it had to contend with the adjustments generated by the MIMT procedure.

DISCUSSION OF RESULTS

The use of neural nets for the detection of gross errors in process data is best explained in analogy with statistical tests presently used for the same purpose. These tests are based either on the constraint residuals generated by the inconsistent data, or directly on the measurement residuals generated when the observed values of the variables are reconciled with the constraints.

Neural net techniques follow much the same approach and the prerequisites to the application of these methods are the same as for statistical tests—analytical redundancy of the process data and knowledge of the distribution of random errors in the
data. An important point is the fact that the knowledge concerning the random errors does not need to be explicit when neural nets are used (i.e. the usual assumption of a normal distribution with a certain variance and a zero mean), since the net can derive its own representation of this distribution when provided with sufficient suitable process data. Once the net has constructed a model of the random errors in the measurements (from examples presented to it in the training phase), it uses this representation as the exclusive standard against which errors are classified.

When statistical methods based on constraint residuals are used, the distribution of the constraint residuals has to be known and since this distribution is a function of the distributions of the individual variable measurements, statistical methods are generally limited to linear systems. (In non-linear systems the distribution of the constraint residuals is no longer of the same type as that of the measurement residuals and generally unknown.) In neural nets this limitation does not apply, since the net learns the distribution of the constraint residuals for the particular system prior to classification of these residuals. Having learned this relationship for a particular set of constraint equations and a random error distribution, the net can be used to detect the presence of gross errors in the set of variable measurements. Once detected, other techniques can then be used to isolate these systematic errors.

Alternatively (and in order to isolate gross errors directly), tests can be based on measurement residuals. In contrast with the constraint residuals, these measurement residuals can unfortunately not be determined directly from the variable measurements themselves, since the true values of the variables at the time of measurement are not known (assuming the process model to be correct, the true values of the constraint residuals at the time of measurement are known to be zero). As a result it is not possible to use the actual measurement residuals (the differences between the true and the observed values of the variables) to locate gross errors in the system. The estimated measurement residuals, however, can be used (the differences between the estimated or reconciled and the observed values of the variables). As a consequence, the detection of gross errors based on measurement residuals is dependent on the accuracy with which the true values of the variables can be estimated, i.e. the reconciliation procedure used to filter the data.

When the measurement residuals are evaluated by means of neural nets, two similar approaches are possible. In the first approach the net can be trained with the measurement residuals per se to identify gross errors in the reconciled data. The relation between the measurement residuals is typically relatively simple and training of the net uncomplicated, so that accurate classification of the errors is possible, provided that the residuals presented to the net during training and testing are reasonably close estimates of the actual residuals of the measurements.

In an alternative approach, the reconciliation procedure can be dispensed with altogether, and the net can be trained with the observed values of the variables only. From these values, the net constructs its own model of the measurement residuals and uses these to identify measurements that are biased. This procedure is less robust than the previous two, in that the net is sensitive to the actual measurements used during training, i.e. if the net is trained with observed values centered around a particular true value of the process variable, a significant change in the true value of the variable will result in erroneous classification of errors in the measurements. Owing to this drawback the previous approach based on the measurement residuals is recommended.

The nets used for the detection of gross errors in the examples described in this paper have simple single-layer structures. This can, among others, be attributed to the use of the absolute magnitudes of the residuals, which simplifies the decision space that the net has to model.

In the light of these comments the following can be concluded that:

- neural nets constitute a powerful means of detecting gross errors in sets of constrained variables, regardless of the nature of the constraints
- explicit knowledge of the distribution of random errors in the variables is not a prerequisite to the use of neural net methods to detect gross errors
- neural nets can be used to isolate gross errors in variable measurements, regardless of the constraints of the system
- existing schemes can be adapted readily to incorporate the use of neural net methods for the detection of gross errors in process systems
- due to their parallel architecture, the use of neural nets in the so-called neural circuits (electronic circuits designed to derive maximal benefit from the parallelism of neural nets) for the on-line detection of gross errors in process systems appears to be an attractive option.

### NOTATION

- $A$: matrix of process constraint coefficients
- $b$: magnitude of leak vector
- $b_j$: bias of gross error
- $\text{cov}(\cdot)$: covariance
- $d_{o,j}$: desired output value of process unit $j$
- $DW$: mass balance constraint, $W_1 - W_1 - W_2 = 0$
- $DX_1$: mass balance constraint, $W_1 x_{11} - W_2 x_{21} - W_3 x_{31} = 0$
- $e$: errors in measurements of process variables
- $E(\cdot)$: expected value
- $e_v$: gross error in measured variable
- $e_i$: error generated in neural net during training
- $E_i$: process element $i$ in output layer of neural net trained to classify errors in measurement data
- $e_r$: random error in measured variable
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$g$</td>
<td>transfer function of neural net process element</td>
</tr>
<tr>
<td>$k_1$</td>
<td>constant</td>
</tr>
<tr>
<td>$k_2$</td>
<td>constant</td>
</tr>
<tr>
<td>$m$</td>
<td>leak vector</td>
</tr>
<tr>
<td>$m_{ij}$</td>
<td>momentum term in back propagation learning rules</td>
</tr>
<tr>
<td>$Q$</td>
<td>covariance matrix of process variable measurements, $y'$</td>
</tr>
<tr>
<td>$r$</td>
<td>residuals of process constraints</td>
</tr>
<tr>
<td>$R_j$</td>
<td>actual value of flow stream $j$</td>
</tr>
<tr>
<td>$R'_j$</td>
<td>adjusted value of flow stream $j$</td>
</tr>
<tr>
<td>$R_i$</td>
<td>observed value of flow stream $i$</td>
</tr>
<tr>
<td>$r_p$</td>
<td>$p$th process constraint in measured data</td>
</tr>
<tr>
<td>$t$</td>
<td>time</td>
</tr>
<tr>
<td>$u_i$</td>
<td>potential of process unit $i$ in neural net</td>
</tr>
<tr>
<td>$V$</td>
<td>covariance matrix of constraint residuals</td>
</tr>
<tr>
<td>$V^N_i$</td>
<td>component $j$ of vector $V^N$</td>
</tr>
<tr>
<td>$V_{out}^k$</td>
<td>component $j$ of vector $V_{out}$</td>
</tr>
<tr>
<td>$V_{in}^k$</td>
<td>$k$th training vector</td>
</tr>
<tr>
<td>$V_{in}$</td>
<td>component or subvector of training vector containing input values</td>
</tr>
<tr>
<td>$V_{out}$</td>
<td>component or subvector of training vector containing desired output values</td>
</tr>
<tr>
<td>$W_i$</td>
<td>actual flow rate of stream $i$</td>
</tr>
<tr>
<td>$W'_i$</td>
<td>estimated or adjusted flow rate of stream $i$</td>
</tr>
<tr>
<td>$w_{ij}$</td>
<td>connection strength or weight between process units $i$ and $j$ in neural net</td>
</tr>
<tr>
<td>$x_{ij}$</td>
<td>actual mass fraction of component $j$ in flow stream $i$</td>
</tr>
<tr>
<td>$x'_{ij}$</td>
<td>estimated or adjusted mass fraction of component $j$ in flow stream $i$</td>
</tr>
<tr>
<td>$x''_{ij}$</td>
<td>observed fraction of component $j$ in flow stream $i$</td>
</tr>
<tr>
<td>$y$</td>
<td>true values of process variables</td>
</tr>
<tr>
<td>$y'$</td>
<td>adjusted values of process variables</td>
</tr>
<tr>
<td>$y^*$</td>
<td>measured values of process variables</td>
</tr>
<tr>
<td>$y_e$</td>
<td>a measured variable containing a gross error</td>
</tr>
<tr>
<td>$y_r$</td>
<td>a measured variable containing a random error, $e$</td>
</tr>
<tr>
<td>$z_i$</td>
<td>state of process unit $i$ in neural net</td>
</tr>
<tr>
<td>$z_{out,i}$</td>
<td>actual output of process unit $j$</td>
</tr>
</tbody>
</table>

**Greek letters**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varepsilon$</td>
<td>error criterion for adjustment of weight matrix of neural net</td>
</tr>
<tr>
<td>$\Theta_i$</td>
<td>bias of process unit $i$ in neural net</td>
</tr>
<tr>
<td>$\tau$</td>
<td>learning rate of process unit $i$</td>
</tr>
</tbody>
</table>

**REFERENCES**


The use of connectionist systems to reconcile inconsistent process data

C. Aldrich and J.S.J. van Deventer
Department of Metallurgical Engineering, University of Stellenbosch, Private Bag X5018, Stellenbosch 7599 (South Africa)

(Received August 1, 1993; in final form January 4, 1994)

Abstract

Since measurements of variables in chemical and metallurgical plants generally violate the conservation and other constraints of these systems owing to random measurement errors, these data have to be reconciled with the constraints prior to further use. In multicomponent systems the reconciliation of process data normally results in a non-linear constrained optimization problem, which constitutes a formidable computational burden when large systems have to be solved by conventional techniques. Connectionist systems, such as artificial neural networks, can be implemented to considerable advantage for the solution of optimization problems such as these and in this paper their use is explored. Three variants of crossbar feedback connectionist systems were investigated, two are based on gradient descent techniques and one on a direct search method. The results of simulations, as well as a comparison with traditional computational procedures, indicate that systems such as these based on gradient descent techniques can be used to solve large systems efficiently.

1. Introduction

A clear understanding of the operational behaviour of a plant is essential for the identification of process phenomena, as well as for the optimization and control of the plant or process circuit. The collection and analysis of data from processes are therefore important means for evaluating the performance of a plant or an individual process unit. The available data are generally subject to random noise, or even gross errors, which can among others be attributed to failure or miscalibration of measuring instruments, the departure of the process from steady state owing to malfunctioning process equipment, or significant changes in the environment. It is thus vital that the inconsistent data are reconciled with the process constraints prior to further use.

The usual approach to the reconciliation of measured variables is aimed at minimization of the weighted sum of the squares of the measurement residuals, subject to conservation and other constraints of the process [1] i.e.

$$\min \ (x-x')^T V^{-1}(x-x')$$

subject to $f(x')=0$

where $V^{-1}$ is a residual weighting matrix, usually the inverse of the variance-covariance matrix of the measurements and $x'$ the vector of reconciled measurements (see Appendix A). Where an estimate of this variance-covariance matrix is not available, a numerical weighting system can also be used by defining $V^{-1}$ as the inverse of the elements of the measurements of the variables $\{x\}$.

The solution of the problem in effect ensures that the flow parameters are adjusted as little as possible, that all the conservation constraints are satisfied, and that more reliable variable measurements are adjusted less than less reliable variable measurements.

Most techniques [2-5] involve interactive procedures and are computationally demanding, especially as far as large complex plants are concerned, or where on-line material balancing is required. Although more powerful computer processors are continuously being developed, the last few years have seen a strong shift towards parallel computer systems which are capable of orders of magnitude increases in computational speed when compared with current serial machines. These computers are broadly characterized by the degree of parallelism exhibited as well as the power of the processors in these systems. At one end of the spectrum only a few very powerful processors are arranged in parallel, while at the other end of the spectrum large numbers of simple processing units are connected in massively parallel structures (such as are...
found in neural networks or connectionist systems). Both designs have their strengths and weaknesses and the use of a particular system is best determined by the nature of the application. Despite the obvious advantages of parallel computers, they are presently only used on a limited scale in process engineering, mainly owing to the lack of suitable computational algorithms that can exploit their parallel architectures.

Owing to their massively parallel structures, and recent advances in very-large-scale integration (VLSI) and ultra-large-scale integration (ULSI) electronic circuits [6–8], neural nets show great potential for the solution of computational problems of high dimension in processing times several orders of magnitude less than required with sequential computational devices [9, 10]. Since artificial neural networks do not require special algorithms, they can be used as special purpose supercomputers without resorting to complicated programming.

In this paper the use of connectionist systems for the rectification of inconsistent redundant variable measurements is proposed and demonstrated by way of two elementary examples.

2. Connectionist systems

Recurrent or feedback nets, especially those known as crossbar or Hopfield nets, have been used for a wide range of optimization problems ranging from the solution of non-polynomial (NP) complete combinatorial problems such as the travelling salesperson problem [11], combinatorial optimization problems subject to inequality constraints [12], assignment problems [13], systems of complex-valued linear equations [14], the four colour mapping problem [15], the identification and recognition of visual images [16, 17], as well as the solution of linear [11, 18, 19], non-linear [20, 21] and dynamic programming [22] problems. These nets differ from feedforward systems (such as back propagation nets) in that information is not only passed forward through the layers of the net, but backwards or laterally as well. The performance of three different connectionist systems explained below and referred to as CS-I, CS-II and CS-III was investigated.

2.1. Connectionist system I (CS-I)

The architecture of CS-I corresponds to that of a crossbar or Hopfield neural net, as shown in Fig. 1. The system consists of three layers, i.e. an input layer, a hidden layer with full lateral connections, and an output layer. All layers have the same number \( N \) of elements, and all are provided with linear

![Diagram of connectionist system CS-I](https://scholar.sun.ac.za)

Transfer functions, of the form

\[
v(u) = mu + k, \quad (m, k \text{ constant})
\]

The exact number of elements in each layer is determined by the number of process variables to be reconciled (i.e. one element for each process variable).

2.1.1. Neurodynamics

When these networks are viewed as dynamic systems, the network computation process can be seen as a system moving in a state space* through the constant application of some transition rules. These transition rules are procedures for updating the state of the system, depending on its current state. The system dynamics or neurodynamics of the net are determined by the transition rule, as well as the order in which the system variables or node outputs are updated. If application of the transition rule ceases to affect the current state of the system, the system is said to have converged to a fixed point or attractor in the state space. The set of all initial states or points leading to this fixed point is known as the attractor basin of the particular attractor [23].

In order to analyse the dynamics of the system it is usually convenient to define a scalar function which depends on the state of the system and which has a definite value for each point in the state space. If the value of this energy or cost function \( E \) does not increase with a change in the state \( v \) of the system (i.e. \( \frac{dE}{dv} < 0 \), for all possible \( v \)) and is bounded from below, it is also a Lyapunov function, and an indication that the system is unconditionally stable.

*The state space of a set of variables \( z = [x_1, x_2, x_3, \ldots, x_n] \) is the Cartesian product of the domains of the variables, i.e. \( D = \{a_1, a_2, a_3, \ldots, a_n\} \). The state space \( A \) of the set of \( n \) variables \( z \) is thus \( n \)-dimensional.
By mapping the objective function and the constraints of an optimization problem onto this energy function, these problems can be solved in that the optimal solution to the problem is forced to coincide with the minimum energy of the system. The dynamics of the net amount to a constraint relaxation process, when the energy measure is defined by the degree of constraint violation of the system.

To use crossbar or Hopfield nets for material balance reconciliation problems, it is necessary to map the objective function and the constraints of the system onto this energy measure, these problems can be solved in a gradient descent style, with constant step size lengths.

The neurodynamics of the net are thus defined by means of the Newton equations, i.e.

$$\frac{du_i}{dt} = -\frac{dE}{dv_i}$$

(4)

the computational energy function $E$ is forced to decrease monotonically, regardless of the nature of this function.

Proof [15]:

$$\frac{dE}{dt} = \sum_i(\frac{du_i}{dt})(\frac{dE}{dv_i})$$

$$= \sum_i(\frac{du_i}{dt})(-\frac{du_i}{dt})$$

$$= -\sum_i(\frac{du_i}{dt})(\frac{du_i}{dv_i})(\frac{dv_i}{dt})$$

$$= -\sum_i(u_i^2) \leq 0$$

(5)

As long as the transfer function $v_i(u_i)$ is continuous and non-decreasing, $du_i/dt$ is always positive and $dE/dt$ always negative or zero. The resultant state of the system can consequently be related to a solution of the problem.

The neurodynamics of the net are thus defined by a set of ordinary differential or difference equations, which have to be integrated at each time increment to determine the output states of the neurons after each change of state:

$$du_i/dt = f_i(v)$$

(6)

$$u_i^{t+1} = u_i^t + (f_i(v)) \Delta t$$

(7)

Integration of these equations continues until the system has reached a point of stability (i.e., its energy has been reduced to a minimum, so that $du_i/dt = 0$, for all $i$). In practical terms the system is considered to be stable when

$$\sum_i|du_i/dt| \leq \varepsilon$$

(8)

where $\varepsilon > 0$ is an arbitrary small convergence criterion. The solution of this system of non-linear equations is based on the use of a gradient descent technique, with constant step size lengths.

2.1.2. Scaling of data

Before the data are presented to the net, it is important that they are scaled to ranges that are useful with regard to the neurodynamic function being used. Without proper scaling, process elements could become saturated, which could eventually have a severe effect on the movement of the system through state space. It is moreover desirable to scale all the variables to similar ranges, so that the influence of variables with relatively large numeric values does not affect the decrease in energy of the system at the expense of variables with relatively small numeric values. Scaling is usually accomplished by mapping the minimum and maximum values of the actual input and output data linearly to the respective minimum and maximum values of the network ranges. If a measurement vector presented to the net consists of $l$ fields, $[x_1, x_2, \ldots, x_l]$, two sets of corresponding vectors can be defined $[m_1, m_2, \ldots, m_l]$ and $[M_1, M_2, \ldots, M_l]$, where $m_i$ and $M_i$ typically correspond to the minimum and maximum values that $x_i$ could assume. (These indices can assume any values, as long as $m_i < M_i$.) If the ranges allowed for the input and output layers of the net are respectively defined as $(R_i_{\text{min}}, R_i_{\text{max}})$ and $(R_o_{\text{min}}, R_o_{\text{max}})$, $i_j$ as the scaled network input corresponding to the actual input $x_j$, $o_k$ as the actual output of the net and $x'_k$ as the corresponding descaled output, then the mappings of the real world data to those of the network can be described as follows:

input

$$i_j = \frac{(R_i_{\text{max}} - R_i_{\text{min}})x_j + M_iR_i_{\text{min}} - M_jR_i_{\text{min}}}{M_j - M_i}$$

output

$$x'_k = \frac{(M_k - m_k)o_k + R_o_{\text{max}}m_k - R_o_{\text{min}}M_k}{[R_o_{\text{max}} - R_o_{\text{min}}]}$$

Missing field values could be mapped to the middle of the target range, that is $\frac{1}{2}(R_i_{\text{max}} + R_i_{\text{min}})$ or $\frac{1}{2}(R_o_{\text{max}} + R_o_{\text{min}})$ when necessary.
2.1.3. Connection weights

In order to represent appropriately the function to be minimized, the weights of the net are defined by the variance-covariance matrix of the measurements, as well as the weights associated with the process constraint residuals, as presented in eqn. (3). Since estimates of the variance-covariance matrix elements are often not available, a weight matrix based on the actual values of the variable measurements will be defined. This ensures that the values of small variables are not adjusted by increments that are unduly large in relation to the value of the variables themselves.

The scaled input (measured values of the process variables) of the net is clamped to the input layer, and the states of the elements in the hidden layer are updated repeatedly and asynchronously (simulated by a random updating procedure) through numeric integration of the potential of each element. The system is allowed to settle into a minimum point, and the output of the hidden layers (the solution) is passed forward to the process elements in the output layer, from where it is descaled to yield a solution to the optimization problem.

2.2. Connectionist system II (array of crossbar feedback nets)

The optimal point (energy minimum) attained by the CS-I system depends on the initial starting conditions and it is therefore quite possible that the system might be slowed down by shallow attractor basins in the search space or trapped in local minima when a complex response surface is associated with the particular reconciliation problem. These problems can be circumvented through further expansion of the CS-I type of system, by constructing a system essentially consisting of series of CS-I systems in parallel.

The CS-II system is thus essentially a generalized version of the first system, in that instead of having a single hidden layer, it has a P-dimensional array of hidden layers (if \( P = 1 \), the system reduces to CS-I), each containing \( N \) elements in general, as shown in Fig. 2. The input section of the system consists of a single input layer, each element of which is connected to a corresponding element in each of the \( P \) hidden layers. The elements in the hidden layers are similarly connected to corresponding elements in a single output layer. The input and output layers do not process the data, but merely serve as distribution points for data input and output.

The same neurodynamic principles concerning CS-I are applicable, except that once the measurement vector has been fed to each of the different layers in the hidden array, \( P \) different sets of initial conditions are generated in the array prior to the commencement of relaxation of the energy of the net. Cycles of state changes are allowed to take place independently in each layer in the array, and when the local rate of change of the energy of the system in a particular layer \( L \) decreases beyond a certain limit, the layer is allowed to poll the other layers in the unit. If the energies of one or more of these layers are lower than that of the polling layer, the latter assumes a new state in the neighbourhood of the state of the layer \( K \) with the lowest energy and continues its convergence to a lower energy state if possible. That is if \((\frac{dE}{dt})_L < \varepsilon_1\), and

\[
\text{if } u_{i,K} < u_{i,L}, \text{ then } u_{i,K} = u_{i,K} + \phi (0 < \phi < 0.1)
\]

In this way a number of layers operating in parallel are also used in parallel to search for an optimum. It is especially when negotiating complex response surfaces that this second order of parallelism can be expected to yield superior results (as will be demonstrated by an example in the following).

2.3. Connectionist system III (three-layer net with feedback from output to input layers)

Direct search procedures are attractive for the solution of sets of non-linear equations, since they are easy to use and are generally regarded as computationally efficient [24]. A direct random search procedure with systematic search space contraction, such as proposed by Luus and Jaakola [24] and Luus and Wang [25] was incorporated in the neurodynamics of the third system, shown schematically in Fig. 3. CS-III is equivalent to CS-II, with the difference that instead of a gradient-based search, use is made of a direct method with a systematic reduction in the search space associated with each interval. The reduction in the search intervals associated with each of the search variables is aimed at a more efficient search procedure, since unless the search domain is in the immediate vicinity of the optimum, convergence by means of a random search can be very ineffective [26]. The procedure is implemented as follows:

1. set the time increment counter \( j = 1 \);
2. set up the system, so that the initial states \( v^0 \) of the artificial neurons in all \( P \) hidden layers correspond to the measured values \( x \) of the process variables;
3. define an initial search range \( r_j \) for each of the system states \( v_j \) of the neural net;
4. determine \( P \) sets of values, so that \( v_j^t = v_j^{t-1} + \phi_j^{t-1} r_j^{t-1} \), where \( \phi_j^{t-1} \) is a random
number associated with the state of artificial neuron \( i \) at time \( j-1 \), and \( 0 < \phi_{ij}^{-1} \leq 1 \), for all \( i \) and \( j \);

(5) of these \( P \) sets, determine the set which minimizes \( \sum |u_j| \);

(6) if \( \sum |du_j/dt| \leq \epsilon \), terminate the search, if not, reduce the search ranges \( r_j \) by an amount \( \tau \), i.e. \( r_j/(1-\tau) \), if \( \tau \neq 1 \), terminate the search, if \( \tau = 1 \), repeat the procedure.

After convergence a set of values \( v \) will remain, which should correspond to a minimum in the energy of the system, i.e. where \( du_i/dt = -dE/dv_i = 0 \), for all \( i \). This minimum will be the one closest to the initial state of the system, and in multimodal systems it might be necessary to incorporate stochastic procedures which would allow the system to find a global minimum point. The incorporation of procedures such as these was not pursued in this investigation.

3. Examples

3.1. Example 1: two-product classifier

In this example a two-product classifier (such as a hydrocyclone or screw classifier) is considered, which classifies a feed stream \( F_1 \) with \( n \) components into two output streams \( (F_2 \) and \( F_3) \). Measurements of the flow rates \( F_1 \) and component concentrations \( x_{ij} \) typically violate the mass conservation equations pertaining to the classifier, i.e.

\[
F_1 - F_2 - F_3 = 0
\]

\[
F_1 x_{i1} - F_2 x_{i2} - F_3 x_{i3} = 0, \text{ for } i = 1, 2, \ldots, n
\]
TABLE 1. Reconciled and measured values of the process variables in the two-product classifier (example 1); two-component system

<table>
<thead>
<tr>
<th>Measured values</th>
<th>F₁</th>
<th>F₂</th>
<th>F₃</th>
<th>x₁₁</th>
<th>x₁₂</th>
<th>x₂₁</th>
<th>x₂₂</th>
<th>x₃₁</th>
<th>x₃₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>F₁ 0.961</td>
<td>0.602</td>
<td>0.347</td>
<td>0.198</td>
<td>0.768</td>
<td>0.127</td>
<td>0.817</td>
<td>0.354</td>
<td>0.608</td>
<td></td>
</tr>
<tr>
<td>Typical reconciled values (CS-I and CS-II)</td>
<td>F₁</td>
<td>F₂</td>
<td>F₃</td>
<td>x₁₁</td>
<td>x₁₂</td>
<td>x₂₁</td>
<td>x₂₂</td>
<td>x₃₁</td>
<td>x₃₂</td>
</tr>
<tr>
<td>F₁ 0.956</td>
<td>0.608</td>
<td>0.348</td>
<td>0.205</td>
<td>0.752</td>
<td>0.123</td>
<td>0.828</td>
<td>0.351</td>
<td>0.613</td>
<td></td>
</tr>
</tbody>
</table>

Ratio of initial energy of system to final energy: \( E_0 / E_f = 11000 \).

Fig. 4. Performance of CS-I (example 1): effect of step size on the decrease in energy.

Fig. 5. Performance of CS-II (example 1): effect of number of layers on the decrease in energy; step size 0.2.

than 0.2, after which diminishing progress is made with further computation. (A constant step size was used for all the variables throughout the optimization procedure.) Step sizes larger than 0.2 resulted in unstable behaviour of the system. The iteration steps referred to in Fig. 4 comprise cycles through which each variable is updated once on average. For the two-product classifier, with 21 process variables \( F_i \), \( x_{ij} \) (\( i = 1, 2, 3 \), and \( j = 1, 6 \)), an iteration step thus consisted of a series of 21 random variable selections and subsequent adjustments of the selected variables. The reconciled values of the flow rates \( F_i \) and concentrations \( x_{ij} \) resulted in a threefold order of magnitude decrease in the objective function (energy function of the net), which is more or less comparable with results obtainable with other optimization techniques.

The performance of CS-II (number of layers 10 and 100) is compared with that of CS-I (number of layers 1) in Fig. 5. It is clear that the additional layers in the system do not lead to a significant improvement in performance. This is not surprising, since the process system considered is subject to bilinear constraints only, and does not not have a highly non-linear character which can be exploited to better advantage by the more powerful search procedure embodied by the CS-II system.

The connectionist system based on direct search techniques (CS-III) did not perform very well compared with those based on gradient descent techniques (CS-I and CS-II), as can be seen from Fig. 6. The system used in this case consisted of 200 layers and had an initial range of 0.1 for each search variable. This range was contracted to zero as the search progressed, but only resulted in a decrease of about 60%–70% in the initial energy of the system. Other initial search ranges and contraction procedures did not lead to significantly better results.

3.2. Example 2 [4]

This example is based on the one used by Pai and Fisher [4], as well as Tjoa and Biegler [5] and was chosen in order to make a rough comparison of the performance of the neural net with the computational procedures used by these authors. The example involves five measured variables \( x_1^m, x_2^m, x_3^m, x_4^m, x_5^m \) and three unmeasured variables \( x_1^u, x_2^u, x_3^u \), subject to six non-linear constraints:
The exact values of these variables are
\[ x^m = \{4.5124, 5.5819, 1.9260, 1.4560, 4.8545\} \] and
\[ x^u = \{11.070, 0.61647, 2.0504\} \] [5]. Tjoa and Biegler [5] corrupted 100 sets of these data with Gaussian noise in order to conduct a statistical evaluation of a tailored objective function in a non-linear computational routine, as well as a hybrid successive quadratic programming (SQP) routine.

In order to evaluate the use of a neural net to reconcile inconsistent constrained data, the exact values of the variables are similarly corrupted by Gaussian noise of 10% and 30%. The errors of a single set of variables resulting from the errors of corruption are shown as measurements in Tables 2 (10% noise) and 3 (30% noise).

One of the salient features of the system is the highly irregular response surface of the energy function of the net. The consequence of this highly non-linear character of the system is that the energy function is extremely sensitive to adjustment of the variables, especially at points where the derivative of the energy or objective function with regard to the adjustable variable \( \frac{\delta E}{\delta x_i} \) is very large (positive or negative). As a result very small time steps had to be used to ensure that the adjustment of a variable does not lead to overshooting of a local optimum in the response surface of the energy function.

In the case of network CS-I, the optimal step size for each variable is determined by a subroutine which systematically decreases the value of the initial time step if at first it does not result in a decrease in the system energy, until an improvement in the objective function is found. In this way relatively large time steps can be taken initially, which can be adjusted near troublesome spots on the surface of the energy function when necessary.

The results which compare favourably with those obtained by other non-linear methods [5] are shown in Tables 2 and 3, and typically led to a reduction of three orders of magnitude in the energy of the system after approximately 40 iteration steps with CS-I (multistep) and about four steps with CS-II with ten hidden layers. The percentage errors in the values of the variables before and after reconciliation (compared with the exact values of the variables) are also shown in Tables 2 and 3. The percentage error values have no particular meaning as such and merely provide an indication of the redistribution of the measurement errors among the variables. In accordance with general practice, the parameters of the objective function were specified so as to ensure very small process constraint residuals (at the expense of the adjustments made in the variables themselves). Figure 7 depicts graphically some of these results obtained with CS-I. Note that step sizes larger than approximately \( 10^{-5} \) lead to an unstable search procedure (compare with the values of 0.2–0.3 in the previous example). The use of different step sizes for the different search variables (MULTISTEP) instead of a constant step size for all variables, resulted in considerable improvements in the performance of the system.

In contrast to the situation highlighted by example 1, much is to be gained by using a multilayer system such as CS-II. In Fig. 8, the significant improvement in convergence based on the use of ten layers, versus one (CS-I), is illustrated. This difference can be attributed to the non-linear character of the response surface of the energy function. By making use of CS-II with many hidden layers operating in parallel, the trajectory of the system through state space can proceed to a global minimum without becoming trapped in local minima along the way.

Connectionist system CS-III displays the same less favourable convergence behaviour as was the
TABLE 2. Reconciled and corrupted values (10% Gaussian noise) of the process variables used in example 2

<table>
<thead>
<tr>
<th>Corrupted values</th>
<th>$x_1^m$</th>
<th>$x_2^m$</th>
<th>$x_3^m$</th>
<th>$x_4^m$</th>
<th>$x_5^m$</th>
<th>$x_6^m$</th>
<th>$x_7^m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e_{i1}^m$</td>
<td>4.786</td>
<td>5.564</td>
<td>1.917</td>
<td>1.365</td>
<td>5.307</td>
<td>10.225</td>
<td>0.617</td>
</tr>
<tr>
<td>$e_{i2}^m$</td>
<td>6.06</td>
<td>-0.32</td>
<td>-0.47</td>
<td>-6.25</td>
<td>9.32</td>
<td>-7.63</td>
<td>0.99</td>
</tr>
</tbody>
</table>

Reconciled values (CS-I)

<table>
<thead>
<tr>
<th>$x_1^m$</th>
<th>$x_2^m$</th>
<th>$x_3^m$</th>
<th>$x_4^m$</th>
<th>$x_5^m$</th>
<th>$x_6^m$</th>
<th>$x_7^m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.742</td>
<td>5.604</td>
<td>1.903</td>
<td>1.398</td>
<td>4.927</td>
<td>10.942</td>
<td>0.597</td>
</tr>
<tr>
<td>$e_{i1}^m$</td>
<td>$e_{i2}^m$</td>
<td>$e_{i3}^m$</td>
<td>$e_{i4}^m$</td>
<td>$e_{i5}^m$</td>
<td>$e_{i6}^m$</td>
<td>$e_{i7}^m$</td>
</tr>
<tr>
<td>5.09</td>
<td>2.01</td>
<td>-1.19</td>
<td>-3.98</td>
<td>1.49</td>
<td>-1.16</td>
<td>-3.16</td>
</tr>
</tbody>
</table>

Ratio of initial energy of system to final energy: $E_0/E_f=1000$.
The percentage error values $e_{i1}^m$ and $e_{i2}^m$ were calculated as \[\frac{\left| \left( x_{i1}^m - x_{i1}^m \right) \right|}{\left( x_{i1}^m \right)} \text{ and } \frac{\left| \left( x_{i2}^m - x_{i2}^m \right) \right|}{\left( x_{i2}^m \right)}\] respectively.

TABLE 3. Reconciled and corrupted values (30% Gaussian noise) of the process variables used in example 2

<table>
<thead>
<tr>
<th>Corrupted values</th>
<th>$x_1^m$</th>
<th>$x_2^m$</th>
<th>$x_3^m$</th>
<th>$x_4^m$</th>
<th>$x_5^m$</th>
<th>$x_6^m$</th>
<th>$x_7^m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e_{i1}^m$</td>
<td>3.713</td>
<td>4.699</td>
<td>1.305</td>
<td>1.528</td>
<td>3.680</td>
<td>0.080</td>
<td>0.622</td>
</tr>
<tr>
<td>$e_{i2}^m$</td>
<td>-17.72</td>
<td>-15.82</td>
<td>-29.13</td>
<td>4.95</td>
<td>-24.19</td>
<td>-17.98</td>
<td>0.90</td>
</tr>
</tbody>
</table>

Reconciled values (CS-I)

<table>
<thead>
<tr>
<th>$x_1^m$</th>
<th>$x_2^m$</th>
<th>$x_3^m$</th>
<th>$x_4^m$</th>
<th>$x_5^m$</th>
<th>$x_6^m$</th>
<th>$x_7^m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.103</td>
<td>5.367</td>
<td>1.860</td>
<td>1.078</td>
<td>5.041</td>
<td>11.833</td>
<td>0.593</td>
</tr>
<tr>
<td>$e_{i1}^m$</td>
<td>$e_{i2}^m$</td>
<td>$e_{i3}^m$</td>
<td>$e_{i4}^m$</td>
<td>$e_{i5}^m$</td>
<td>$e_{i6}^m$</td>
<td>$e_{i7}^m$</td>
</tr>
<tr>
<td>14.42</td>
<td>-3.85</td>
<td>-3.43</td>
<td>-25.96</td>
<td>3.84</td>
<td>6.89</td>
<td>-3.81</td>
</tr>
</tbody>
</table>

Ratio of initial energy of system to final energy: $E_0/E_f=1000$.
The percentage error values $e_{i1}^m$ and $e_{i2}^m$ were calculated as \[\frac{\left| \left( x_{i1}^m - x_{i1}^m \right) \right|}{\left( x_{i1}^m \right)} \text{ and } \frac{\left| \left( x_{i2}^m - x_{i2}^m \right) \right|}{\left( x_{i2}^m \right)}\] respectively.

Fig. 7. Performance of CS-I (example 2): effect of step size on the decrease in energy; step sizes constant for all variables, except MULTISTEP.

Fig. 8. Performance of CS-II (example 2): effect of number of layers on the decrease in energy; initial step size maximum, 0.1; 200 hidden layers.

4. Discussion of results

The results obtained with neural nets simulated on a computer appear to be comparable with those...
normally associated with traditional non-linear optimization methods, even though the neurodynamics used in these nets are relatively basic. If necessary the results can be improved by making use of more sophisticated neurodynamic functions. These functions could incorporate other stochastic procedures such as simulated annealing and its variants [27–29] or hill climbing terms [15] to avoid entrapment in local minima, while moving the system through state space. The real advantage of using neural nets for data reconciliation problems is the fact that they can be implemented in electronic hardware which could fully exploit the massively parallel architectures of the nets. By making use of analogue devices [9], which typically converge in the characteristic time of the artificial neurons (of the order of $10^{-4}$ to $10^{-3}$ s), rapid computation is possible [30].

Since this investigation was based on the use of simulated neural nets, and not actual analogue nets implemented in electronic circuits, no direct conclusions can be made with regard to the temporal aspects of the computational procedures. A rough estimate of the speedup is provided by Amdahl’s law, $S=P/(P(1-f)+f)$, where $S$ is the speedup factor, $P$ the number of processors working on the task, and $f$ the fraction of the programming code which can be executed in parallel [31]. The time consumed by computational overheads was estimated to be not more than approximately 5% for all three connectionist systems, and on this basis and the results of the optimization experiments, it was possible to estimate the speedup factors for the solution of the data reconciliation problems outlined in examples 1 and 2. These estimates are summarized in Table 4. Two different situations are highlighted in the table, namely the location of a solution (local minimum) of the problem, and the location of a global solution (or minimum) to the problem (by combining a stochastic procedure with a gradient descent or direct search method). The gradient descent methods (CS-I and CS-II) performed significantly better than the direct search procedure. As can be expected, the larger the problem, the more is gained by making use of these parallel strategies. It is also clear that the speedup factor is quite sensitive to the fraction of computer code that can be executed in parallel (estimated to be 95% in this investigation).

The problem posed in example 2 presented non-linearities of a higher degree than the problem discussed in example 1. This meant that smaller time steps had to be implemented to ensure a monotonic decrease in the energy of these systems, and as a result these systems took longer to converge than the bilinear two-product classification system. After approximately ten iteration steps or cycles (see Fig. 4) the energy of the bilinear system discussed in example 1 (nine variables) did not show further significant decreases. The energy or objective function of the system considered in example 2 (eight variables), however, decreased by approximately two orders of magnitude after 25 iterations (and showed a decrease of approximately three orders of magnitude after 40 iteration steps).

In Fig. 10 the CS-II system is compared with two other non-linear procedures used for the solution of the problem posed in example 2, i.e. that of Broyden [4,32] and the constant derivative approach of Knepper and Gorman [33]. From this graph it can be seen that the CS-II system initially (steps 1 to 3) decreases the value of the energy or penalty function faster than the other two methods. In subsequent iteration steps it loses ground, but in the end (steps 11 and 12) the advantage gained by the methods of Broyden and constant derivatives is largely eradicated. It should be borne in mind that this comparison can serve as a rough guideline only, since the central processing unit (CPU) times

![Fig. 9. Performance of CS-III (example 2).](image-url)

### Table 4. Estimated speedup factors for examples 1 and 2

<table>
<thead>
<tr>
<th>Example</th>
<th>System</th>
<th>Number of processors</th>
<th>Speedup factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search for first local minimum</td>
<td>1</td>
<td>CS-I</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>CS-I</td>
<td>8</td>
</tr>
<tr>
<td>Search for global minimum</td>
<td>1</td>
<td>CS-II</td>
<td>4200</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>CS-II</td>
<td>1600</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>CS-III</td>
<td>4200</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>CS-III</td>
<td>4200</td>
</tr>
</tbody>
</table>

*200 hidden layers.
associated with the execution of the iteration steps in the different algorithms cannot be compared directly. If anything, a comparison of actual CPU times could only be to the advantage of the CS-II system with its relatively simple computational procedures.

Another important factor that should not be overlooked is that the efficiency of the CS-II system is not affected significantly by an increase in the dimensionality of the process system, while other non-parallel procedures such as those depicted in Fig. 10 are very sensitive to increases in the size of the problem. In large systems consisting of hundreds or even thousands of variables, the use of the CS-II system or a similar variant warrants further investigation, since it has the potential to perform significantly better than any other traditional procedure.

5. Conclusions

In this paper the use of connectionist systems (which were simulated on a digital computer) for reconciling inconsistent measurement data is discussed. The following has been shown.

1. The measurements of flow streams and assays inconsistent with process models can be reconciled accurately by procedures based on the use of connectionist systems.

2. The use of connectionist systems can lead to a significant reduction in the computational effort needed to optimize data reconciliation procedures, especially in highly non-linear systems.

3. Even with small problems the performance of connectionist systems is at least comparable with that of conventional procedures.

4. In problems of high dimensionality, procedures based on the use of connectionist systems are much more efficient than those based on conventional strategies.

References


Appendix A: Nomenclature

\( A \)
- Incidence matrix of a process circuit

\( e_i^m \)
- Percentage error in measured variable (example 2)

\( e_i^u \)
- Percentage in unmeasured variable (example 2)

\( E \)
- Scalar energy or cost function serving as a measure of the overall state of a Hopfield net

\( \epsilon \)
- Convergence criterion

\( \epsilon_r \)
- Convergence criterion associated with rate of change of energy in connectionist system CS-II

\( f(\cdot) \)
- The fraction of computer code that can be executed in parallel

\( f_i(\cdot) \)
- Set of process constraints

\( F_i \)
- Flow rate of process stream \( i \)

\( \phi \)
- Random number in general

\( \phi_{ij} \)
- Random number associated with variable \( i \) at time \( j \) (0 \( \leq \phi_{ij} \leq 1 \))

\( h \)
- Step length in numerical integration routine

\( H_i \)
- Enthalpy of stream \( i \)

\( i_j \)
- Scaled network input corresponding to the actual input \( x_j \)

\( m_k \)
- Minimum value that measurement \( x_k \) can assume

\( M_k \)
- Maximum value that measurement \( x_k \) can assume

\( n \)
- Gain of a sigmoidal activation function

\( P \)
- Number of processors or processing elements in a parallel computational structure

\( r_i^j \)
- Search range associated with variable \( i \) at time \( j \)

\( R_{\text{min}} \)
- Minimum value of scaled variable \( x_i \) in output layer

\( R_{\text{max}} \)
- Maximum value of scaled variable \( x_i \) in output layer

\( R_{i,\text{min}} \)
- Minimum value of scaled variable \( x_i \) in output layer

\( R_{i,\text{max}} \)
- Maximum value of scaled variable \( x_i \) in output layer

\( s_{ki} \)
- Standard deviation of assay datum \( X_{ki} \)

\( \sigma_{ki} \)
- Time increment, equivalent to an iteration step in terms of the computational algorithm

\( \Delta t \)
- Time

\( u_i \)
- Output state of computational element \( i \) in a neural net

\( u_i(\cdot) \)
- Activation function

\( \nu \)
- Set of output states of computational elements in the neural net

\( \nu^0 \)
- Initial output states of computational elements in the neural net

\( \nu^* \)
- Initial output states of computational elements in the neural net

\( \nu^t \)
- Initial output states of computational elements in the neural net

\( V \)
- Variance–covariance matrix of the process variable measurements \( \sigma \)

\( W \)
- Weight associated with the material balance constraints incorporated in the energy function \( E \)

\( x_j \)
- Output of neural net (adjusted value of \( x_j \))

\( x_{ij} \)
- Measured concentration of component \( j \) in flow stream \( i \)

\( x_{ij}^* \)
- Adjusted concentration of component \( j \) in flow stream \( i \)

\( x^* \)
- Measurements of process variables of a plant or circuit

\( x_{ij}^t \)
- Adjusted values of process variables
Process Modeling with the Regression Network

Tjaart van der Walt, Etienne Barnard, and Jannie van Deventer

Abstract — A new connectionist network topology called the regression network is proposed in this paper. The structural and underlying mathematical features of the regression network are investigated. Emphasis is placed on the intricacies of the optimization process for the regression network and some measures to alleviate these difficulties of optimization are proposed and investigated. The ability of the regression network algorithm to perform either nonparametric or parametric optimization, as well as a combination of both, is also highlighted. It is further shown how the regression network can be used to model systems which are poorly understood on the basis of sparse data. A semi-empirical regression network model is developed for a metallurgical processing operation (a hydrocyclone classifier) by building mechanistic knowledge into the connectionist structure of the regression network model. Poorly understood aspects of the process are provided for by use of nonparametric regions within the structure of the semi-empirical connectionist model. The performance of the regression network model is compared to the corresponding generalization performance results obtained by some other nonparametric regression techniques. Although sigmoidal backpropagation neural networks performed relatively satisfactorily for the modeling of this system, the semi-empirical regression network models outperformed the sigmoidal backpropagation network models. This superiority of the semi-empirical regression network models became even more prominent with a decrease in the population density of training data. This is indicative of a remarkable improvement in extrapolative properties of the semi-empirical regression network over other regression models investigated for the relevant application.

1. INTRODUCTION

MODELING research in process engineering is motivated by the fact that the efficiency of many processing operations can be linked directly to the adequacy of predictive models for the behavior of such processes. With the rapid explosion of technology in the field of process engineering, a higher level of sophistication is required in the design of processing equipment and in the operation of industrial processes. Process control strategies are becoming more advanced and the possibilities for implementation of such improved control methods are supported by the rapid enhancement of computational power. It is therefore important to construct accurate models of processing systems in order to facilitate optimal process control and processing plant design.

The fundamentals of most processing operations are poorly understood. This demands modeling methods with the ability to capture relations between response and predictor variables with limited a priori knowledge about the physics of the process. A model for such a process should therefore contain satisfactory robustness in its functional form so that a priori inexplicable behavior of the process can be predicted adequately. For this reason nonparametric modeling offers an appealing alternative to fundamental and empirical modeling. Unfortunately, for many industrial processing operations, process data are expensive and in some cases difficult to obtain. Since nonparametric modeling techniques generally experience difficulty to perform adequate function approximation in predictor variable spaces which contain process data of low population density, such techniques are usually not able to model systems in the processing industry if the availability of process data is limited. This phenomenon is a major concern to the modeling engineer and is often referred to as the curse of dimensionality [1].

The various nonparametric regression techniques suffer in varying degrees from the curse of dimensionality. Although three-layered sigmoidal backpropagation neural networks can emulate any arbitrary function [2], many nodes and therefore many free model parameters are usually necessary to approximate a functional relation. A simple linear relation for example needs a relatively complex sigmoidal model to describe the relation adequately in the case of a sigmoidal backpropagation network used as model. This can cause weak generalization properties when limited training data are available owing to the large number of parameters in the network model, while a simple linear equation will be able to fit the relation with a minimal number of model parameters and improved generalization performance.

Although a truly fundamental model which represents the true chemistry and physics of a process still remains the ultimate goal and is completely impervious to the problem of high dimensionality, it is not possible in most cases to formulate such fundamental models, because of a lack of understanding of the process. In some cases, however, some knowledge about the process mechanism is available, while some other features of the system to be modeled are either partially understood or not comprehended at all. Although various regression techniques are available such as neural nets (different topologies) [3], MARS [4] and nonlinear time series techniques [5], these techniques cannot use such knowledge which is available prior to training.

In this paper a new neural network topology which we call the regression network is proposed. This connectionist network deserves its general name due to its wide applicability as regression tool. In the first place it can be used as a parametric regressor to estimate the model parameters.
of a known relation which needs to be fitted to a set of data. It can also be employed to perform pure nonparametric regression if the system to be modeled is ill-defined in toto (it can be used for example as any multilayer perceptron or radial basis function network). Thirdly the regression network provides a connectionist framework whereby fundamental knowledge of a system can be combined with the attractive nonparametric properties of existing connectionist networks. For this application, the regression network can be used to construct a regression model which contains both parametric and nonparametric parts corresponding respectively to a priori knowledge and aspects which are poorly understood. The use of mechanistic knowledge in such a model inhibits the freedom of the model so that the generalization properties of the model are improved accordingly.

Due to its connectionist configuration, the regression network can be specified to represent almost any arbitrary functional relation. The model parameters of the regression network are estimated through gradient optimization in a completely general fashion. The first-order partial derivatives of the model parameters are determined during the optimization process by means of backpropagation. However, the training process of the regression network seems to be very complex in some cases. Although much still needs to be learned about the problems experienced by the network during training, some problems are identified and intrinsic causes to these complications are defined. Possible solutions to the problem are also suggested.

The following main trends of this paper can be emphasized:

- A new objective function, i.e., the relative LMS function, is proposed to be used if the sensitivity of a response variable is more or less proportional to its norm.
- Possible solutions to the complex optimization process of the regression network for some systems are further proposed. The training performance of some other minimization techniques is investigated and compared with the performance of conjugate-gradient optimization. Some general guidelines in the quest for finding the optimal solution of the regression network are suggested.
- In the final part of this paper it is shown how semi-empirical regression network models can be developed to estimate the performance of a typical metallurgical processing operation, i.e., a hydrocyclone classifier, of which the fundamentals are poorly understood. The generalization performance of the semi-empirical regression network models are compared to the modeling performance of other nonparametric regression methods.

II. CONFIGURATION OF THE REGRESSION NETWORK

The regression network can be viewed as a framework by which any functional form can be represented using a connectionist architecture. The connectionist structure consists of a number of nodes which are interconnected in a feedforward fashion. Depending on the functional form, the network can either be layered or non-layered. As for sigmoidal regression networks and other layered connectionist networks, the regression network contains an input layer of nodes. One of the input nodes is the bias node with constant value of unity. Each of the other input nodes represents an input variable. Thus far, the description could apply equally well to conventional backpropagation networks. What distinguishes the regression network is that considerable flexibility exists in the choice of node behavior. Each non-input node in the network is characterized by an input type and transfer function. The input to a specific node can either be an additive input or a product input as defined in Table I. Note that the nomenclature used in Table I is explained graphically in Fig. 1. The activity and input to a node \( q \) are represented by \( y_q \) and \( x_q \) respectively,
while \( w_{pq} \) is the weight of the connection between two nodes \( p \) and \( q \). Fig. 1 also describes the connectionist structure between successive nodes in the feedforward regression network.

Various node transfer functions can be used. A few node functions are listed in Table I. It should be noted that a very wide range of functional relations can be represented by the regression network by employing the transfer functions of Table I in conjunction with the two node input types.

Each node is named after both its input type and transfer function type. If a node is characterized by a product input and an exponential transfer function, the node is called an exponential-product node and is referred to as an \([e-p]\) node in short. The abbreviations used for the transfer functions and input types are also given in Table I. According to this notation the hidden nodes of a sigmoidal backpropagation network are sigmoidal-additive \((s-a)\) nodes.

Any combination of connections for the regression network can be specified as long as a feedforward configuration is maintained. If some parameters of the network function are known, the weight values for such connections can be specified prior to training. Such weights are then kept constant during the optimization process. More importantly, prior knowledge regarding the regression function to be modeled is incorporated by choosing appropriate node types and connection topologies. Thus, if a certain variable is, for instance, known to enter linearly in a regression function, that variable will be connected to an output \([l-a]\) node.

**A. Backpropagation for the Regression Network**

During the training process the network weights can be adjusted in order to optimize an error function using a gradient optimization technique. If a local gradient optimization procedure is used, the first-order partial derivatives of the error function in weight space are determined with backpropagation. The first step of a single training iteration the error gradient of the output node (referred to as node \( m \)) for a network containing \( m \) nodes is calculated. (Hence, the nomenclature used in this paper to label the output node corresponding to the response variable is \( m \).) The LMS criterion function and its error gradient for a training data set containing \( n \) data points is given by (1a) and (1b), where \( d_k \) represents the actual functional value prescribed for the \( k \)th training data point.

\[
E = \frac{1}{2} \sum_{k=1}^{n} (y_{m,k} - d_k)^2 \tag{1a}
\]

\[
\frac{\partial E}{\partial y_m} = \sum_{k=1}^{n} (y_{m,k} - d_k) \tag{1b}
\]

The error is represented by \( E \), while \( y \) is the activity of the relevant node of the regression network. In subsequent expressions the input to a network node is depicted by \( x \).

The second step in the backpropagation procedure will be to determine the error gradient in weight space for each connection with weight \( w_{pq} \) which connects node \( p \) to the output node \( (y = m) \).

\[
\frac{\partial E}{\partial w_{pq}} = \frac{\partial E}{\partial y_q} \frac{\partial y_q}{\partial x_q} \frac{\partial x_q}{\partial w_{pq}} \tag{2}
\]

With the error gradient with respect to the activity of a node \( q \) known, (2) can be used again to calculate the error-weight gradients of all the incoming weights to the relevant node \( q \). This procedure is repeated for all nodes until the input layer is reached.

For a node \( q \) with a product input, the input value \( x_q \) and its partial derivative for a specific weight can be calculated using two different approaches. Firstly the upper input node activity values can be restricted to be positive, so that training is performed with the norms of the inputs to the product node only. It is however preferable to consider negative inputs because the nonlinear characteristics of product nodes are centered on the origin. There are two main alternatives to dealing with the resulting complex input values. The whole network can be handled in the complex domain during training, whereas if the real component can be fitted to the data by either ignoring the imaginary component or by fitting it to 0. Durbin and Rumelhart [6], however, found that for most problems it is preferable to keep the system in the real domain by ignoring the imaginary component of the output from each product node. It was also found that no benefits result from working in the complex domain, although the solution is more complicated. The approach to work along the real axis of the complex domain can be used in this regression network in the presence of product nodes [6]. This proved to be an overall improvement over the method of using absolute input values. Although this technique seems to work adequately for many optimization problems, there are many complications experienced during training which will be dealt with now in more detail.
III. Complications in the Training Process

As in the case of many regression techniques, training is equivalent to optimizing an error function in a model parameter space. For this regression network, however, training can become very complex. This complexity in training is caused mainly by two factors. In the first instance, for most systems the error surfaces contain many non-convex regions so that the weight matrix of the regression network can become trapped in numerous shallow local minima. Although the problem of local minima is a common and well-studied optimization problem for most regression techniques, the problem can become extremely difficult to overcome if relations with an exponential nature are present. A study of the error surfaces for the LMS criterion function also revealed that the error surfaces contain numerous flat areas or “plateaus” for some functional relations. This phenomenon was also identified by Bilbro and Snyder [7]. Due to the sensitivity of the exponent, the LMS energy space possesses the large plateaus.

A second cause of the optimization problems experienced by some regression networks containing exponential features was found to be its inability to move smoothly from the real to the complex domain during training, as required for product nodes.

The conjugate-gradient optimization method with restart procedures [8] is a well-established optimization technique for backpropagation connectionist networks and was used as the point of departure in this optimization study. A few simple target functions are used below to illustrate the optimization phenomena and findings. The functional forms of these examples are listed in Table II, while their connectionist configurations are portrayed in Fig. 2. For each of these examples, one or more training data sets were generated and these unscaled data used to train the regression network. No noise was applied to the data for these experiments.

A. The “Relative” LMS Criterion Function

For many systems where the degree of interaction between the independent variables is high, the function values vary over different orders of magnitude. In some cases there can also be a change in sign for the function values. For such systems logarithmic scaling of data for the purpose of regression becomes inappropriate owing to a loss in sensitivity with a transformation of all real data to the positive domain. For these reasons it will be preferable to relate the influence of each training data point during training to the norm of its function value, rather than to quantify the influence of a data point relative to the absolute deviation of the predicted functional value from the actual value. The latter is done if the conventional LMS criterion function is fitted to the training data. In doing so the necessity to use logarithmic preprocessing for the values of the independent variables also becomes unnecessary. The relative LMS criterion function scales the conventional LMS error by the norm of the actual value of a
response, and is given by

\[ E_{rel} = 1/2 \sum_{k=1}^{n} \frac{(y_{m,k} - d_k)^2}{\|d_k\|^2}. \]  

(4)

Unless specified otherwise, the relative LMS criterion function was employed for all experiments in this optimization study.

B. Investigating the Error Surface

In order to gain some insight into the optimization problems experienced by the regression network, the error surface can be studied for specific networks by plotting the error surface between and in the vicinity of two different locations in weight space. If \( \hat{w}_1 \) and \( \hat{w}_2 \) are the locations of two minima in weight space (either a local and global minimum, or two local minima), then the error surface in the plane which intersects both coordinates is given by

\[ E = E(\hat{w}_1 + a \Delta) \]

where

\[ \Delta = \hat{w}_2 - \hat{w}_1 \]

and

\[ a \in (-\beta \cdots 1 + \beta) \].

(5)

The fraction \( \beta \) determines the boundaries of the error surface to be investigated in the direction \( \Delta \). To allow for more sensitive evaluation close to \( \hat{w}_1 \) and \( \hat{w}_2 \) the change in step size of \( \beta \), i.e., \( \delta \), can be adapted within the range \((-\beta \cdots 1 + \beta)\) as follows:

\[ \delta = \frac{\|A\|^C_T + C_T}{C_N} \]

where

\[ A = \begin{cases} a, & a < 0.5 \\ a - 1, & a > 0.5 \end{cases} \].

(6)

A reduction in the positive constant \( C_T \) will allow for more sensitive calculation of error values close to the relevant two coordinates, while the denominator \( C_N \) determines the average step size.

Although the error surfaces of numerous regression networks have been investigated, error surface plots are given here only for examples (d) and (e) of Table II, Fig. 3(a) and (b) contain error plots for example (d). For this experiment 25 training runs were performed with different initial weight matrices. Fifteen training data points were used and were generated at random within the variable ranges \( x_1, x_2 \subseteq (0 \cdots 2) \). With conjugate-gradient (CG) optimization the rate of success was quite high in this case with 48% of the training runs finding the global minimum. The weight matrices of two unsuccessfully trained regression nets (weight matrices \( \hat{w}_d \) and \( \hat{w}_e \), respectively) were used together with the weight matrix \( \hat{w}_a \) of a well-trained network to plot the error surfaces of Fig. 3. These three networks were selected, since they seem to typify the error surface behavior for the specific problem.

The Euclidean distance \( \|\Delta\| = \|\hat{w}_d - \hat{w}_e\| \) in Fig. 3(a) is 248 (norm between the global minimum at \( \hat{w}_d \) and the location of the second network—i.e., \( \hat{w}_e \)). The norm of \( \|\Delta\| \) of the global minimum and the third network in Fig. 3(b) amounts to 5.9. This shows how far these three networks are separated in weight space if the order of magnitude of the model parameters of example (d) is considered. The weight matrix of the second network is located at a “far-off” plateau with \( E_{rel} \approx 293 \) as can be observed from Fig. 3(a), while the weight matrix of the third network became trapped in a fairly bad error domain. This can be seen in Fig. 3(b) which shows that the weight matrix converged into a local minimum and that the regression process was terminated against an error wall.

For example (e) the same experiment was done with 150 training runs using the LMS criterion function with no success at all. If the relative LMS error was however fitted to the data, 8% of the networks converged into reasonably acceptable local minima (i.e., local minima with fairly small error values). Although CG optimization was still not able to locate the global minimum, this is a significant improvement over the training performance with straightforward LMS data fitting. Two error surface plots between the weight matrices of three of the networks found during a relative LMS fit (viz. \( \hat{w}_d, \hat{w}_e, \) and \( \hat{w}_f \)) can be seen in Fig. 4. The Euclidean distances between the relevant two networks used for the first and second plots are \( \|\hat{w}_d - \hat{w}_e\| = 14.8 \) and \( \|\hat{w}_d - \hat{w}_f\| = 12.0 \), respectively, while \( \|\hat{w}_e - \hat{w}_f\| = 8.9 \) (the latter being the Euclidean distance between the two networks with respective errors of 339 and 117). From Fig. 4(a) it can be seen that the relative LMS error surface in the weight space for a complex function such as example (e) contains numerous gorges so that the network can easily become trapped in local minima. Another very common feature of the error surface for this example is the existence of extended plateaus. Although the Euclidean distance between the networks of Fig. 4(b) is relatively large, a very flat plateau can be observed within this plane. It is also interesting to note that for the network with an \( E_{rel} \) of 1.15 a steep error gorge exists in weight space in the plane of Fig. 4(a), while the error
The error surface is found to be relatively flat in the plane displayed in Fig. 4(b). Although no plots are shown here, the error surface in the vicinity of the network with $E_{rel} = 1.17$ is flat also in the direction of the regression network with $E_{rel} = 339$ (i.e., in the direction of $\tilde{\omega}_u - \tilde{\omega}_f$).

The reason for the existence of error plateaus if a function contains intrinsic exponential features is that exponential relations are switched on very strongly in certain regions in weight space, so that the other parameters become ineffective. Although the mountainous regions in error surface of some complex functions (such as the error surface of example (e) as displayed in Fig. 4(a)) can be linked to the complicated interaction between the different node functions of a complex functional relation, the precise cause for this phenomenon has not been identified yet.

In order to develop a better understanding of the intrinsic causes of this optimization problem, experiments were conducted to evaluate the training performance also for simpler functions. The results and general findings of these experiments are given here only qualitatively. The CG optimization algorithm does not experience any trouble in finding the global minimum for either a linear or product function with multiple variables (in this case the regression network contains only a single [l-a] or a [l-p] node together with the nodes in the input layer). If two or more intrinsic product relations are combined in an additive way, this causes error plateaus in weight space. The same results were found for functions containing exponential nodes. For a function described by a single [e-a] node with some additive variables, the error surface contains huge plateaus and no mountainous regions. Compared to this, the parameters of simple exponential functions with multiple and interacting variables (a single [e-p] node besides the input layer) can be estimated successfully without exception. If no additive node is present in a network with more than one exponential type of relation, no plateaus exist in the error surface. The valleys in the error surface also seem to be smoother compared to the error surface of a function which contains additive as well as exponential relations.

Some general conclusions can be formulated from the results of the above mentioned and other experiments for more complex functions. Error plateaus exist in weight space for any function that contains one or more intrinsic exponential relations which are combined additively. With an increase in complexity of such a combination of intrinsic exponential and additive features there is a corresponding increase in the frequency of occurrence of mountainous error regions within weight space. The optimization problems for complex regression networks with exponential features are collectively caused by plateaus and mountainous regions of the error surface in weight space.

C. Investigating the Intrinsic Behavior of Some Network Nodes

Studies into the intrinsic behavior of the nodes for various complex regression networks revealed that a network sometimes experiences trouble to adapt itself by moving from the real into the complex domain. This phenomenon can be identified by investigating the response of specific nodes of a network to certain data samples through the weight space. Fig. 5 gives the activity plots of the [l-p] node marked by an asterisk * in Fig. 2(c) for example (c). These activity plots were determined along a plane in weight space as was previously done to calculate the error surface plots. It should be noted that the activity plots represent the node activity responses to individual training data points. From Fig. 5 a discontinuous activity pattern is observed at the location of the second unsuccessfully trained network. The optimization process came to a halt at the transition level between the real and complex domains of this product node. Although it is not shown in this paper, it can be mentioned that the corresponding node activity plots for the upper [l-a] and [l-p] nodes in Fig. 2(c) are continuous and uncomplicated. The complex behavior of the [l-p]* node is propagated to the output node, as can be seen from the outlying peak in the error surface plot for this example in Fig. 6. This optimization problem is caused by a discontinuity in the partial derivative of the node activity at the boundary between the real and complex domains.

IV. Possible Solutions to the Optimization Problems

It was shown in the previous section that the training process of the regression network can be very complicated because of complicated error surfaces. In this section the performance of some other optimization techniques than CG are investigated to alleviate the complexity of the optimization process to some extent.

For reasons discussed earlier, conjugate-gradient (CG) optimization [8] has limited success in finding global minima for this regression network. CG experiences difficulty to estimate the
the model parameters of functions of which the error fit contains numerous local minima. For the cases where the weight matrix of a regression network converges onto a plateau during optimization, CG is normally incapable of finding an escape route so that the training process terminates on the plateau.

A few optimization techniques other than CG were investigated in order to enhance training performance. The regression networks for examples (a), (b), (c), and (e) of Table II were used to investigate the training performance of the different optimization techniques. Note that an additional and unnecessary [l-α] node was specified for example (a) as can be seen in Fig. 2(a). This was done to investigate the ability of the network to identify and eliminate unnecessary parameters from the network structure.

A. Alternating Projection

Alternating projection (AP) is a global optimization approach which searches for minima in a D-dimensional bounded weight space $S \subset \mathbb{R}^D$. This search process can be viewed as a circular procedure during which the error surface is sequentially evaluated on a grid in each dimension $k \leq D$ in $S$. This is the case for one-dimensional AP. For AP of higher dimensionality $n$ the error surface is sequentially searched through $D$ hyperplanes, each of dimensionality $n$. These hyperplanes are orthogonal in relation to each other in weight space in the sense that for a search within a specific dimension $k$ the weight values of all dimensions $i \neq k$ for $i \leq D$ are kept constant. Algorithms for one- and two-dimensional alternating projection have been developed.

Experiments were performed using the one- and two-dimensional AP algorithms. As is the case for most search optimization techniques, AP is time-consuming. The time required to perform a single AP iteration or circle increases exponentially with an increase in the AP-dimensionality. Both one- and two-dimensional AP were however used in alternation with CG.

For each training run five alternations between AP and CG were performed, always starting with AP. For each alternation five AP iterations were completed. The maximum step size for the AP search was specified for all experiments at $\eta_{\text{max}} = 0.2$, while the weight space was constrained between $-15$ and $15$ for all weight dimensions. The training results are listed in Table III. From these results for examples (a) and (b) it can be seen that both one- and two-dimensional AP in some cases initialized the weight matrix into regions from where it was impossible for CG to find the global minimum, although CG was able to locate the optimal solution on its own (refer to the runs listed in brackets). In other instances where CG was unable to find the global minimum, AP however initiated an adequate weight matrix from where CG was able to converge down to the global minimum. Hence, if adding the additional percentage of successful runs indicated in brackets in Table III to the success rate for AP and CG, it can be observed that the net success rate of training the relevant regression networks can be enhanced by using both the "standalone" CG optimization technique, as well as AP and CG in alternation.

All success rates for example (e) are local minima which are relatively close to the global minimum. The global minimum was not found in any run. Although AP optimization used in alternation with CG was unable to locate optimal solutions for examples (c) and (e), it was observed that deeper local minima were in general found as compared to the training results of CG used on its own. It was also found that in only a few isolated instances an AP run was able to force the weight matrix out of a plateau or local minimum during the second to the fifth alternating runs between CG and AP. According to this, no more than two alternations between CG and AP are necessary if AP is to be used.

It can therefore be concluded that AP in general is unable to overcome the optimization problems of the regression network. It can however find local minima which lie closer to the optimal solution for some complex functions than can be found with CG used on its own.

B. Tree Annealing

The performance of another global optimizer, i.e., tree annealing (TANN) was investigated. Like AP, TANN is a global search optimization method which is constrained between

<table>
<thead>
<tr>
<th>example</th>
<th>number of training samples</th>
<th>rate of success</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>CG alone</td>
</tr>
<tr>
<td>(a) 15</td>
<td>66%</td>
<td>56% (+24%)</td>
</tr>
<tr>
<td>(b) 15</td>
<td>24%</td>
<td>32% (+16%)</td>
</tr>
<tr>
<td>(c) 10</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>(d) 4%</td>
<td></td>
<td>4%</td>
</tr>
<tr>
<td>(e) 25</td>
<td>4%</td>
<td>4%</td>
</tr>
</tbody>
</table>

(Note: The additional successful runs shown in brackets are for the runs where CG had success and AP & CG met.)

Fig. 5. Node activity plots of some data points for the [l-α] node of the regression network for example (c) indicated within Fig. 2(c).
search boundaries. The method is described by Bilbro and Snyder [7]. It is based on the principles of simulated annealing, but handles deterministic relations in a more natural way. During the optimization process, a search tree is built on the basis of subdividing the weight space in the vicinity of more promising regions into more sensitive hyperrectangular subspaces. After subdivisions are made, the probability to move down specific branches to seek a solution within the more promising regions is increased.

There are some disadvantages to this optimization technique. Firstly TANN is memory intensive owing to the creation of two more nodes each time a promising position in weight space is identified. Secondly the probability to walk down a wrong branch can become dominant so that the optimal solution becomes hidden to the optimizer. Another disadvantage of TANN is its sensitivity to the initial temperature \( T \), while the way in which \( T \) is annealed is also very important. From experiments with some relatively simple regression networks it was found that a relatively small initial value for \( T \) can improve the adequacy of the tree building process in the initial stages of training, which is a positive step to enhance the possibility of finding the global minimum.

Since factors such as choice of initial \( T \) and the order in which the constraint model parameter space should be divided into hyperrectangular spaces play a decisive role in the overall success of TANN in this application, a detailed TANN optimization study is required. Such a statistical analysis is beyond the scope of this paper, so that no optimization results of TANN have been listed in Table III.

Although Bilbro and Snyder [7] remarked that TANN can optimize functions of up to 3 variables, this was not the case for the regression networks. As soon as the number of model parameters exceeded about six, TANN was usually unable to estimate the parameters successfully. However, it should be emphasized that TANN is the only optimization method investigated in this study that was able to regress simple functions containing harmonic relations. Hence, there is a place for TANN in the regression network’s optimization library.

C. Leap Frog Dynamic Optimization

Another gradient optimization technique which was investigated is the dynamic leap frog minimization method of Snyder [9]. The search trajectories for this unconstrained method are derived from the equation of motion of a particle in a conservative field (e.g., gravitation), where the function to be minimized represents the potential energy. The trajectories are modified to increase the probability of convergence to a comparatively low local minimum. This results in a corresponding increase in the regions of convergence of the global minimum. A full description of the optimization algorithm is given in [9].

The training performance of leap frog is compared with the performance of the other techniques in Table III. Leap frog optimization enhanced the training performance in general. However, this technique was also not able to locate the global minimum of the network for example (c) with 15 training data points. In this experiment the leap frog trained networks converged onto high plateaus for almost all training runs, while CG-trained networks became trapped in local minima much closer to the optimum. For the training runs with 10 data points, leap frog showed a significant improvement over CG and AP. The 24% “successful” searches produced by leap frog for example (e) again represent a number of relatively good fits to the data; the global minimum was not found in any of these training runs. The results for example (e) however show that leap frog can be employed more confidently for the regression of functions of such complexity as example (e).

For all these examples leap frog in some cases has the ability to modify a weight matrix which converged onto a plateau during CG optimization off the plateau into better regions. This is however not a general phenomenon. It was found that leap frog succeeded in leading the network weights off a plateau if the distance between the weight matrix and the origin (0-matrix) in weight space is relatively small. For the cases where the error fit converged into “far-off” asymptotic plateaus, i.e., the Euclidean distance between the weight matrix and the origin is large, leap frog was unable to find a way back to the global minimum. For the experiments with 15 training samples of example (c) it was also found that leap frog was able to take some relatively good CG-trained networks and improve their training, although it was incapable of reaching lower error levels on its own.

V. Guidelines in the Search for the Optimal Solution of a Regression Network

In conclusion a few guidelines are proposed that can be used in conducting a search for the optimal solution of a regression network.

i) Select the appropriate criterion function (either LMS or relative LMS) according to the intrinsic nature of the functional relation to be investigated as discussed in Section III-A.

ii) If the regression network contains harmonic nodes, use TANN optimization. For any other function, perform a few training runs from different initial positions in weight space with CG.

iii) If no success with CG, repeat

iv) with the leap frog optimization routine.

v) If unsuccessful with leap frog, investigate the error surface in the vicinity of the weight matrices of the CG-trained networks. For each network which converges onto a plateau, evaluate the network’s weight matrix to see if the network did not converge into “far-off” regions onto an asymptotic plateau. If so, use leap frog in an attempt to get the network weight matrix off the error plateau.

Although these few guidelines can be followed in the quest for a solution of a regression network for a complex system, an optimal solution is not guaranteed. Much still needs to be learned about the intricacies and behavior of the regression network during training. The development of a unique optimization routine as solution to the complications experienced during training of the regression network should not be excluded. Such a routine can typically contain heuristics to detect and prevent convergence into “far-off” regions at
an early stage, thus alleviating the optimization problems experienced owing to the existence of plateaus.

In the remainder of this paper it will be shown how the regression network algorithm can be used in both parametric and nonparametric fashions, as well as in a combined way to construct models for a metallurgical processing system of which the fundamentals are not understood adequately.

VI. MODELING THE PERFORMANCE OF A HYDROCYCLONE CLASSIFICATION PROCESS

Hydrocyclones are widely used as classifiers in wet grinding circuits in the mineral industry and they have a significant influence on the performance of these circuits. It is important to be able to predict the behavior of the hydrocyclone adequately for any changes in process conditions. This requires the formulation of a suitable model for hydrocyclone operation. Although simplistic empirical and fundamental models are available, the intrinsic behavior of the process is most complex and difficult to predict on the basis of such models. For this reason there is a need for an adaptive modeling strategy which has the ability to model tendencies which cannot easily be described in a fundamental fashion. It is illustrated here how models for the efficiency of the hydrocyclone classification process can be constructed with the regression network algorithm. The modeling performance of these regression network models is compared with the models for hydrocyclone efficiency of some other regression techniques.

A. Semi-Empirical Model for the Performance of the Hydrocyclone Process

Owing to the absence of published real data (real process data are not readily made public in the processing industry) the semi-empirical model of Lynch and Rao [10] was used to generate typical process data which in turn could be employed by the different regression techniques to approximate the model. This semi-empirical model relates the operating and design variables of the system to the efficiency of the separation process. The configuration of a hydrocyclone is schematically presented in Fig. 7. The design variables in a hydrocyclone are the diameters of the vortex finder, spigot and inlet, while the operating variables are the flow rate, percent solids and the size distribution of the solid particles in the pulp.

The model of Lynch and Rao [10] postulates that the two major mechanisms by which particles enter the coarse product are (i) classification and (ii) entrainment or short-circuiting. The size of a particle which has an equal probability of moving either to the overflow or the underflow due to centrifugal action in the cyclone only, is generally defined as \( d_{50}^{(C)} \) and can be estimated by

\[
\log_{10} d_{50}^{(C)} = k_1 \text{VF} + k_2 \text{SPIG} + k_3 \text{INLET} + k_4 \text{FPS} + k_5 \text{LPM} + k_6.
\]  

(7)

This expression contains the necessary model information for the fraction of the particles entering the product due to classification, \( \text{VF} \) [cm], \( \text{SPIG} \) [cm] and \( \text{INLET} \) [cm] are the design variables and represent the diameters for the vortex finder, the spigot and the inlet respectively. The respective operating variables \( \text{FPS} \) and \( \text{LPM} \) \([\text{L min}^{-1}]\) are the percent solids in the feed by weight and the flow rate of the pulp feed.

Particle entrainment in any size is regarded by the semi-empirical model as being entirely dependent on the water distribution between the vortex finder overflow and the product underflow and can be described by

\[
R_f = K_1 \frac{\text{SPIG}}{\text{WF}} + K_2 \frac{\text{WF}}{\text{WF}} + K_3
\]  

(8)

where \( R_f \) is the percentage feed water reporting to the underflow product and \( \text{WF} \) \([\text{L}]\) represents the flow rate of water in the feed.

\( \text{WF} \) can be calculated with a mass balance expression in terms of \( \text{LPM} \) and \( \text{FPS} \), if it is assumed that the ore is homogeneously suspended within the water phase of the slurry feed to the hydrocyclone.

\[
\text{WF} = 0.06 \text{LPM} \cdot \frac{\text{SG}_{0}^{100-FPS}}{1 + \text{SG}_{0}^{100-FPS}}
\]  

(9)

The corrected efficiency curve is used to describe the efficiency of classification and an expression commonly employed for this purpose is

\[
\gamma^{(C)} = \frac{e^{aX} - 1}{e^{aX} + e^a - 2}.
\]  

(10)

The argument \( X \) equals \( d/d_{50}^{(C)} \) for a particle size \( d \), whereas the parameter \( a \) gives an indication of the sharpness of
separation and therefore the efficiency of classification. A corrected efficiency curve can be converted into a corresponding practical efficiency curve by (11) if \( d_{50}^{(C)} \) and \( R_f \) are known.

\[
Y = r_f + (1 - r_f)Y^{(C)}
\]

(11)

Typical curves for the corrected and practical efficiencies at specific process conditions are depicted in Fig. 8.

The six-input variable model for \( Y \) in terms of SPIG, INLET, VF, FPS, LPM, and \( d \) was used as basis of comparison in this study. Lynch and Rao [10] used parameter values of \( k_1 = 0.0419, k_2 = -0.071, k_3 = 0.0467, k_4 = 0.0406, k_5 = -0.0001 \) and \( k_6 = 0.33 \) in (7), as well as \( K_1 = 125.7, K_2 = -213.9 \) and \( K_3 = 6.67 \) in (8) for a coarse size distribution. The data points employed by Lynch and Rao [10, fig. 5] to construct their figure of corrected efficiency plots for some ore types were used to estimate \( a \) in a parametric fashion with the regression network. For a limestone with specific gravity \( SG_0 = 3.25, a \) was estimated to be \( a = 5.0 \), which represents a relatively sharp separation process. These model parameters were selected for the model used in this study.

B. Generation of Typical Plant Data

Typical plant data for the performance of the hydrocyclone classifier were generated using the semi-empirical model described in the previous section. The ranges of all input variables except the particle size \( d \) are listed in Table IV and were taken from [10, table 4]. The training data which were employed by different regression techniques to perform model building were generated as follows: For each data series the design variables of the hydrocyclone were kept constant (eleven different hydrocyclone geometries were used), while values for FPS and LPM were randomly selected within the specified ranges. For each combination of the five input variables listed in Table IV a particle size distribution consisting of ten values for \( d \) (i.e., 1180, 850, 600, 425, 300, 212, 150, 106, 75, and 53 \( \mu m \)) was used so that ten data points in the training data sets actually represent a single experimental sample taken at fixed processing conditions for a specific hydrocyclone.

Three different training sets were constructed for each of \( Y^{(C)} \) and \( Y \). These data sets contained 420, 210, and 70 data points, respectively. A test data set for each of \( Y^{(C)} \) and \( Y \) was generated following the same procedure as for the training data sets with the exception that values for the three design variables were also generated at random within the vicinity of the geometric specifications as listed in Table IV. In doing so the different regression models could also be evaluated for their usefulness as models for the purpose of process design. The test data sets for both \( Y^{(C)} \) and \( Y \) contained 420 data points each and no noise was added to the efficiency criteria of the test data.

The training data were corrupted with additive noise. The noise term of each data point was calculated by the random expression \( \xi = Y^{(C)}(1 - Y^{(C)}) \times \eta \) where \( \xi \) is the stochastic or noise component and \( \eta \) is a random value within the range \((-0.1 \ldots 0.1) \). By using this sigmoidal gradient expression, realistic noise for the efficiency criteria could be emulated.

C. Different Regression Models

In order to investigate the efficiency of semi-empirical regression networks in the modeling of hydrocyclone performance, some existing regression techniques were employed to model the process for the purpose of comparison. The four main regression approaches which were used are:

(I) Sigmoidal backpropagation networks.

(II) Sigmoidal backpropagation networks with a single \([s-a]\) node in the hidden layer.

(III) Different semi-empirical regression network models.

(IV) Multiple adaptive regression splines (MARS) [4], [11].

The number of \([s-a]\) hidden nodes of the sigmoidal backpropagation networks was varied from 3 to 9 for the modeling of both \( Y^{(C)} \) and \( Y \).

As stated in the introductory section, it will be beneficial to include linear relations into a model if the system contains intrinsic linearities. Holcomb and Morari [12] modified the topology of a sigmoidal backpropagation network by including a single linear hidden node. They showed that this topological change allows sigmoidal backpropagation networks to recover linear performance while not sacrificing the ability of sigmoidal backpropagation networks to reproduce nonlinear mappings. The required number of free modeling parameters will be fewer if such a connectionist network is used for the modeling of a system which intrinsically behaves linearly. For this reason sigmoidal backpropagation networks containing a single \([s-a]\) hidden node were also evaluated as an option for modeling hydrocyclone performance. Four configurations with different numbers of \([s-a]\) nodes (1, 2, 3 and 4 \([s-a]\) hidden nodes respectively) were specified for the networks of both \( Y^{(C)} \) and \( Y \). It should be noted that this approach can also be viewed as a form of the regression network algorithm.

The performance of the regression network models for the hydrocyclone (modeling approach (III)) was also compared with the modeling performance of the MARS procedure of
are assumed to be valid. Using this mechanistic knowledge, particle entrainment can be described by network part (a). The expression for $R_f/r_f = R_f/100$ as described in (8) is empirical and case-specific. A regression network model for $Y$ is therefore defined to be nonparametric in the part where $r_f$ is determined intrinsically, thus ensuring it to be a more general model. WF is calculated externally with (9) as a known function of FPS and LPM, and is used as an input to an intrinsic "hidden layer". Two different "hidden" node types were used in network part (a). For architecture (a), either 2, 3 or 4 [s-a] nodes were specified, while two [l-p] nodes were used for architecture (b). For the fraction of the particles which enter the product underflow due to classification, the empirical correlation for $d_{50}$ described by (7) is a reliable measure for the estimation of $d_{50}$ values. Network part (b) represents the calculation of particle size $d_{50}$ and can accordingly be specified to be parametric. Note that an [e-a] node is used here.

Much still needs to be learned about the hydrocyclone process before a general and adequate expression for the corrected efficiency will be available. The corrected efficiency is therefore intrinsically determined in a nonparametric fashion as $y(C) = \phi(X)$ (refer to (10)), where the argument $X = d/d_{50}$ is a fixed relation represented with an [l-p] node with constant input weights of 1 and -1 as shown in the configuration of network part (c) in Fig. 9. For the semiempirical regression network models of both $y(C)$ and $Y$, only two intrinsic [s-a] "hidden nodes" were used in network part (c). This was found to be adequate for the purpose of this study.

The last part of the semiempirical regression network models for the practical efficiency $Y$ relates $Y$ to $r_f$ and $y(C)$ according to the known correlation of (11). The connectionist configuration for this relation is given in part (d) of Fig. 9 with all weight values kept fixed as indicated in the figure. A few regression network models for $Y(C)$ and $Y$ are now available which have been constructed on the basis of some mechanistic knowledge and which also provide for nonparametric modeling within ill-defined regions of the process.

The number of free model parameters for each of the different regression models of the modeling approaches (I), (II), and (III) is documented in Table V. It is notable that fewer model parameters were generally used in the semiempirical regression network models than in the models of the nonparametric modeling approaches (I) and (II).

**E. Training of the Regression Models**

The training data were scaled prior to training. For the sigmoidal backpropagation networks (modeling approach (I)) and their peers with a single [l-a] node (modeling approach (II)) the predictor variables as well as the response variables $Y(C)$ and $Y$ of the different data sets were scaled to be within the range $-3$ to $3$. Three principal reasons for this have been outlined by Van der Walt et al. [13]. Firstly the input node activities of the neural net should be bounded so that the weights between the inputs of the sigmoidal hidden nodes are not too large. If so, it will cause extremely slow training...
Table V: Number of Free Model Parameters of the Regression Models for $Y^{(C)}$ and $Y$, Using the Modeling Approaches (I), (II) and (III)

<table>
<thead>
<tr>
<th>Semi-empirical regression networks (III) (refer to Figure 6.3)</th>
<th>Sigmodial backpropagation networks (III)</th>
<th>Sigmodial backpropagation networks with a single (l-a) hidden node (III)</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of hidden nodes</td>
<td>number of free model parameters</td>
<td>number of hidden nodes</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>3</td>
<td>24</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>32</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>40</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>48</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>56</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>64</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>72</td>
<td>4</td>
</tr>
</tbody>
</table>

For the modeling approaches (III) and (IV), i.e., semi-empirical regression network models and MARS, all predictor variables were scaled linearly, except for $d$ which was scaled logarithmically. By investigating the memorization performance of each network model (i.e., the efficiency of a model in predicting the functional values of the data sets used to train the specific model) a decision could be made whether a model was adequately trained. If not, such a model was not considered further during comparative studies between the models of the different regression techniques. The number of unsuccessfully trained network models as trained by the different training sets for both $Y^{(C)}$ and $Y$ are listed in Table VI. It should be noted that a missing entry represents 100% success in training.

The success in training is only given for the sigmodial backpropagation networks with an [l-a] node and for the semi-empirical regression network models. All the sigmodal backpropagation network models containing three to nine hidden nodes were adequately trained. It can be noted from Table VI that the semi-empirical regression networks for the practical efficiency $Y$ were successful in about 30% cases to locate an acceptable solution during training. Although adequate models were located for this example, this again emphasizes the complexity of the LMS error surface for complex regression networks, as discussed previously.

F. Performance of the Different Regression Models

The performance results of the different regression models for the efficiency criteria of the hydrocyclone process are compared in Figs. 10–15. Note that the performance results of the unsuccessfully trained network models are not incorporated in
TABLE VI
NUMBER OF UNSUCCESSFUL TRAINING RUNS FOR EACH SPECIFIC MODEL CONFIGURATION OF MODELING APPROACHES (II) AND (III)

<table>
<thead>
<tr>
<th>Sigmodial backpropagation networks with single [l-a] node (II)</th>
<th>Semi-empirical regression networks (III)</th>
</tr>
</thead>
<tbody>
<tr>
<td>corrected efficiency Y*:</td>
<td></td>
</tr>
<tr>
<td>N = 3</td>
<td>N = 5</td>
</tr>
<tr>
<td>number of [l-a] nodes</td>
<td>number of [l-a] nodes</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

the results displayed in these figures. The normalized absolute error \( E_{\text{norm}} \) of (12) is used to evaluate the performance of the different hydrocyclone models.

\[
E_{\text{norm}} = \frac{1}{N \times n} \sum_{s=1}^{N} \sum_{a=1}^{n} \| y_s - \hat{y}_{s,a} \| \times 100\%.
\]  

In this expression, \( n \) is the number of data samples in the validation set and \( N \) is the number of models with the same configuration trained on the same data. The actual value for the response variable as prescribed by data sample \( s \) in the validation data set is represented by \( y_s \), while \( \hat{y}_{s,a} \) represents the predicted value at the coordinates of data sample \( s \) as estimated by model \( S \).

This evaluation criterion was chosen due to a uniform absolute sensitivity of the response variables \( Y^{(C)} \) and \( Y \) through the complete range \((0 \ldots 1)\). Single-validation sets for both \( Y^{(C)} \) and \( Y \) (each set containing 420 test data points) which were constructed as described earlier, were used to evaluate the generalization performance of the different models.

Because of the weak performance of the MARS-procedure in modeling the performance of the hydrocyclone process, this modeling approach (IV) is not compared graphically with the performance results of the models for the other three modeling approaches. The memorization and generalization performance of the different MARS models are however listed in Table VII.

It was found that the predicted error values which represent the memorization performance of the different MARS models are much higher than the memorization performance of the other regression modeling techniques. It can therefore be concluded that the MARS procedure at its best was unable to find an adequate model for both \( Y^{(C)} \) and \( Y \). Note that
additive MARS models with $mi = 1$ performed the worst for both $Y^{(G)}$ and $Y$. The memorization performance however improved with an increase in the multivariate interaction ($mi$) for modeling of $Y^{(G)}$. The generalization performance decreased with an increase in $mi$ from 2 to 6 which indicates limited interaction between the independent variables for $Y^{(G)}$. The MARS models for $Y$ displayed no significant differences between their memorization and generalization performance with $mi$-values of 2 and 6 respectively which shows that the degree of variable interaction is higher in the case of $Y$.

Figs. 10–12 display the generalization performance of the different models for the corrected efficiency $Y^{(G)}$ according to the modeling approaches (I), (II), and (III). Although it is not shown here, it was found that the sigmoidal backpropagation networks improved their memorization performance with an increase in the number of hidden nodes or free model parameters. Overfitting is partially responsible for this expected improvement, especially if trained with the 210 and 70 data point training sets. This can be concluded from the results of Figs. 11 and 12, where a gradual decrease in generalization performance was observed with an increase in the number of hidden nodes. This phenomenon corresponds to the bias-variance dilemma studied by Geman et al. [14]. It can further be observed from the results of Figs. 10–12 that the sigmoidal backpropagation networks containing a single [1-4] hidden node (approach (II)) were incapable of outperforming sigmoidal backpropagation networks. On the contrary the training process was only complicated by adding the single [1-4] node to the hidden layer. The conclusion can therefore be made that $Y^{(G)}$ is intrinsically nonlinear by nature.

It was further observed that the memorization performance of the sigmoidal backpropagation network improved significantly with a decrease in the number of training samples, if compared to the memorization performance of the semi-empirical regression networks. Although the corresponding training performance of the semi-empirical regression networks was not as good as that of the sigmoidal backpropaga-
tion networks, the semi-empirical regression networks outperformed the sigmoidal backpropagation networks if validated on the test data set. This promising result is especially clear in Fig. 12 for the regression models of $Y^{(C)}$ trained with 70 data points. Although the sigmoidal backpropagation networks generally converged into much lower LMS minima, the generalization performance of the semi-empirical regression networks was substantially better than that of the sigmoidal backpropagation networks.

The different regression models for the practical efficiency $Y$ are compared in Figs. 13–15. Also in this case the nonlinear nature of the hydrocyclone performance criterion $Y$ is validated by the performance of the sigmoidal backpropagation networks with a single $[l-a]$ hidden node, which gained no significant improvement over sigmoidal backpropagation networks. For this modeling problem the significance of improvement of the modeling performance of the semi-empirical regression networks again increased with a decrease in the population density of training data samples. As was the case for the results of Fig. 12, it can also be observed from Fig. 15 that the semi-empirical regression network models trained with low density data (70 data points) contained significantly better generalization properties as compared to the corresponding sigmoidal backpropagation networks, although the training processes of the semi-empirical regression networks terminated at significantly higher LMS error values.

The different architectures (a) and (b) for the semi-empirical regression network models performed equally well, so that either $[s-a]$ or $[l-p]$ nodes can be used for this application in the nonparametric “hidden layer” of network part (a) of Fig. 9.

These promising results show that the regression network can be used successfully as a model building tool for hydrocyclone classifiers where mechanistic knowledge of the process is available. Due to the a priori knowledge contained by such a semi-empirical regression network, the generalization properties of the model are significantly better than the generalization performance of the other regression models. This improvement becomes especially significant when the different regression models are trained with limited data.

VII. CONCLUSION

A general regression tool, i.e., the regression network, has been proposed in this paper. The general nonlayered configuration of this new connectionist network was described and the training procedure by means of backpropagation discussed. It was shown how the regression network can be specified to present almost any arbitrary function. This regression network can therefore be used as a straightforward parametric regressor, and also for nonparametric modeling if a system is inadequately understood. For the extreme case where the mechanism and physics of a process are known, the functional form of the mechanistic model can be specified to the regression network and the model parameters can be estimated with the network. From data sets of some systems it is possible to formulate an idea about some features of a system from tendencies within the data. If so, the known features can be used as basis for modeling by the regression network and the model can be extended and refined by adding more relations and parameters. If a process is completely ill-defined, i.e., nothing is known about such a process, the regression network can be applied in a completely nonparametric fashion, much like a sigmoidal backpropagation network or a radial basis function network.

Although the regression network can theoretically be used for various applications, the fundamental study presented here revealed that the training process of the regression network for some systems can become very complicated. A new objective function, the relative LMS criterion function, was proposed and displayed superiority over the conventional LMS criterion function, if the degree of interaction between some independent variables of the system to be modeled is high. LMS error surface investigations showed that the problems experienced during training are caused mainly by the presence of either mountainous regions in the error surface, and/or extended plateaus if functional relations that are intrinsically exponential by nature are to be fitted to the corresponding data. During investigations into the usefulness of some optimization routines other than conjugate-gradient optimization, i.e., alternating projection, tree annealing and leap frog optimization, it was observed that especially dynamic leap frog minimization holds promise as alternative optimization method for the regression network, as this optimization technique was able in some cases to lead the weight matrix of the regression network off a plateau. The applicability of each optimization routine was qualified and some general guidelines proposed for finding an optimal solution for the regression network of a system.

To illustrate how the regression network algorithm can be used to model systems which are fundamentally poorly understood, semi-empirical regression network models were developed for the classification efficiency of a hydrocyclone classifier, a typical metallurgical processing system. Both the corrected and practical efficiency curves $Y^{(C)}$ and $Y$ of this process were modeled on the basis of artificial data using four different regression techniques. The regression network algorithm was employed to construct connectionist models by specifying configurations which are purely nonparametric, as well as in the development of semi-empirical regression network models. For the latter regression model, available knowledge about the mechanism of a specific process was built into the connectionist structure of the regression network, while the ill-defined aspects of a specific system were provided for by use of nonparametric regions within the structure of a semi-empirical regression network for the relevant process.

It was found that MARS [4] was unable to emulate any of the relevant relations adequately. Although sigmoidal backpropagation networks performed relatively satisfactorily for the modeling of this system, which in fact exhibits an inherent sigmoidal type of behavior, semi-empirical regression network models outperformed the sigmoidal backpropagation network models for both $Y^{(C)}$ and $Y$. This superiority of the semi-empirical regression network models for $Y^{(C)}$ and $Y$ became even more prominent with a decrease in the population density of training data points. This is indicative of a remarkable improvement in extrapolative properties of the semi-empirical
regression network over other regression models investigated for this application.

Further research on the development and further application of the regression network should result in improved optimization of this new connectionist network topology. Research is also currently being conducted to employ the regression network algorithm for the purpose of topological optimization, i.e., the automated identification of mathematically extractable features during optimization without the need to specify such relations manually prior to training.

REFERENCES


Etienne Barnard received the B.Eng. degree in electronics from the University of Pretoria, South Africa, the degrees B.Sc. (Hons) and M.Sc. in physics from Witwatersrand University, South Africa, and the Ph.D. in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA.

He is currently a Research Assistant Professor in the Center for Spoken Language Understanding at the Oregon Graduate Institute, on leave of absence from the University of Pretoria, South Africa. His research interests include pattern recognition with application to speech and image understanding, neural networks, and computer-generated holography; he has acted as consultant to industry in all of these fields.

Jannie S. J. van Deventer received the Ph.D. in chemical engineering from the University of Stellenbosch, as well as the Ph.D. in business economics from the University of South Africa. He is currently Professor and Head of the Department of Metallurgical Engineering at the University of Stellenbosch, South Africa. He spent some time in Australia as a visiting fellow, as well as seven months in industry in Belgium. His research activities include mathematical modeling, artificial intelligence, adsorption, surface separation systems, and environmental engineering.
Comparison of Different Artificial Neural Nets for the Detection and Location of Gross Errors in Process Systems

Chris Aldrich* and Jannie S. J. van Deventer
Department of Chemical Engineering, University of Stellenbosch, Private Bag X5018, Stellenbosch, 7599, South Africa

The reliability of the data which characterize the behavior of a plant is critical to the effective monitoring and improvement of plant performance. It is thus essential that gross errors in these data, which can arise from measurement problems or inadequate mathematical models, are detected and eliminated before the performance of the plant is evaluated. Procedures for the detection of gross errors based on back propagation neural nets have recently been shown to be superior to those based on conventional statistical tests, especially where data processing is dependent on highly nonlinear models. The global detection of gross errors in process systems appears to be a relatively simple problem that can be accommodated with equal efficiency by back propagation, probabilistic, and learning vector quantization neural nets. The location of errors based on the constraint residuals of process systems, on the other hand, poses a more formidable problem that is not handled well by standard back propagation nets. For these types of problems other systems, such as learning vector quantization neural nets, that are significantly more efficient than back propagation neural nets are recommended.

Introduction

The collection and analysis of data from processes, whether they are operated in steady state or not, are an important means for evaluating the performance of a plant or an individual process unit. The reliability of these data is essential for the successful monitoring of the behavior of the plant. Since process data are generally subject to random noise, or even gross errors, owing to sensor biases, and inadequate mathematical models or significant changes in the environment (Hlavacek, 1977; Verneuil et al., 1992), these data have to be filtered prior to interpretation. This is usually accomplished by constrained nonlinear optimization procedures in conjunction with statistical techniques to detect and remove possible gross errors in the data. Unfortunately the theory on which these statistical techniques for the detection of gross errors are based is only valid for linear systems. As a consequence, nonlinear systems are often not accommodated satisfactorily by classical statistical methods.

Aldrich and Van Deventer (1993, 1994a), Aldrich et al. (1993), as well as Gupta and Narasimhan (1993) have recently shown that neural nets provide an alternative means for the detection of gross errors in process constraints. Gupta and Narasimhan (1993) considered a linear process system and used a sigmoidal back propagation neural net with a single hidden layer to detect the presence of gross errors based on the absolute values of the magnitudes of the normalized constraint residuals of the process system. Although use of the absolute values of the normalized constraint residuals is not ideal, since some information regarding the nature of the errors in the system is lost, these authors were not successful in training a neural net to detect errors based on the actual values of the constraint residuals. Aldrich and Van Deventer (1994a) used a similar approach based on the absolute values of the constraint residuals and showed that back propagation neural nets have a powerful ability to detect the presence of gross errors in process systems, regardless of the linearity of the constraints. Moreover, they also proposed an alternative approach to gross error detection with neural nets, based on the measurement residuals of the variables. This approach is more useful in the sense that direct location of gross errors in the variables is possible.

In this paper the application of other types of neural nets for the detection of gross errors on nonlinearly constrained systems is discussed and shown to be superior to back propagation neural nets in some respects.

Gross Error Detection

The process constraints or conservation equations of a typical chemical or physical process system are (Crowe, 1988, 1989)

$$A x = 0$$  \hspace{1cm} (1)

where \(x\) is the \((n \times 1)\) vector of true values of the state variables and \(A\) an \((m \times n)\) constraint matrix of full row rank \(m \geq n\). If

$$x' = x + e$$  \hspace{1cm} (2)

constitutes the \((n \times 1)\) vector of measurements of the true values \(x\), with \((n \times 1)\) error vector \(e\), and covariance matrix \(Q\), then the measured values of the process variables generally violate the process constraints

$$Ax' = c \neq 0$$  \hspace{1cm} (3)

or in terms of the true values and error components

$$A(x + e) = c$$  \hspace{1cm} (4)

and if the constraints are linear

$$A(x + e) = Ax + Ae = c$$

i.e.

$$Ae = c$$  \hspace{1cm} (5)
Under the null hypothesis that no systematic errors are present, \( c \) is a multivariate normal with a zero mean (Madron et al., 1977; Romagnoli and Stephanopoulos, 1981; Mah and Tanhame, 1982; Tanhame and Mah, 1985), i.e.,

\[
E(c) = E(Ae) = A E(e) = 0
\]

and a covariance matrix \( V = \text{cov}(c) \) of the constraint residuals (Crowe, 1988), where

\[
V = E(\text{cc}^T) = A E(e e^T) A^T = A \text{cov}(e) A^T
\]

The alternative hypothesis is that the expected value of \( c \) is not zero, i.e.,

\[
E(c) = b \neq 0
\]

which indicates the presence of an error with a bias of magnitude \( b \). These two hypotheses are subsequently evaluated and rejected or accepted on the basis of specific statistical criteria (e.g., a 95% or 99% confidence limit). Note that the data need not be reconciled in order to calculate \( c \) and its derived statistics.

**Artificial Neural Nets**

The literature on neural nets is vast and only a very brief overview is provided in this paper (Hecht-Nielsen, 1990; Lippmann, 1987, 1989; Rumelhart et al., 1986; Wasserman, 1989).

A neural net is a parallel distributed information processing structure, consisting of an arrangement of interconnected primitive processing elements. Each processing element can have an arbitrary number of input connections, but only one output connection (that can branch or fan out to form a multiple output connection). Processing elements or artificial neurons can have local memory and also possess transfer functions that can use or alter this memory, process input signals, and produce the output signals of the elements, as shown in Figure 1.

These processing elements are typically divided into disjoint subsets, called layers, in which the process units usually possess the same computational characteristics. The layers comprising a neural net can be categorized as either input, hidden, or output layers, to denote the way in which they interact with the information environment of the net. The typical structure of a back propagation neural net is shown in Figure 2.

**Back Propagation Neural Nets (BPNN).** Computation in back propagation neural nets is feedforward and synchronous, i.e., the states of the process units in lower levels or layers of the net are updated before units in layers higher up in the net. The activation rules of process units are typically of the form

\[
z_i(t+1) = g(u_i(t))
\]

where \( u_i(t) \) designates the potential of a process unit at time \( t \), i.e., the difference between the weighted sum of all the inputs to the unit and the unit bias

\[
u_i(t) = \sum_j w_{ij} z_j(t) - \theta_i
\]

The form of the transfer function \( g \) may vary, but could be a linear, step or sigmoidal transfer function, among others, with a domain typically much smaller than that of the potential of the process unit, such as \([-1,1]\) or \([-1;1]\), for example.

The training of back propagation neural nets such as the one shown in Figure 2 is an iterative process involving the changing of the weights of the net, typically by means of a gradient descent method, in order to minimize an error criterion \( \epsilon \)

\[
\epsilon = \frac{1}{2} \sum (d_j - z_j)^2
\]

on the difference between the desired \( (d_j) \) and the actual \( (z_j) \) outputs of the unit. To minimize the error,

\[
\frac{\partial \epsilon}{\partial w_{ij}} = z_j (1 - z_j) B_j
\]

is calculated, where \( B_j = z_j - d_j \) for output units and

\[
B = \sum j w_{ij} z_j (1 - z_j) B_j
\]

for hidden units. The value of \( B \) is thus propagated back through the net in order to calculate all derivatives \( \partial \epsilon / \partial w_{ij} \) (hence back propagation neural nets). Weights are updated by \( \Delta w_{ij} = -\delta z_j \).

This process is repeated until no further decrease in the error \( \epsilon \) is obtained and the net is then considered to have converged.

**Probabilistic Neural Nets (PNN).** Probabilistic neural nets are based on the use of Bayesian classification methods and provide as such a very powerful general framework for pattern classification problems. These nets use exemplars (input vectors with known classifications) to develop distribution functions which are used to estimate the likelihood of a feature vector belonging to a particular category. Moreover, the estimates can be modified by the incorporation of priori information. Suppose a classification probe consists of \( N \) different classes \( C_1, C_2, ..., C_N \), and that the data on which the classification process is based can be represented by a feature vector with \( M \) dimensions:

\[
x = [x_1, x_2, ..., x_M]^T
\]

If \( F(x) = [F_1(x), F_2(x), ..., F_N(x)]^T \) the set of probability density functions of the class populations and \( A = [a_1, a_2, ..., a_N]^T \) is the set of a priori probabilities that a feature vector belongs to a particular class, then the Bayes classifier compares the \( N \) values \( a_1 F_1(x), a_2 F_2(x), ..., a_N F_N(x) \) and determines the class with the highest value.
Before this decision rule (in which the multivariate class of probability density functions are evaluated, weighted and compared) can be implemented, the probability density functions have to be constructed. Parzen estimation is a nonparametric method of doing so, in which no assumption is made with regard to the nature of the distributions of these functions. These estimators are comprised of sets of smaller parametric functions, which are typically multivariate Gaussians. A small Gaussian curve is in effect determined for each exemplar, after which the curves are added together and smoothed. The Parzen estimators used in the probabilistic neural net are of the form

$$P_k(x) = \left(\frac{1}{2\pi} \cdot \frac{M}{N} \right) \cdot S_k^{-1} \cdot \exp \left(-\frac{(x - x_k)^2}{2\mu^2} \right)$$

for class \(k\), where \(x_k\) is the \(j\)th exemplar in class \(k\), and \(S_k\) is the number of training samples in category \(k\). The exponential constituents of eq 13 are known as Parzen kernels if all the inputs to the classifier have a unit norm, i.e., \(x^T x = 1\). These kernels can be conveniently expressed as

$$\exp \left(\frac{x^T x_k}{1/\mu^2} \right)$$

where the term \(x^T x_k\) is the dot product of the feature vector \(x\) to be classified, with a training vector \(x_k\). The transfer functions of the processing elements are thus of the form \(\exp(x - 1/\mu^2)\).

Smoothing is accomplished by the parameter \(\mu = \mu(S_k)\), where

$$\lim_{\mu(S_k) \to 0} \mu(S_k) = 0, \quad \text{as } S_k \to \infty$$

and

$$\lim_{\mu(S_k) \to \infty} S_k \mu(S_k) = \infty, \quad \text{as } S_k \to \infty$$

These conditions are satisfied by

$$\mu(S_k) = cS_k^{b}, \quad \text{where } 0 \leq b \leq 1$$

The probabilistic neural net consists of an input buffer, a normalizing layer which normalizes the input vectors, a pattern layer which sums the Parzen kernels for each class, and a classification layer which determines the most likely class to which an input belongs, as shown in Figure 3.

**Learning Vector Quantization Neural Net (LVQNN)**

The essential concept on which learning vector quantization neural nets (LVQNN) are based is that a set of vectors can be distributed across a space in such a way that their spatial distribution corresponds with the probability distribution of a set of training data. Learning vector quantization nets differ from back propagation nets in that they are not supervised during training and construct their own representations of categories among input data.

A learning vector quantization network contains an input layer, a Kohonen layer which performs the classification based on the previously learned features of the various classes, and an output layer as shown in Figure 4. The input layer is comprised of one process element for each feature or input parameter of the various classes, while the output layer contains one process element for each class. Although the number of classes is predefined, the categories assigned to these classes are not.

During training the Euclidean distance \((d_i)\) between a training exemplar \((x)\) and the weight vector \((w_i)\) of each process element is computed, i.e.

$$d_i = \text{norm}(w_i - x) = \sqrt{\frac{1}{m} \sum_{j=1}^{m} (w_{ij} - x_j)^2}$$

If this winning process element and the training vector share the same class, the winning process element is moved toward the training vector, otherwise it is moved away, or repulsed. That is,

$$w_p = w_p + \beta_i(x - w_p)$$

if the winning process element is in the correct class

$$w_p = w_p - \beta_i(x - w_p)$$

if the winning process element is not in the correct class.

As a consequence, the process elements assigned to a class migrate to a region associated with their class. In the classification mode, the distance of the input vector to each process element is determined and the vector is assigned to the class of the winning process element.

**Classification of Errors**

In order not to unduly complicate the classification of errors in constrained data, only two types of errors are considered, viz. random and gross errors. The errors can assume any arbitrary distribution, but for the purpose of this investigation these errors are considered to have the same (normal) distributions and differ only with respect to their expected values; i.e., for random errors \(E(e) = 0\) and for gross errors \(E(e) = b \neq 0\).

To use a neural net to detect the presence of these errors in the data, a set of exemplars is constructed. These sets of training exemplars generally consist of \(k\)
vectors \( V_k \), each of which can be partitioned into a subvector \( V_{k}^{IN} \) containing all the inputs to the net and a subvector \( V_{k}^{OUT} \) which contains the desired outputs associated with the corresponding inputs, i.e.

\[
V_k = [V_k^{IN}, V_k^{OUT}]_k
\]

The inputs to the net generally consist of some function of the constraint residuals of the system, while the outputs consist of some label to identify the type or class of error. That is

\[
v_i^{IN} = f(c_p) \quad (i = 1, 2, \ldots, N), \text{ where } f(c_p) = |c_p|/\sigma
\]

and

\[
v_i^{OUT} = 0, \quad \text{if } v_i^{IN} \text{ is associated with a random error}
\]

\[
v_i^{OUT} = 1, \quad \text{if } v_i^{IN} \text{ is associated with a biased error}
\]

Since these outputs are binary vectors, the output of continuous-valued neural nets (such as back propagation neural nets) have to be post-processed prior to interpretation. In this investigation continuous-valued outputs were rounded off to the nearest binary class indicator, i.e.

\[
\text{IF } |z_{oi} - d_{oi}| < \pi, \quad \text{THEN } z_{oi} = d_{oi}
\]

Slightly modified output vectors were used to train the probabilistic neural net (PNN) and the learning vector quantization (LVQ) net.

\[
v_i^{OUT} = [0,1], \quad \text{if } v_i^{IN} \text{ is associated with a random error (i = 1, 2, \ldots, N)}
\]

\[
v_i^{OUT} = [1,0], \quad \text{if } v_i^{IN} \text{ is associated with a biased error (i = 1, 2, \ldots, N)}
\]

These exemplars are constructed by artificially corrupting a consistent set or sets of values to generate constraint residuals and labeling the residuals accordingly. In actual process systems it is often desirable to detect as high a percentage of gross errors as possible, provided that the misclassification of random errors is kept to a minimum. Although the same restrictions can be implemented readily in neural net detectors through appropriate labeling of the exemplars, no such measures were used in this investigation. In the rest of this paper the ability of a neural net to classify errors denotes its ability to classify both gross and random errors.

**Examples**

**Example 1: Detection of Gross Errors in a Gravity Separator Subject to a Nonlinear Constraint.** Consider a gravity separator designed to separate dense valuable material from less dense waste. The following constraint describes the interrelationship between variables governing the operation of the separator:

\[
x_1 x_1 = \arctan(x_2^{1/2} + 3x_3 x_4) = 1.0129 = 0 \quad (21)
\]

where

\[
x_1 = \text{feed rate/maximum feed rate}
\]

\[
x_2 = \text{slope of streaming current acting as separating medium}
\]

\[
x_3 = \text{fraction of valuable material in concentrate product}
\]

\[
x_4 = \text{(density of waste - density of separating medium)/(density of valuable dense material - density of separating medium)}
\]

Besides the random errors that could occur in the measurement of these variables, it is evident that faulty flowmeters, on-line analyzers, density gauges, and slope indicators could give serious gross errors. This simple, but highly nonlinear system will serve to illustrate the ability of the various neural nets to detect gross errors in constrained sets of variables. Owing to its nonlinearity the above system is not amenable to classical statistical methods of gross error detection. Consistent sets of variables \( x_1, x_2, \ldots, x_4 \) (with \( 0 < x_i < 1 \)) were corrupted with uniformly random errors with a standard deviation of 5% and gross errors with biases of up to 100%. From these sets training and test sets each containing 200 vectors were constructed (different sets of consistent variable values were used to construct test and training exemplars).

The first group of training and test sets was based on exemplars of the form \( [c_P^{CLASS}] \), while the second group was based on exemplars of the form \( [c_P/\sigma^{CLASS}] \). The latter group of exemplar sets allowed a more sophisticated approach to gross error detection in that information on the direction (sign) of the bias in the gross error is retained. In this way more accurate estimates of the ability of the nets to generalize the classification of errors are realized. In all cases the test and training vector sets contained approximately equal proportions of gross and random errors (i.e., the probability of a given exemplar containing a gross error was approximately 50%). This means that a classifier with no ability to classify errors (i.e., a completely random assignment of errors to the various classes) would on average achieve a success rate in the region of 50%.

**Back Propagation Neural Net.** A two-layer hyperbolic tangent back propagation neural net with a single input node, a three-node hidden layer, and a single output node was trained with the generalized delta rule (Hinton, 1989) on exemplars of the form \( [c_P/\sigma^{CLASS}] \) and \( [c_P/\sigma^{CLASS}] \). After training, the net was evaluated against test sets containing errors with biases similar to those on which the net had been trained. The results of a typical number of runs are displayed in Figures 5 and 6. Figure 5 depicts the rates of successful classification of random and gross errors based on feature vectors of the type \( [c_P/\sigma] \), while Figure 6 depicts similar results based on feature vectors of the type \( [c_P/\sigma^{CLASS}] \). From both these figures it is apparent that the ability of the nets to detect gross errors increases with an increase in the bias of the gross errors and approximately 80–90% of these errors could be detected when the biases in the gross errors exceeded 70% (as is shown especially in Figure 6). When feature vectors of the type \( [c_P/\sigma^{CLASS}] \) were used, the performance of the back propagation net tended to be erratic, as can be seen from the scatter in the data points in Figure 5. The same-
what inconsistent behavior of the back propagation neural net (shown in Figure 5) is not fundamentally significant, since the decision boundaries which the net has to cope with can in principle (and in practice) be accommodated by more sophisticated back propagation techniques (not attempted in this investigation).

More consistent performance was observed when the absolute values of the normalized constraint residuals were used to classify the errors, rather than the normalized constraint residuals as such (with the loss of information previously mentioned).

Although hyperbolic tangent transfer functions were used, the performance of the net did not appear to be sensitive to the use of other popular transfer functions such as sigmoidal, sine, or even linear functions, especially as far as classification based on feature vectors of the form \(|c, \theta|\) was concerned. Similar results were also obtained with a simple single layer neural net trained on these types of features (but not features of the form \(|c, \theta|\)).

**Learning Vector Quantization Neural Net.** The learning vector quantization net consisted of an input layer with a single node (corresponding to the single constraint of the system), a Kohonen layer with four nodes, and an output layer with two nodes (one for each type of error). Of the three nets investigated, the learning vector quantization net was the only one whose performance did not appear to be affected by the form of the feature vectors on which classification was based (as can be seen from Figures 5 and 6 the learning vector quantization net exhibited essentially the same capacity to detect gross errors, regardless of the form of the feature vectors).

**Probabilistic Neural Net.** The probabilistic neural net was comprised of an input layer with a single input node, a normalizing layer with a single node, a pattern layer consisting of 200 nodes (one for each exemplar or pattern in the training set), a summation layer with two nodes, and an output layer with two nodes (one for each error class).

The probabilistic neural net trained approximately 1–2 orders of magnitude faster than the back propagation or learning vector quantization nets, and yielded essentially identical results when presented with features of the form \(|c, \theta|\). When presented with feature vectors of the form \(|c, \theta|\), however, the probabilistic neural net was also prone to inconsistency as is indicated by Figure 5.

**Example 2a: Detection of Gross Errors in a System Subject to Multiple Nonlinear Constraints.** This example has been used previously by Pai and Fisher (1988), Tjoa and Bingler (1991), as well as Aldridge and Van Deventer (1994b) (albeit for the evaluation of data reconciliation algorithms and not for the detection of gross errors) and comprises five measured variables \(x_i\) (\(i = 1, 2, \ldots, 5\)) and three unmeasured variables \(x_i\) (\(i = 6, 7, 8\)), subject to six nonlinear constraints.

\[
\begin{align*}
1/2(x_1)^2 - 0.7x_2 + x_3x_6 + (x_2)^2x_7 + 2x_3x_6^2 - 2.55.8 &= 0 \quad (2') \\
1 - 2x_2 + 3x_3x_5 - 2x_4x_6 - x_5x_7x_8 + 111.2 &= 0 \quad (2') \\
x_3x_6 - x_4 + 3x_2 + x_7x_8 - x_4(x_8)^{1/2} - 33.57 &= 0 \quad (2') \\
x_4 - x_1 - (x_3)^2 + x_7 + 3x_8 &= 0 \quad (2') \\
x_8 &= 2x_3x_7x_8 \quad (2') \\
2x_1 + x_3x_7x_8 - x_1 - x_8 &= -126.6 = 0 \quad (2)
\end{align*}
\]

Although this example does not relate to any specific practical system, it was selected due to its high nonlinear nature and the fact that it has been studied before. Therefore, it provides an ideal basis for comparison of different network architectures. As a comparison of a back propagation neural net is concerned not more than six output nodes are required to monitor errors in each of the six subsets of variables, regardless of the number of gross errors present. Since classes are defined by numeric labels, any number of classes (e.g., different types of gross errors) can moreover be accommodated by this configuration.

The use of probabilistic or learning vector quantization neural nets is not quite as convenient, especially as far as the detection of multiple gross errors concerned, since classes are not defined by numeric labels as such, but by binary vectors. This means that the number of output nodes or classes is dependent on the number of possible combinations of gross errors. General \(E^N\) classes have to be defined, where \(E\) is the number of error types and \(N\) the number of system constraints). For example, considering multiple gross errors in only two constraint residuals would require four process nodes when the probabilistic or learning vector quantization nets are used. The vector \((1, 0, 1)\) could indicate the absence of any gross errors, \((0, 1, 1)\) the presence of a gross error in one of the vari
associated with the first constraint residual, \([0,0,1,0]\) a gross error associated with one of the variables in the second constraint residual, and \([0,0,0,1]\) gross errors associated with both constraint residuals. This problem, which can constitute a significant computational burden in large process systems, can of course be circumvented by decomposing the system into a series of smaller subsystems or even single constraints if necessary. A separate neural net could then be trained for each of these subsystems (the performance of the decomposed system can be expected to be the same as that of a single large multivariable system).

In order not to complicate the analysis unnecessarily, only single gross errors are consequently considered. (Multiple gross errors can be treated in the same way, by expanding the number of classes appropriately to accommodate the different possibilities concerning the location of various errors.)

To evaluate the neural nets, test and training data sets were compiled by corrupting a consistent set of values \(x = [4.5124, 5.5819, 1.9260, 1.4560, 4.8545, 11.070, 0.61647, 2.0504]\) with random errors with a standard deviation of 10% and gross errors of various biases. In contrast to the previous example, these errors had a Gaussian (and not a uniform) distribution.

The ability of the neural nets to classify the errors based on feature vectors of the form \([\epsilon_j/\sigma; \text{CLASS}]\) is depicted in Figure 7. This figure shows that the three neural nets performed essentially the same, although results obtained with the probabilistic neural net appeared to be less consistent than those obtained with the back propagation and learning vector quantization neural nets.

**Example 2b: Location of Gross Errors.** It is usually necessary not only to detect the global presence of gross errors in process systems, but also to identify the sources of these errors in order to eliminate or compensate for them. This poses a more formidable problem than the detection of gross errors as such, since the classifier cannot directly relate the presence of a gross error to the magnitude of a particular constraint residual, but has to interpret a pattern of constraint residual values.

Considering the system represented by eqs 22–27 once again, there are eight variables (possible sources of gross errors) and six constraints which serve as an indication of the presence of errors in the system. To locate these errors, it is necessary to evaluate the constraints (as was demonstrated in the first part of this example) and then to combine these results in order to isolate gross errors as far as possible. This can be accomplished by training a neural net with the normalized constraint residuals as input, but instead of a single measure (or at least one such measure per constraint) to indicate the presence or absence of a gross error, the net was provided with the locations of the variables containing the gross errors, i.e., with feature vectors of the form \([\epsilon_j/\sigma; \text{LOCATION}]\).

The variable LOCATION could assume integral values from 0 to 8, with 0 indicating the absence of gross errors, and \(i\) an error in variable \(x_i\), with \(1 \leq i \leq 8\). When the probabilistic neural net and the learning vector quantization neural nets were used, the LOCATION variable was modified to nine-digit binary vectors; that is, \([1, 0, 0, ..., 0]\) indicated the absence of any gross errors, \([0, 1, 0, 0, ..., 0]\) indicated the presence of a gross error in variable \(x_1\), \([0, 0, 1, 0, 0, ..., 0]\) indicated the presence of a gross error in variable \(x_2\), etc.

The results of these experiments are shown in Figure 8. The back propagation neural net largely failed to converge during training and performed generally very poorly as far as the location of gross errors was concerned. This is in sharp contrast to the learning vector quantizer and probabilistic neural nets, which performed considerably better the back propagation net.

**Example 3: Location of Gross Errors in a Flotation Circuit Subject to Linear Constraints.** A similar approach was followed with a flotation circuit previously described in the literature (Cutting, 1976). The circuit consists of 12 process units, viz. 6 flotation banks, 5 hydrocyclones and sumps, and 1 mill. Since only total flow rates are considered, the effect of the mill can be ignored, yielding a system represented by 19 process variables subject to 11 linear process constraints:

Flotation banks

\[
F_1 - F_2 - F_3 = 0 \tag{28}
\]

\[
F_4 - F_5 - F_6 = 0 \tag{29}
\]

\[
F_9 - F_{10} - F_{11} = 0 \tag{30}
\]
Hydrocyclones and sumps

\[
\begin{align*}
F_{12} - F_{13} - F_{14} &= 0 \\
F_{15} - F_{16} - F_{17} &= 0 \\
F_{16} - F_{18} - F_{19} &= 0
\end{align*}
\]

As in example 1, measurement devices such as magnetic flowmeters, density gauges, and pressure gauges used to determine the \( F_j \) mass flow rates are notorious for being defective in mineral processing plants, so that gross errors are quite common. Consistent measurements of these \( F_j \) variables were corrupted with uniformly random (standard deviation of 5%) and gross errors with various biases, while sets of training and test exemplars were compiled as discussed previously. The back propagation net was trained and tested on sets of exemplars which consisted of the 11 process constraints as input, as well as the number of each variable as output, that is

\[
V_k = [c_1, c_2, ..., c_{11}, \text{LOCATION}]_k
\]

with \( \text{LOCATION} = 0 \) in the absence of gross errors, and \( \text{LOCATION} = j \) with a gross error present in variable \( F_j \).

The learning vector quantizer and probabilistic neural nets were trained and tested with exemplars of similar form, but with binary \( \text{LOCATION} \) variables, with the position of the 1 in the output vector an indication of the variable containing a gross error. (As before an extra element is added to the binary vector to indicate the absence of gross errors; i.e., \([1, 0, 0, ..., 0]\) indicates the absence of gross errors, \([0, 1, 0, 0, ..., 0]\) indicates a gross error in variable 1, \([0, 0, 1, 0, 0, ..., 0]\) indicates a gross error in variable 2, etc.)

The results of these tests are shown in Figure 9. As can be seen from this figure, the learning vector quantizer performed significantly better than the probabilistic neural net, or the back propagation neural net which completely failed to converge, despite numerous modifications and trial and error approaches. This observation can probably be ascribed to the fact that the back propagation net had to contend with a complicated error surface not easily navigated by gradient descent methods.

**Example 4: Partial Location of Gross Errors in a Two-Product Hydrocyclone Classifier.** Consider a two-product hydrocyclone classifier in which the feed stream \( W_1 \) is classified into two product streams, i.e., a coarse underflow stream \( W_2 \) and a fines overflow stream \( W_3 \). The mass flow rates of \( W_1, W_2, \) and \( W_3 \) are determined by magnetic flowmeters, and could be based on faulty readings. Similarly, nuclear density gauges are used to determine the specific gravity and hence the percentage of solids \( x_1 \) in each stream. Modern on-line particle size analyzers could be used to determine the mass of each size fraction. In practice it is possible that systematic or gross errors could occur in the measurement of \( x_1 \) and \( x_2 \) owing to defective instruments. All variables are measured and have to satisfy the following mass conservation constraints

\[
\begin{align*}
&\omega_1 - \omega_2 - \omega_3 = 0 \\
&\omega_1 x_{11} - \omega_2 x_{21} - \omega_3 x_{31} = 0 \\
&\omega_1 x_{12} - \omega_2 x_{22} - \omega_3 x_{32} = 0
\end{align*}
\]

As before, consistent values of the variables were corrupted with random Gaussian errors with a standard deviation of 10%, as well as with gross errors which had, except for their biases, the same distributions as the random errors. Sets of exemplars consisting of the three constraint residuals of the system, as well as indicators of the approximate location of gross errors (if present) were generated, i.e.

\[
V_k = [c_1, c_2, c_3, \text{LOCATION}]_k
\]

where for the back propagation neural net \( \text{LOCATION} \) assumed a value of 0 if no gross errors were present, a value of 1 if the gross error was present in any of variables \( W_1, W_2, \) or \( W_3, \) a value of 2 if the gross error was present among variables \( x_{11}, x_{21}, \) or \( x_{31}, \) and a value of 3 if the gross error was present among variables \( x_{12}, x_{22}, \) or \( x_{32}, \) As before, the feature vector was also expressed in binary form for use with the probabilistic and learning vector quantization neural nets.

Each training and test set consisted of exemplars which contained the four classes of errors as discussed above in roughly equal proportions. The ability of the neural nets to locate errors in the system is depicted in Figure 10. As shown in Figure 10, the learning vector quantization net performed better than both the probabilistic and back propagation neural nets.

**Discussion and Conclusions**

The advantage of using neural nets to classify errors in measurement data subject to constraints is based on the fact that unlike maximum likelihood techniques and other statistical methods, for example, no assumptions with regard to the underlying error distributions are made. Neural nets are known to work well when the inputs are generated by nonlinear processes (Lippmann, 1987) and are thus ideally suited for the classification.
of errors with unknown or strongly non-Gaussian distributions.

As was shown in examples 1 and 2 and also previously by Aldrich and Van Deventer (1993, 1994a), the global detection of gross errors based on the constraint residuals of the system can be accommodated successfully by any of the three neural net classifiers considered in this investigation. Similar studies by the authors indicate that other neural net systems such as radial basis and general regression neural nets work just as well, and as far as the ability to detect gross errors is concerned, the choice of a particular classifier is largely arbitrary.

In contrast, the location of gross errors in a process system constitutes a more complicated problem. In this case neural nets have to interpret complex patterns of constraint residuals which are not only dependent on the actual values of the process variables at the time of the location of gross errors in the system, but also on the actual values of the process variables at the time of sampling, especially in nonlinear systems.

As can consequently be expected (and as was demonstrated in examples 2–4), not all neural net systems are equally appropriate for the identification of sources of gross errors in process systems, at least not when based on the constraint residual patterns generated by the system. In this investigation learning vector quantization neural nets have performed marginally better than probabilistic neural nets and significantly better than standard back propagation neural nets as far as the location of gross errors was concerned. To summarize,

1. The global detection of gross errors in process systems subject to constraints of an arbitrary nature can be accommodated equally well by back propagation, probabilistic, and learning vector quantization neural nets.

2. Owing to the simpler form of the feature vectors required, back propagation neural nets are especially convenient to use for the global detection of errors.

3. In contrast, standard back propagation neural nets do not appear to be efficient as far as the location of errors in complex circuits is concerned, and in this regard much better use can be made of especially learning vector quantization neural nets.

Nomenclature

\[ A = \text{matrix of process constraint coefficients} \]
\[ A = \text{set of } a \text{ priori probabilities } a_i \]
\[ a_i = \text{ith a priori probability of a feature vector belonging to a certain class} \]
\[ B_j = \text{error passed back through a back propagation neural net in order to optimize the weight matrix of the net} \]
\[ b = \text{constant} \]
\[ b = \text{vector of error bias} \]
\[ c_p = \text{pth constraint residual in a process system} \]
\[ C_i = \text{ith class to which an exemplar can be assigned} \]
\[ c = \text{constant} \]
\[ c = \text{vector of process constraint residuals} \]
\[ c_{ij} = \text{covariance} \]
\[ d_{ij} = \text{desired output value of process unit } j \]
\[ d_i = \text{Euclidean distance between a training exemplar and the weight vector of a learning vector quantizer neural net} \]
\[ E = \text{number of error categories} \]
\[ E(x) = \text{expected value} \]
\[ e = \text{measurement error vector of process variables} \]
\[ F_j = \text{flow rate} \]
\[ F(x) = \text{set of probability density functions} \]
\[ F_i(x) = \text{ith probability density function} \]
\[ g(\cdot) = \text{transfer function of neural net process element} \]
\[ P_k(x) = k\text{th Parzen estimator} \]
\[ Q = \text{covariance matrix of process variable measurements} \]
\[ x = \text{vector of process variables} \]
\[ N = \text{number of system constraints} \]
\[ S_i = \text{number of training exemplars in category } k \]
\[ t = \text{time} \]
\[ u_i(t) = \text{potential of process unit } i \text{ in neural net at time } t \]
\[ V = \text{covariance matrix of constraint residuals} \]
\[ V^\text{IN} = \text{component } j \text{ of vector } V_j \]

\[ V^\text{OUT} = \text{component } j \text{ of vector } V \]
\[ V = k\text{th training vector} \]
\[ V^\text{OUT} = \text{component or subvector of training vector containing desired output values} \]
\[ W_i = \text{flow rate of process stream } j \]
\[ w_{ij} = \text{connection strength or weight between process units } i \text{ and } j \text{ in neural net} \]
\[ \Delta w_{ij} = \text{change in connection strength or weight between process units } i \text{ and } j \text{ in neural net} \]
\[ x = \text{measured values of process variables} \]
\[ x = \text{true values of process variables} \]
\[ z_{ij} = \text{mass fraction of component } j \text{ in flow stream } i \]
\[ z(t) = \text{state of process unit } i \text{ in neural net at time } t \]
\[ z_{ij} = \text{actual output of process unit } j \]

Greek Letters

\[ \beta_1, \beta_2 = \text{constants} \]
\[ \alpha = \text{step size} \]
\[ \epsilon = \text{error criterion for adjustment of weight matrix of neural net} \]
\[ \Theta_i = \text{bias of process unit } i \text{ in neural net} \]
\[ \mu = \text{smoothing parameter} \]
\[ \pi = \text{arbitrary constant} \]
\[ \sigma = \text{standard deviation of process constraint } c_p \]

Literature Cited


Received for review January 28, 1994
Accepted September 16, 1994

Chapter 13

Estimation of measurement error variances and process data reconciliation

C. Aldrich and J. S. J. van Deventer

Department of Chemical Engineering, University of Stellenbosch, Private Bag X5018, Stellenbosch 7599, South Africa, Fax 27 (21) 8082059, E-mail: cal@maths.sun.ac.za

1. Introduction

The monitoring of plant performance and the verification of system models are crucially dependent on reliable sets of steady state component and total flow rate data. In general the measured data violate the process constraints of the system, owing to random fluctuations in the observed values, or even systematic errors in these values due to erroneous measurements or large discrepancies between the actual behaviour of the system and the behaviour predicted by the system model. The observations or measurements consequently have to be reconciled with the process constraints, usually through the minimization of a quadratic criterion. This criterion is typically a function of the differences between the measured and the adjusted values, weighted by the inverse of the measurement error covariance matrix [1]. General reconciliation methods are usually based on the assumption that measurement errors are randomly Gaussian with a known covariance matrix, and distributed around a zero mean. In many practical situations, this matrix is unknown and has to be estimated [2,3].

Moreover, it is essential that gross errors are detected and accounted for prior to, or during reconciliation of the data, since failure to do so could result in a severely distorted picture of the process [4]. Since repeated measurement of a variable is not an effective means for the detection of a systematic error, virtually all gross error detection schemes involve statistical tests based on the characteristics of the constraint residuals of the measurement errors. Unfortunately these tests are generally only useful as far as systems subject to linear constraints are concerned.
In the chemical and mineral processing industries this is a major drawback, since most process systems in these industries are non-linear.

Once the variances of the process variables have been estimated, and gross errors have been eliminated, the variable measurements can be reconciled with the system constraints. The usual approach to the reconciliation of measured variables is aimed at the minimization of the weighted sum of the squares of the measurement residuals, subject to conservation and other constraints of the process [1], i.e.

\[
\min (x' - x'\prime \prime)^T V^{-1} (x' - x'\prime \prime)
\]

subject to

\[
d(x'\prime \prime) = 0
\]

where \(V^{-1}\) is a residual weighting matrix, usually the inverse of the variance-covariance matrix of the measurements \(x'\), while \(x'\prime \prime\) is the vector of reconciled measurements. The set of multivariate functions \(d(x'\prime \prime)\) represents the constraints of the system. Where an estimate of this variance-covariance matrix is not available, a numerical weighting system can also be used by defining \(V^{-1}\) as the inverse of the elements of the measurements of the variables \(\{x\}\). More formally, the problem can be considered as follows, for a process described by a system of linear equations

\[
C.x = 0
\]

where \(C\) is a \((n \times p)\) coefficient matrix with \(n < p\), representing the system constraints, and \(x\) is the true value of the state vector. The observed or measured value \(x'\), of this vector

\[
x' = x + e
\]

is typically subject to an error \(e\), so that the constraint residuals \(r\) are related to the measurement vector \(x'\) as follows

\[
r = C.x' = C.(x + e)
\]

where

\[
C.(x + e) = C.x + C.e = C.e
\]

If \(e\) is considered to be a Gaussian variable with a zero mean and a covariance matrix \(V_{\text{ee}}\), then

**Measured variables**

\[
E(e) = 0
\]

and

\[
\text{var}(e) = E(ee^T) = V_{\text{ee}}
\]

**Constraint residuals**

\[
E(r) = E(C.e)
\]
and
\[ \text{var}(r) = E[(C.e)(C.e)^T] = V_r \] (10)

Since the system is linear
\[ E(r) = C.E(e) = 0 \ldots \text{ (random errors only)} \] (11)
\[ E(r) = b \neq 0 \ldots \text{ (gross errors or biases present)} \] (12)
\[ \text{var}(r) = C.E(ee^T).C^T = CV_rC^T = V_r \] (13)

If the system is non-linear, such as would arise from conservation equations involving the products of stream flows \( F \) and component fractions \( f \), as well as other non-linear relationships which may ensue from different process circuit configurations, the relationship expressed by the above equations is no longer valid and cannot be used in traditional analytical procedures to obtain estimates of the statistical properties of the measured variables.

This ill-defined relationship between the constraint and measurement residuals thus restricts the use of analytical methods to linear systems, since in non-linear systems the distributions of the constraint residuals are generally unknown. This complication can be avoided by making use of artificial neural networks to represent the relationship between the properties of the constraint residuals and those of the measurement residuals. This relationship can then be used as a basis for the estimation of measurement error variances or gross error detection schemes.

The use of neural networks for variance estimation, gross error detection and the reconciliation of process data is considered below, by way of examples.

2. Estimation of Variances and Covariances in Quasi-Steady State Systems

In order to demonstrate the use of a neural network to estimate measurement error variances, a quasi-steady state system is considered. The system considered is subject to periodic fluctuations in which the expected values of the process variables are continuously changing, without a significant change in the variances of these variables. Under these circumstances direct estimates of variances can be highly inaccurate, and indirect methods which are based on the variances of the constraint residuals are thus more appropriate [7,8]. Note that the expected values of the constraint residuals remain zero, regardless of the expected values of the process variables. By presenting an artificial neural network with exemplars of the relationship between the covariances of the constraint residuals and that of the measurement errors, the network forms an internal representation of this relationship which can be used to provide an estimate of the covariance matrix of the measurement errors. These exemplars would necessarily contain additional implicit assumptions concerning the nature of the variances of the measurement residuals, in order to uniquely define the relationship between the constraint and measurement variances.
Table 1. Expected values of process variables.

<table>
<thead>
<tr>
<th>TIME</th>
<th>$F_1$</th>
<th>$F_2$</th>
<th>$F_3$</th>
<th>$f_{1,1}$</th>
<th>$f_{1,2}$</th>
<th>$f_{2,1}$</th>
<th>$f_{2,2}$</th>
<th>$f_{3,1}$</th>
<th>$f_{3,2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_0$</td>
<td>1.000</td>
<td>0.350</td>
<td>0.650</td>
<td>0.260</td>
<td>0.740</td>
<td>0.297</td>
<td>0.703</td>
<td>0.240</td>
<td>0.760</td>
</tr>
<tr>
<td>$t_1$</td>
<td>1.100</td>
<td>0.370</td>
<td>0.730</td>
<td>0.280</td>
<td>0.720</td>
<td>0.398</td>
<td>0.602</td>
<td>0.220</td>
<td>0.780</td>
</tr>
<tr>
<td>$t_2$</td>
<td>1.050</td>
<td>0.380</td>
<td>0.670</td>
<td>0.290</td>
<td>0.710</td>
<td>0.431</td>
<td>0.569</td>
<td>0.210</td>
<td>0.790</td>
</tr>
<tr>
<td>$t_3$</td>
<td>1.150</td>
<td>0.360</td>
<td>0.790</td>
<td>0.250</td>
<td>0.750</td>
<td>0.360</td>
<td>0.640</td>
<td>0.200</td>
<td>0.800</td>
</tr>
<tr>
<td>$t_4$</td>
<td>1.170</td>
<td>0.390</td>
<td>0.780</td>
<td>0.240</td>
<td>0.760</td>
<td>0.300</td>
<td>0.700</td>
<td>0.210</td>
<td>0.790</td>
</tr>
<tr>
<td>$t_5$</td>
<td>1.220</td>
<td>0.410</td>
<td>0.810</td>
<td>0.270</td>
<td>0.730</td>
<td>0.448</td>
<td>0.552</td>
<td>0.180</td>
<td>0.820</td>
</tr>
<tr>
<td>$t_6$</td>
<td>1.180</td>
<td>0.400</td>
<td>0.780</td>
<td>0.290</td>
<td>0.710</td>
<td>0.524</td>
<td>0.476</td>
<td>0.170</td>
<td>0.830</td>
</tr>
<tr>
<td>$t_7$</td>
<td>1.210</td>
<td>0.420</td>
<td>0.790</td>
<td>0.310</td>
<td>0.690</td>
<td>0.555</td>
<td>0.445</td>
<td>0.180</td>
<td>0.820</td>
</tr>
<tr>
<td>$t_8$</td>
<td>1.270</td>
<td>0.405</td>
<td>0.865</td>
<td>0.340</td>
<td>0.660</td>
<td>0.746</td>
<td>0.254</td>
<td>0.150</td>
<td>0.850</td>
</tr>
<tr>
<td>$t_9$</td>
<td>1.280</td>
<td>0.440</td>
<td>0.840</td>
<td>0.350</td>
<td>0.650</td>
<td>0.770</td>
<td>0.230</td>
<td>0.130</td>
<td>0.870</td>
</tr>
<tr>
<td>$t_{10}$</td>
<td>1.300</td>
<td>0.430</td>
<td>0.870</td>
<td>0.360</td>
<td>0.640</td>
<td>0.805</td>
<td>0.195</td>
<td>0.140</td>
<td>0.860</td>
</tr>
</tbody>
</table>

The general structure of the network is determined by the number of process variables and constraints of the system. If the system is generally described by $N$ system variables and $M$ process constraints, the corresponding network has an input layer consisting of $M^2$ process nodes (with each node corresponding to an element in the variance-covariance matrix of the constraint residuals), and an output layer consisting of $N^2$ process nodes (with each node corresponding to an element in the variance-covariance matrix of the measurement residuals). As mentioned, this general structure of the neural network can be modified to account for the incorporation of additional information in the system. If the covariances of the state variables are deemed to be negligible for example, the circuit structure can be reduced to accommodate the diagonal of the covariance matrix of the measurement errors only. Besides alteration of the structure of the network to reflect additional knowledge of the process, such knowledge can also be accommodated in the set of exemplars used to train the network. The use of neural networks such as these is demonstrated in the following example.

2.1. Example 1: Two-product separator subject to non-linear process constraints

In this example a high tension roll separator is considered. The separator classifies a feed stream $F_1$, consisting of two components with mass fractions $f_{1,1}$ and $f_{1,2}$, into two product streams $F_2$ with component mass fractions $f_{2,1}$ and $f_{2,2}$, as well as $F_3$, with component mass fractions $f_{3,1}$ and $f_{3,2}$. The flow streams and mass fractions are measured and typically violate the conservation equations of the system, viz.

$$F_1 - F_2 - F_3 = 0$$ (14)
Table 2. Measurements of process variables

<table>
<thead>
<tr>
<th>TIME</th>
<th>( F'_1 )</th>
<th>( F'_2 )</th>
<th>( F'_3 )</th>
<th>( f'_{1,1} )</th>
<th>( f'_{1,2} )</th>
<th>( f'_{2,1} )</th>
<th>( f'_{2,2} )</th>
<th>( f'_{3,1} )</th>
<th>( f'_{3,2} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_0 )</td>
<td>0.988</td>
<td>0.361</td>
<td>0.548</td>
<td>0.232</td>
<td>0.599</td>
<td>0.249</td>
<td>0.747</td>
<td>0.217</td>
<td>0.734</td>
</tr>
<tr>
<td>( t_1 )</td>
<td>0.944</td>
<td>0.346</td>
<td>0.690</td>
<td>0.235</td>
<td>0.745</td>
<td>0.352</td>
<td>0.672</td>
<td>0.195</td>
<td>0.816</td>
</tr>
<tr>
<td>( t_2 )</td>
<td>1.162</td>
<td>0.374</td>
<td>0.775</td>
<td>0.238</td>
<td>0.581</td>
<td>0.473</td>
<td>0.455</td>
<td>0.191</td>
<td>0.783</td>
</tr>
<tr>
<td>( t_3 )</td>
<td>1.272</td>
<td>0.382</td>
<td>0.852</td>
<td>0.257</td>
<td>0.631</td>
<td>0.316</td>
<td>0.732</td>
<td>0.168</td>
<td>0.855</td>
</tr>
<tr>
<td>( t_4 )</td>
<td>1.146</td>
<td>0.466</td>
<td>0.917</td>
<td>0.282</td>
<td>0.745</td>
<td>0.302</td>
<td>0.659</td>
<td>0.188</td>
<td>0.768</td>
</tr>
<tr>
<td>( t_5 )</td>
<td>1.184</td>
<td>0.365</td>
<td>0.796</td>
<td>0.233</td>
<td>0.808</td>
<td>0.466</td>
<td>0.509</td>
<td>0.215</td>
<td>0.786</td>
</tr>
<tr>
<td>( t_6 )</td>
<td>1.021</td>
<td>0.459</td>
<td>0.889</td>
<td>0.236</td>
<td>0.792</td>
<td>0.506</td>
<td>0.552</td>
<td>0.149</td>
<td>0.695</td>
</tr>
<tr>
<td>( t_7 )</td>
<td>1.438</td>
<td>0.389</td>
<td>0.730</td>
<td>0.316</td>
<td>0.681</td>
<td>0.563</td>
<td>0.518</td>
<td>0.192</td>
<td>0.776</td>
</tr>
<tr>
<td>( t_8 )</td>
<td>1.509</td>
<td>0.359</td>
<td>1.032</td>
<td>0.354</td>
<td>0.621</td>
<td>0.753</td>
<td>0.212</td>
<td>0.145</td>
<td>0.790</td>
</tr>
<tr>
<td>( t_9 )</td>
<td>1.328</td>
<td>0.368</td>
<td>1.006</td>
<td>0.346</td>
<td>0.552</td>
<td>0.864</td>
<td>0.204</td>
<td>0.155</td>
<td>0.846</td>
</tr>
<tr>
<td>( t_{10} )</td>
<td>1.459</td>
<td>0.419</td>
<td>0.721</td>
<td>0.296</td>
<td>0.724</td>
<td>0.673</td>
<td>0.181</td>
<td>0.167</td>
<td>0.778</td>
</tr>
</tbody>
</table>

\[
F_1 f_{1,1} - F_2 f_{2,1} - F_3 f_{3,1} = 0
\]
\[
F_1 f_{1,2} - F_2 f_{2,2} - F_3 f_{3,2} = 0
\]

All the variables in this system are considered to be in a quasi-steady state, so that a direct estimate of the covariance matrix of the state variables at some time \( t_0 \) yields inaccurate results. Typical samples of the expected and measured values of the system variables are shown in Tables 1 and 2. Owing to the non-linearity of the process constraints (equations 14 - 16), the relationship between the constraint residuals \( r \) and the measurement errors \( e_F \) and \( e_f \) is ill-defined (the true values of \( F_1 \) and \( f_{1,j} \) are unknown).

To obtain an estimate of the covariance matrix \( V_{F,f} \) of the process variables \( F_i \) and \( f_{i,j} \) \((i = 1, 2, 3; \ j = 1, 2)\), an arbitrary set of values of the variables is corrupted by errors with known covariances (and zero means) and a neural network is subsequently trained by means of these artificially generated exemplars to construct an internal representation of the relation between the covariances of the resultant constraint residuals and those of the variable residuals. Since this relationship is not uniquely defined, additional assumptions are therefore necessary in order to constrain the problem. For the purposes of this example, the variances are all assumed to be approximately equal.

The particular set of values used as a basis for the generation of synthetic training data is not critical, as long as it is sufficiently large to ensure that the covariances of the constraint residuals (on which the estimation of the covariances of the variable measurements will be based) would be a subset of the training set. Failure to do so could result in grossly inaccurate estimates of the covariances of the variable measurements. In addition to the relationship between the statistical properties of variable measurements and those of the constraint residuals, all measurements are considered to be totally independent, so that all off-diagonal
elements of the covariance matrix are deemed to be zero. This information is incorporated directly into the structure of the neural net, which could subsequently be trained to construct an internal representation of the relationship between the variances of the constraint and measurement residuals.

The neural network trained to estimate the covariance matrix of the state variables consisted of an input and an output layer only. The input layer was composed of three computational elements, corresponding to the variances of each of the three constraint residuals, while the output layer consisted of nine process nodes which corresponded to the variances of each of the nine system variables. The network was trained by repeatedly presenting it with exemplars of the relation between the variances of the constraint and measurement residuals. Training of the sigmoidal output units was accomplished by the generalized delta rule [9,10], through which the weights of the network could be modified until it was able to form an internal representation of the relationship between the covariance matrix of the flow variables $V_{F,F}$ and the covariance matrix of the constraint residuals $V_{r,r}$, as shown in Figure 1. After convergence, the trained network was used to estimate the covariance matrix $V_{F,F}$ of the system variables, by presenting it with the computed sample variances of the constraint residuals. The estimates obtained by the neural network are compared with the actual values in Table 3.

To further illustrate the way in which neural networks can be used in conjunction with heuristic data to estimate variances of sets of measurements, the assumption that the covariances of the variables are zero is modified by assuming a correlation between the measurements of the component mass fractions $f_{1,1}'$, $f_{1,2}'$, $f_{2,1}'$, $f_{2,2}'$, $f_{3,1}'$ and $f_{3,2}'$, i.e. non-zero covariances $\text{cov}(f_{1,1} | f_{1,2})$, $\text{cov}(f_{2,1} | f_{2,2})$ and

![Figure 1. Relationship between variances of measurement and constraint residuals, as represented by neural network.](image-url)
Table 3. Estimated and actual variances of process variables.

<table>
<thead>
<tr>
<th>Actual variances</th>
<th>$F_1$</th>
<th>$F_2$</th>
<th>$F_3$</th>
<th>$f_{1,1}$</th>
<th>$f_{1,2}$</th>
<th>$f_{2,1}$</th>
<th>$f_{2,2}$</th>
<th>$f_{3,1}$</th>
<th>$f_{3,2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0134</td>
<td>0.0016</td>
<td>0.0057</td>
<td>0.0009</td>
<td>0.0073</td>
<td>0.0012</td>
<td>0.0066</td>
<td>0.0008</td>
<td>0.0077</td>
<td></td>
</tr>
</tbody>
</table>

Estimates based on direct method (equation 2.1)

<table>
<thead>
<tr>
<th>Estimates made by neural net</th>
<th>$F_1$</th>
<th>$F_2$</th>
<th>$F_3$</th>
<th>$f_{1,1}$</th>
<th>$f_{1,2}$</th>
<th>$f_{2,1}$</th>
<th>$f_{2,2}$</th>
<th>$f_{3,1}$</th>
<th>$f_{3,2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0120</td>
<td>0.0015</td>
<td>0.0051</td>
<td>0.0008</td>
<td>0.0069</td>
<td>0.0011</td>
<td>0.0061</td>
<td>0.0007</td>
<td>0.0071</td>
<td></td>
</tr>
</tbody>
</table>

Owing to the quasi-steady state of the system, direct computation of the sample covariances of the system from measurements is inaccurate once again and as a consequence the covariances have to be estimated by an indirect procedure. A neural network similar to the one used previously in conjunction with the assumption of zero covariances can again be used. In this case the net’s structure would have to be modified to accommodate the three covariances of the mass fraction variables. The network is consequently composed of an input layer with three process nodes (one for each variance and covariance of the constraint residuals), as well as twelve output nodes (one for the variance of each of the nine variables, as well as the three covariance elements).

After training, the network is presented with the sample variances of the residuals, from which the nine variances and three covariances are estimated. The estimated and the actual covariances of the process variables are shown in Table 4. (Since estimates of the variances are very similar to those shown earlier on, only the estimates of $\text{cov}(f_{3,1} \mid f_{3,2})$, $\text{cov}(f_{2,1} \mid f_{2,2})$ and $\text{cov}(f_{3,1} \mid f_{3,2})$ are shown in Table 4.) It is clear that the estimates made by the neural net are more accurate than the estimates based on direct computation of the sample covariances.

3. Detection and Isolation of Systematic Errors in Steady State Systems

By presenting a neural network with examples of process measurement and constraint residuals as input, and appropriate classes indicating the presence of different types of errors as output, the network can be trained to generalize the relationship between residuals and the types of errors giving rise to these residuals [11-16]. When presented with test vectors consisting of constraint and/or measurement residuals, it is then able to assign the input to the error categories it had been trained to recognize (analogous to the statistical hypothesis tests traditionally
Table 4. Estimated and actual covariances of process variables.

<table>
<thead>
<tr>
<th>Actual covariances</th>
<th>Estimates based on direct method (equation 2.2)</th>
<th>Estimates made by neural net</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{cov}(f_{1,1} \mid f_{1,2}) )</td>
<td>0.00090</td>
<td>0.00117</td>
</tr>
<tr>
<td>( \text{cov}(f_{2,1} \mid f_{2,2}) )</td>
<td>0.00118</td>
<td>0.00138</td>
</tr>
<tr>
<td>( \text{cov}(f_{3,1} \mid f_{3,2}) )</td>
<td>0.00077</td>
<td>0.00124</td>
</tr>
</tbody>
</table>

used to categorize errors). Since neural networks are not limited by the nature or distributions of the process constraints (unlike many statistical methods), they can be used to considerable advantage in different error detection schemes. Two such strategies are outlined in this chapter.

The first strategy is the simplest and can be used to detect gross errors in sets of variables associated with nodes in the process circuit, similar to the global test in statistics [17]. Note that this strategy as such does not allow the location of errors beyond the sets of variables associated with nodes in the process circuits. Despite this drawback the strategy is useful in that no information regarding the true values of the process variables is required. The method is based on the effect that measurement residuals have on the process constraints of the system. The measurement and constraint residuals are directly proportional, i.e. zero-valued measurement residual vectors (associated with the true or reconciled values of the process variables) generate zero-valued constraint residual vectors, while a monotonous increase in the measurement residuals also results in a corresponding monotonous increase in the constraint residuals. Systematic errors (which are usually significantly larger than random errors) generally result in constraint residuals that are larger than normal, and which can be distinguished from smaller constraint residuals which are usually associated with smaller random errors.

In traditional statistical test methods (assuming that the distribution functions of the variable measurements are known) the detection of gross errors is limited to linear or linearized process constraints which have essentially the same types of distribution functions as the process variables. When the constraints are non-linear their distributions are generally unknown and the constraint residuals can consequently not be subjected to statistical hypothesis testing. By making use of neural networks, this restriction is obviated, since the network can learn arbitrary distributions of the constraint residuals a priori, as is explained below.

The general detection strategy involves training a neural network with examples of constraint residuals generated by measurement residuals of a known...
class. No mathematical models or explicit parameter specifications are involved in
the process — the data used to train the network are the standard from which the
network learns the distinction between residuals considered to be normal and those
considered to be indicative of a bias in the process data. Plant data can be used
for training the net, but artificial data are also convenient, since they are easy to
generate and there is no uncertainty as far as the classes to which the residuals be­
long are concerned. During the training process, the network constructs an internal
model of the relationship between the constraint residuals and the classes associ­
ated with the germane measurement residuals. This model can subsequently be
used to detect gross errors in measurements not encountered previously, as shown
in the following examples.

3.1. Example 2: Measurement data from a two-product classifier con­
taining gross errors of various magnitudes

Consider a two-product classifier, such as a hydrocyclone, in which the feed
stream $W_1$ with component mass fractions $x_{11}$ and $x_{12}$ is classified into two product
streams, $W_2$ with component mass fractions $x_{21}$ and $x_{22}$, and $W_3$ with component
mass fractions $x_{31}$ and $x_{32}$. All flow variables $W_i$ and $x_{ij}$ are measured and the
following mass balance constraints have to be satisfied:

$$W_1 - W_2 - W_3 = 0$$  \hspace{1cm} (17)
$$W_1 x_{11} - W_2 x_{21} - W_3 x_{31} = 0$$  \hspace{1cm} (18)
$$W_1 x_{12} - W_2 x_{22} - W_3 x_{32} = 0$$  \hspace{1cm} (19)

The above system consisting of nine state variables and three bilinear pro­
cess constraints is typical of measurement data reconciliation problems in chemical
or metallurgical process engineering, in which total flow streams and component
flows have to be reconciled. In order to demonstrate the detection and isolation
of gross errors in the system, an arbitrary but consistent set of values (in effect
assumed to be the true values of the variables) is corrupted by both random and
gross errors. The corruption of the vectors is allowed to occur on a random basis,
so that measurement vectors can contain more than one gross error. All errors
have Gaussian distributions with standard deviations of 12%, and differ only with
regard to their expected values.

3.1.1. General structure of the neural network

The network used to detect or isolate systematic errors consists of an input
layer and an output layer, sparsely connected in a feedforward mode (i.e. only
corresponding process units in the input and output layers are connected). State
changes in the process units are effected through a hyperbolic tangent transfer
function, which maps input to the $[-1;1]$ domain. Sigmoidal transfer functions,
which are commonly used in feedforward networks, were also investigated, but
did not perform as well as networks operating with hyperbolic tangent activation functions. The net was trained by means of the generalized delta rule [9], so that weight changes were proportional to the error ($e_i$) and the input to the particular connection ($z_i$). A momentum term ($m_{ij}$) equal to the change in the weight of the process element was used to facilitate the smoothing of weight changes, which resulted in better convergence of the network.

3.1.2. Training data

The sets of training exemplars generally consisted of $n$ training vectors $V_k$, each of which could be partitioned into a subvector $V^I_k$ containing all the inputs to the net, and a subvector $V^O_k$ which contained the desired outputs associated with the corresponding inputs, i.e.

$$V_k = \{V^I_k | V^O_k\}_k$$ (20)

The inputs to the network were based on either the constraint residuals ($r_p$), the measurement residuals ($W''_i - W'_i$ or $x''_{ij} - x'_{ij}$), or the measurements of the variables ($W'_i$ or $z_{ij}$). Each of these approaches is considered in turn.

3.1.3. STRATEGY I: Detection of the presence of gross errors in variables based on the constraint residuals of the system.

In the first approach the inputs to the network consist of the magnitudes of the normalized constraint residuals generated by the errors in the observed values of the variables, that is

$$v^{IN}_p = \frac{|r_p|}{\sigma_p}; p = 1, 2, \ldots n + 1$$ (21)

(where $n$ is the number of components in the circuit)

$$v^O_i = 0,$$ (22)

if $v^{IN}_i$ is associated with a random error ($i = 1, 2, \ldots n + 1$)

$$v^O_i = 1,$$ (23)

if $v^{IN}_i$ is associated with a biased error ($i = 1, 2, \ldots n + 1$)

A training (100 vectors) and a test set (50 vectors) were constructed from constraint residuals generated by corrupting a single set of arbitrary consistent measurements \{\text{variables} \} with the previously mentioned random and gross errors.

After convergence the performance of the network was evaluated against a test set consisting of 50 test vectors, with a composition similar to that of the
exemplars in the training set. These vectors consisted of a set of input values, as well as a set of actual output values which could be compared with the output values generated by the network.

On average the network managed to correctly classify 94% of the large errors with a 50% bias in the data. Although virtually no overlap occurred between the gross error and random error distributions, it should be borne in mind that based on an analysis of the constraint residuals of the system, gross errors in relatively small variables might still be indistinguishable owing to random noise in relatively large variables. As could be expected, the net had somewhat less success (91%) in discriminating between the smaller errors with a 20% bias which were less discernible from the random errors in the data. (It is not difficult to generalize this trend, with results similar to those of other error detection tests; i.e. the smaller the bias of the gross error, the smaller the chance of the error being detected.)

In order to evaluate the robustness of the net, the true values of the variables were changed completely and corrupted at random with the same errors used previously (standard deviations of 5% and biases of 20% and 50%). The previously trained network was subsequently used to detect the presence of gross errors in the new system without retraining. The ability of the network to detect gross errors was found to be essentially the same as in the previous case (within 5%). Stated differently, as long as the magnitudes of the constraint residuals are larger than a certain threshold value, the network will interpret this as an indication of the presence of a gross error, regardless of the actual or observed values of the variables used to generate the constraint residuals.

Note that by using a method such as this gross errors cannot be isolated, although these errors can be confined to subsets of three variables at a time. That is if a gross error is present in any of the flow streams $W_i$, all three constraint residuals $r_1$, $r_2$, and $r_3$ will test positive for the presence of a gross error. If the gross error occurs in any of the variables $x_{11}$, $x_{21}$, or $x_{31}$, only $r_2$ will test positive for a gross error, while only $r_3$ will test positive if $x_{21}$, $x_{22}$, or $x_{32}$ contains a gross error. In order to locate gross errors more precisely, procedures using the measurements or measurement residuals themselves are necessary.

### 3.1.4. STRATEGY II: Detection of the presence of gross errors in variables based on the measurement residuals of the system.

In this approach the same procedure is followed as in the previous approach, except that the inputs to the network consist of the normalized magnitudes of the measurement residuals of the variables, instead of the constraint residuals.

\[
V_i^{\text{IN}} = \frac{|W''_i - W'_i|}{\sigma_{W_i}}; (i = 1, 2, 3)
\]  \(24\)

\[
V_{2i+j+1}^{\text{IN}} = \frac{|x''_{ij} - x'_{ij}|}{\sigma_{x_{ij}}}; (i = 1, 2, 3; j = 1, 2)
\]  \(25\)
\[ v_k^{OUT} = 0, \quad (26) \]

if \( v_k^{IN} \) is associated with a random error \( (k = 1, 2, \ldots 9) \)

\[ v_k^{OUT} = 1, \quad (27) \]

if \( v_k^{IN} \) is associated with a biased error \( (k = 1, 2, \ldots 9) \)

Note that these residuals are regarded as the differences between the true and observed values of the variables. In practice the true values of the variables are not known and have to be estimated by an appropriate reconciliation procedure.

A network with the same general structure as the network used with the constraint residuals of the system was defined, i.e. a single input and output layer consisting of hyperbolic tangent process units. The network was trained with a set of 200 exemplars and after convergence the network was evaluated against a set of test data as before. The network was able to classify all the gross errors with a 50% bias (equivalent to ten standard deviations) correctly, but could only classify approximately 84% of the errors with a 20% bias (equivalent to four standard deviations) correctly.

Tests on data not contained in the net’s training data base, showed that the network could detect virtually all errors with a 100% bias (i.e. approximately eight standard deviations), regardless of the specific values of the variables with which these errors were associated, as shown in Figure 2.

![Figure 2. Gross error detection in a 2-product classifier.](https://scholar.sun.ac.za)
but more complicated neural network structures (typically characterized by a single hidden layer) would then be required.

3.2. Example 3: Detection of gross errors in a non-linear system

This example has been used previously by Pai and Fisher [18], as well as Tjoa and Biegler [6], and comprises five measured variables $x_i'$ ($i = 1, 2, \ldots, 5$) and three unmeasured variables $x_i'$ ($i = 6, 7, 8$), subject to six non-linear constraints.

\[
\begin{align*}
\frac{1}{2}(x_1)^2 - 0.7x_2 + x_3x_6 + (x_2)^2x_6x_7 + 2x_3(x_8)^2 - 255.8 &= 0 \\
x_1 - 2x_2 + 3x_1x_3 - 2x_2x_6 - x_2x_7x_8 + 111.2 &= 0 \\
x_3x_6 - x_1 + 3x_2 + x_1x_7 - x_3(x_8)^\frac{3}{2} - 33.57 &= 0 \\
x_4 - x_1 - (x_3)^2 + x_7 + 3x_8 &= 0 \\
x_5 - 2x_3x_7x_8 &= 0 \\
2x_1 + x_2x_3x_6 + x_7 - x_8 - 126.6 &= 0 \
\end{align*}
\] (28)

The exact values of these variables are $x = \{4.5124, 5.5819, 1.9260, 1.4560, 4.8545, 11.070, 0.61647, 2.0504\}^T$ [6]. Tjoa and Biegler [6] considered the reconciliation of these variables with a hybrid successive quadratic programming (SQP) method which was used to minimize an objective function based on a joint probability distribution of both random and gross errors. The performance of the algorithm is considered in more depth later on in this chapter, when data reconciliation with neural networks is investigated.

In order to compare the error detection capability of a neural network with the method proposed by these authors, 100 data sets were corrupted with 10% Gaussian noise to simulate random errors. In case 1 the 100-vector set is further corrupted with gross errors with a bias equal to four times the standard deviation ($\sigma_i$) of the random errors. The gross errors were distributed equally among the five measured variables ($x_i'$) and in all 20% of the measurement vectors were corrupted (one gross error per measurement vector only). In case 2 every fifth variable set was completely corrupted with gross errors (i.e. five gross errors per variable set) and in case 3 a gross error was placed in each data set for measurements $x_i'$ to $x_5'$ in rotation. Since the success of the method is to a large extent ascribable to the ability of the reconciliation algorithm to generate unbiased estimates of the true values of the process variables, an exact comparison of the method with a neural network is not possible unless the same reconciliation procedure is used in both cases. Nonetheless if the adjustments or residuals arising from the reconciliation of the measurements by the SQP method are considered to be unbiased as concluded by Tjoa and Biegler [6], a reasonable comparison can be made by evaluating the response of the neural network to artificially generated unbiased residuals.

The neural network consisted of an input layer with five process elements (corresponding to the five measured values $x'$ only, as it was not necessary to take
the unmeasured variables into account) and an output layer with five sigmoidal process elements (one again for each variable $x_d$).

Training proceeded with the use of the normalized cumulative delta rule and after convergence of the network after approximately 20,000 iterations, the net was used to detect errors in test data sets 1, 2 and 3. The method proposed by Tjoa and Biegler [6] detected approximately 73% of the gross errors in case 1, 60% of the gross errors in case 2 and 69% of the gross errors in case 3. The neural network detected approximately 72%-74% of the gross errors in all cases. These results should only be regarded as an indication of the capability of a neural network however, since especially in cases 2 and 3 the method used by Tjoa and Biegler [6] might have had to contend with some bias in the measurement residuals prior to evaluation, not taken into account when testing the network.

4. Reconciliation of Inconsistent Process Data

4.1. General material and energy balance problem

Process circuits are often described in terms of a network consisting of $m$ branches and $n$ nodes, usually so that the nodes correspond with process units in the circuit, and branches correspond with connections or flow streams between the units [19-23]. The topology of the circuit can then be described with the use of a Boolean incidence matrix $A(m, n)$, resulting in a set of material balance equations of the form

$$d(x) = A \cdot x = 0 \quad (29)$$

where $x$ is the vector representing the material variable measurements of the system. It should be noted that although these parameters could represent any desirable physical entity, such as particle size fractions, chemical species or specific gravity, the choice of the particular parameter is related to the structure of the incidence matrix $A(m, n)$. A mill for example, would have a profound effect on a flow parameter representing a particle size fraction, but would merely serve as a conduit for a parameter representing a chemical species. Although the general data reconciliation problem is also concerned with variable classification and the determination of unmeasured variables, only measured variables subject to small random errors are considered here, in order not to unduly complicate the evaluation of the parallel systems investigated.

The solution of the problem in effect ensures that the flow parameters are adjusted as little as possible, that all the conservation constraints are satisfied, and that the more reliable variable measurements are adjusted less than the less reliable variable measurements.

Most conventional optimization procedures involve the identification of the overdetermined measurement errors, followed by rectification of these errors, and
then the determination of determinable unmeasured flow parameters. These procedures are used in conjunction with schemes for the identification and elimination of systematic errors, and the readjustment of flow variables where necessary [21, 24].

These techniques usually involve iterative procedures and are computationally demanding, especially as far as large complex plants are concerned, or where on-line material balancing is required. These disadvantages associated with the use of traditional methods make the use of connectionist systems or neural networks an attractive alternative for the optimization of mass balance problems.

The use of neural networks for the reconciliation of process data have only recently been studied. Terry and Himmelblau [25] have demonstrated the use of feedforward neural networks to reconcile inconsistent process data derived from the simulation of a steady-state heat exchanger. In the remainder of this chapter the use of feedback neural network systems for the reconciliation of process data is considered. The architectures of these systems are first considered, after which their performance is evaluated by way of two case studies.

4.2. Types of connectionist systems

Recurrent or feedback networks, especially those known as crossbar or Hopfield networks, have been used for a wide range of optimization problems, ranging from the solution of non-polynomial (NP) complete combinatorial problems, such as the travelling salesperson problem [26], combinatorial optimization problems subject to inequality constraints [27], assignment problems [28], systems of complex-valued linear equations [29], the four colour mapping problem [30], the identification and recognition of visual images [31, 32], as well as the solution of linear [26, 33, 34], non-linear [35, 36] and dynamic programming [37] problems. These networks differ from feedforward systems in that information is not only passed forward through the layers of the net, but backwards or laterally as well. The performance of three different connectionist systems explained below and referred to as CS-I, CS-II and CS-III were investigated.

4.3. Connectionist system I (CS-I)

The architecture of CS-I corresponds to that of a crossbar or Hopfield neural network, as shown in Figure 3. The system consists of three layers, viz. an input layer, a hidden layer with full lateral connections, as well as an output layer. All layers have the same number \( N \) of elements, and all are provided with linear transfer functions, of the form

\[
g(u) = k_1u + k_2, \quad (k_1, k_2 \text{ constant})
\]

The exact number of elements in each layer is determined by the number of process variables to be reconciled (i.e. one element for each process variable).
4.3.1. Neurodynamics

When these networks are viewed as dynamic systems, the network computation process can be seen as a system moving in a state space\(^1\) through the constant application of some transition rules. These transition rules are procedures for updating the state of the system, depending on its current state. The system dynamics or neurodynamics of the network are determined by the transition rule, as well as the order in which the system variables or node outputs are updated. If application of the transition rule ceases to affect the current state of the system, the system is said to have converged to a fixed point or attractor in the state space. The set of all initial states or points leading to this fixed point is known as the attractor basin of the particular attractor [38].

In order to analyze the dynamics of the system it is usually convenient to define a scalar function, which depends on the state of the system and has a definite value for each point in the state space. If the value of this energy or cost function \(E\) does not increase with a change in the state \((v)\) of the system (i.e. \(\frac{dE}{dv} \leq 0\), for all possible \(v\)) and is bounded from below, it is also a Lyapunov function, and an indication that the system is unconditionally stable.

By mapping the objective function and the constraints of an optimization problem onto this energy function, these problems can be solved in that the optimal solution to the problem is forced to coincide with the minimum energy of the

\(^{1}\)The state space of a set of variables \(X = \{x_i \mid i = 1,2,3\ldots n\}\) is the Cartesian product of the domains of the variables, i.e. \(D_\alpha = \alpha_1 . \alpha_2 . \alpha_3 \ldots \alpha_n\). The state space \(D_X\) of the set of \(n\) variables \((x)\) is thus \(n\)-dimensional.
system. The dynamics of the network amount to a constraint relaxation process, where the energy measure is defined by the degree of constraint violation of the system.

To use crossbar or Hopfield networks for material balance reconciliation problems, it is necessary to map the objective function \( F_{OBJ} \) (which incorporates the process system constraints \( d \)) onto the net, such as by defining the energy function \( E \) of the network in terms of the objective function \( F_{OBJ} \)

\[
F_{OBJ} = (x' - x'')^T \mathbf{V}_{lx}^{-1} (x' - x'') + [d(x'')]^T \mathbf{V}_{ld}^{-1} [d(x'')] \tag{31}
\]

i.e. equivalent to

\[
E = (v^0 - v)^T \mathbf{V}_{lx}^{-1} (v^0 - v) + [d(v')]^T \mathbf{V}_{ld}^{-1} [d(v')] \tag{32}
\]

where the scalar function \( E \) represents the energy of the net, \( x' \) the vector of variable measurements, \( x'' \) the vector of reconciled measurements, \( d(\cdot) \) the (equality) constraints of the system, \( v^0 \) the initial output state of the connectionist system, \( v' \) the current output state of the connectionist system, and \( \mathbf{V}_{lx} \) and \( \mathbf{V}_{ld} \) some weighting matrices.

By defining the neurodynamics of the network by means of the Newton equations, i.e.

\[
\frac{du_i}{dt} = - \frac{dE}{dv_i} \tag{33}
\]

the computational energy function \( E \) is forced to decrease monotonically, regardless of the nature of this function.

Proof [30]:

\[
\frac{dE}{dt} = \sum_i \frac{dv_i}{dt} \frac{dE}{dv_i} \tag{34}
\]

As long as the transfer function \( v_i = g(u_i) \) is continuous and non-decreasing, \( \frac{dv_i}{du_i} \) is always positive and \( \frac{dE}{dt} \) always negative or zero. The resultant state of the system can consequently be related to a solution of the problem.
The neurodynamics of the network are thus defined by a set of ordinary differential or difference equations, which have to be integrated at each time increment to determine the output states of the neurons after each change of state.

\[
\frac{du_i}{dt} = -\frac{dE(v)}{dv_i} \\
u_i^{t+1} = u_i^t - \frac{dE(v)}{dv_i} \Delta t_i
\]

Integration of these equations continues until the system has reached a point of stability (i.e. its energy has been reduced to a minimum, so that \( \frac{du_i}{dt} = 0 \), for all \( i \)). In practical terms the system is considered to be stable when

\[
\sum_i \left| \frac{du_i}{dt} \right| \leq \epsilon
\]

where \( \epsilon > 0 \) is an arbitrary small convergence criterion. The solution of this system of non-linear equations is based on the use of a gradient descent technique, with constant step size lengths.

4.3.2. Scaling of data

Before the data are presented to the net, it is important that they are scaled to ranges that are useful with regard to the neurodynamic function being used. Without proper scaling, process elements could become saturated, which could eventually have a severe effect on the movement of the system through state space. Scaling is usually effected by normalizing the input data. After the network has processed the data, the results are descaled to the original units.

4.3.3. Connection weights

The weights of the network are defined by the variance-covariance matrix of the measurements, as well as the weights associated with the process constraint residuals, as presented in equation 32. Since estimates of the variance-covariance matrix elements are often not available, a weight matrix based on the actual values of the variable measurements will be defined. This ensures that the values of small variables are not adjusted by increments that are unduly large in relation to the value of the variables themselves.

The scaled input (measured values of the process variables) of the network is clamped to the input layer, and the states of the elements in the hidden layer are updated repeatedly and asynchronously (simulated by a random updating procedure) through numeric integration of the potential of each element. The system is allowed to settle into a minimum point, and the output of the hidden layers (the solution) is passed forward to the process elements in the output layer, from where it is descaled to yield a solution to the optimization problem.
4.4. Connectionist system II (CS-II)

This system is essentially a generalized version of the first one, in that instead of having a single hidden layer, it has a $P$-dimensional array of hidden layers (if $P = 1$, the system reduces to CS-I), each containing $N$ elements in general, as shown in Figure 4. The input section of the system consists of a single input layer, each element of which is connected to a corresponding element in each of the $P$ hidden layers. The elements in the hidden layers are similarly connected to corresponding elements in the output layer. The input and output layers do not process the data, but merely serve as distribution points for data input and output.

The same neurodynamic principles concerning CS-I are applicable, except that once the measurement vector has been fed to each of the different layers in the hidden array, $P$ different sets of initial conditions are generated in the array prior to the commencement of relaxation of the energy of the network. Cycles of state changes are allowed to take place independently in each layer in the array, and when necessary the state of layer is compared with those of its neighbours and the neighbouring state associated with the lowest energy is assumed by the particular layer in the array. Each layer is then again allowed to relax from a stochastically reinitialized condition close to the previous lowest energy state. Communication with a particular element and other elements in the network is allowed to take place only after a particular element has become trapped in an energy minimum. This ensures that the movement of the hidden layer active in the deepest attractor basin in the array is not slowed down unnecessarily by frequent polling to assess the states of its neighbours.
4.5. Connectionist system III (CS-III)

Direct search procedures are attractive for the solution of sets of non-linear equations, since they are easy to use and computationally efficient. A direct random search procedure with systematic search space contraction, such as proposed by Luus and Jaakola [39] and Luus and Wang [40], has been incorporated in the neurodynamics of the third system, shown schematically in Figure 5. CS-III is equivalent to CS-II, with the difference that instead of a gradient-based search, use is made of a direct method with a systematic reduction in the search space associated with each interval. The reduction in the search intervals associated with each of the search variables leads to a more efficient search procedure, since unless the search domain is in the immediate vicinity of the optimum, convergence by means of a random search can be very ineffective [41]. The procedure is implemented as follows:

1. Set the time increment counter $j = 1$.

2. Set up the system, so that the initial states ($v^0$) of the artificial neurons in all $P$ hidden layers correspond to the measured values ($x'$) of the process variables.

3. Define an initial search range $R^0_i$ for each of the system states $v^0_i$ of the neural network.

4. Determine $P$ sets of values, so that $v^j_i = v^{j-1}_i + \phi^{j-1}_i R^{j-1}_i$, where $\phi^{j-1}_i$ is a
random number associated with the state of artificial neuron \( i \) at time \( j - 1 \), and \( 0 \leq \phi_{i}^{j-1} \leq 1 \), for all \( i \) and \( j \).

5. Of these \( P \) sets, determine the set which minimizes \( \sum_{i} \left| \frac{du_{i}}{dt} \right| \).

6. If \( \sum_{i} \left| \frac{du_{i}}{dt} \right| \leq \epsilon \), terminate the search, if not, reduce the search ranges \( R_{i}^{j} \) by an amount \( \delta \), i.e. \( R_{i}^{j} = (1 - \delta).R_{i}^{j-1} \). If \( \delta = 1 \), terminate the search, if \( \delta \leq 1 \), repeat the procedure.

After convergence a set of values \( v \) will remain, which corresponds to a minimum in the energy of the system, i.e. where \( \frac{du_{i}}{dt} = -\frac{dE}{dv_{i}} \approx 0 \), for all \( i \). This minimum will be the one closest to the initial state of the system, and in multimodal systems it might be necessary to incorporate stochastic procedures which would allow the system to find a global minimum point. The incorporation of procedures such as these was not pursued in this investigation.

In the following two examples, the use of these connectionist systems is demonstrated.

4.6. Example 4: Two-product classifier

In this example a two-product classifier (such as a hydrocyclone or a screw classifier) is once again considered, which classifies a feed stream \( (F_{i}) \) with \( n \) components into two output streams \( (F_{2} \text{ and } F_{3}) \). Measurements of the flow rates \( (F_{i}) \) and component concentrations \( (f_{i,j}) \) typically violate the mass conservation equations pertaining to the classifier, viz.

\[
F_{1} - F_{2} - F_{3} = 0
\]

\[
F_{1}.f_{1,i} - F_{2}.f_{2,i} - F_{3}.f_{3,i} = 0, \text{ for } i = 1, 2, \ldots n \quad (38)
\]

The simulated output of connectionist system CS-I is summarized in Table 5 for a two-component system. As can be seen from Figure 6, which portrays the performance of the CS-I system, the value of the objective function (energy of the net) decreases rapidly at first for step sizes smaller than 0.2, after which diminishing progress is made with further computation. (A constant step size was used for all the variables throughout the optimization procedure.) Step sizes larger than 0.2 resulted in unstable behaviour of the system. The iteration steps referred to in Figure 6 comprise cycles through which each variable is updated once on average. For the two-product classifier, with 21 process variables \( F_{i}, f_{i,j} \) \( (i = 1, 2, 3 \text{ and } j = 1, 2, \ldots 6) \), an iteration step thus consisted of a series of 21 random variable selections and subsequent adjustments of the selected variables. The reconciled values of the flow rates \( F_{i} \) and concentrations \( f_{i,j} \) resulted in a threefold order of magnitude decrease in the objective function (energy function of the net), which is more or less comparable to results obtainable with other optimization techniques.

The performance of CS-II (number of layers = 10 and 100) is compared with that of CS-I (number of layers = 1) in Figure 7. It is clear that the additional layers
in the system do not lead to a significant improvement in performance. This is not surprising, since the process system considered is subject to bilinear constraints only, and does not have a highly non-linear character.

The connectionist system based on direct search techniques (CS-III) did not perform very well compared to those based on gradient descent techniques (CS-I and CS-II), as can be seen from Figure 8. The system used in this case consisted of 200 layers and had an initial range of 0.1 for each search variable. This range was contracted to zero as the search progressed, but only resulted in a decrease of about 60%-70% in the initial energy of the system. Other initial search ranges and contraction procedures did not lead to significantly better results.

4.7. Example 5

This example (see also example 3) is based on the one used by Pai and Fisher [18], as well as Tjoa and Biegler [6]. It is thus possible to make a rough comparison of the performance of the neural network with the computational procedures used by these authors. As mentioned previously, the example involves five measured variables \( x_1, x_2, x_3, x_4 \) and \( x_5 \) and three unmeasured variables \( x_6, x_7 \) and \( x_8 \) subject to six non-linear constraints, see equation 28.

Based on the exact values of these variables (see example 3), Tjoa and Biegler [6] corrupted 100 sets of these data with Gaussian noise in order to conduct a statistical evaluation of a tailored objective function in a non-linear computational routine, as well as a hybrid successive quadratic programming (SQP) routine.

Figure 6. Performance of CS-I: Effect of step size on decrease in energy.
Table 5. Reconciled and measured values of the process variables in the two-product classifier (example 4).

<table>
<thead>
<tr>
<th>TWO-COMPONENT SYSTEM&lt;sup&gt;a&lt;/sup&gt;</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Measured</td>
<td></td>
</tr>
<tr>
<td>( F_1 )</td>
<td>( F_2 )</td>
</tr>
<tr>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>0.961</td>
<td>0.602</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Reconciled (CS-I)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( F_1 )</td>
<td>( F_2 )</td>
</tr>
<tr>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>0.956</td>
<td>0.608</td>
</tr>
</tbody>
</table>

<sup>a</sup>Ratio of initial energy of system to that of final energy: \( \frac{E_i}{E_f} \approx 11000 \)

---

**Figure 7.** Energy decrease with CS-II: Effect of number of layers on decrease in energy.
In order to evaluate the use of a neural network to reconcile inconsistent constrained data, the exact values of the variables are similarly corrupted by Gaussian noise of 10% and 30%. The errors of a single set of variables resulting from the errors of corruption are shown in tables 6 (10% noise) and 7 (30% noise).

One of the salient features of the system is the highly irregular response surface of the energy function of the network. The consequence of this highly nonlinear character of the system is that the energy function is extremely sensitive to adjustment of the variables, especially at points where the derivative of the energy or objective function with regard to the adjustable variable ($\frac{\partial E}{\partial z_i}$ or $\frac{\partial F}{\partial z_i}$) is very large (positive or negative). As a result very small time steps had to be used in order to ensure that the adjustment of a variable does not lead to overshooting of a local optimum in the energy function surface.

In the case of network CS-I, the optimal step size for each variable is determined by a subroutine which systematically decreases the value of the initial time step if at first it does not result in a decrease in the system energy, until an improvement in the objective function is found. In this way relatively large time steps can be taken initially, which can be adjusted near troublesome spots on the surface of the energy function when necessary.

The results which compare favourably with those obtained by other nonlinear methods [6] are shown in Tables 6 and 7, and typically led to a reduction of three orders of magnitude in the energy of the system after approximately 40 iteration steps. The percentage errors in the values of the variables before and after reconciliation (compared to the exact values of the variables) are also shown in tables 6 and 7. Figure 9 depicts some of these results graphically. Note that step sizes larger than approximately $10^{-5}$ lead to an unstable search procedure.
Table 6. Reconciled and corrupted values (10% Gaussian noise) of the process variables used in example 5.

<table>
<thead>
<tr>
<th>CORRUPTED VALUES $^a$$^b$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$x_5$</th>
<th>$x_6$</th>
<th>$x_7$</th>
<th>$x_8$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4.786</td>
<td>5.564</td>
<td>1.917</td>
<td>1.365</td>
<td>5.307</td>
<td>10.225</td>
<td>0.617</td>
<td>2.064</td>
</tr>
<tr>
<td>$e_1$</td>
<td>$e_2$</td>
<td>$e_3$</td>
<td>$e_4$</td>
<td>$e_5$</td>
<td>$e_6$</td>
<td>$e_7$</td>
<td>$e_8$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6.06</td>
<td>-0.32</td>
<td>-0.47</td>
<td>-6.25</td>
<td>9.32</td>
<td>-7.63</td>
<td>0.09</td>
<td>0.66</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RECONCILED VALUES (CS-I) $^a$$^b$</th>
<th>$x'_1$</th>
<th>$x'_2$</th>
<th>$x'_3$</th>
<th>$x'_4$</th>
<th>$x'_5$</th>
<th>$x'_6$</th>
<th>$x'_7$</th>
<th>$x'_8$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4.742</td>
<td>5.694</td>
<td>1.903</td>
<td>1.398</td>
<td>4.927</td>
<td>10.942</td>
<td>0.597</td>
<td>2.041</td>
</tr>
<tr>
<td>$c_1$</td>
<td>$c_2$</td>
<td>$c_3$</td>
<td>$c_4$</td>
<td>$c_5$</td>
<td>$c_6$</td>
<td>$c_7$</td>
<td>$c_8$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5.09</td>
<td>2.01</td>
<td>-1.19</td>
<td>-3.98</td>
<td>1.49</td>
<td>-1.16</td>
<td>-3.16</td>
<td>-0.46</td>
</tr>
</tbody>
</table>

$^a$Ratio of initial energy of system to that of final energy: $\frac{E_a}{E_f} = 1000$

$^b$The percentage error values $e_i$ were calculated as $100 \frac{(x'_i) - (x_i)}{(x'_i)}$

Table 7. Reconciled and corrupted values (30% Gaussian noise) of the process variables used in example 5.

<table>
<thead>
<tr>
<th>CORRUPTED VALUES $^a$$^b$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$x_5$</th>
<th>$x_6$</th>
<th>$x_7$</th>
<th>$x_8$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3.713</td>
<td>4.699</td>
<td>1.365</td>
<td>1.528</td>
<td>3.680</td>
<td>9.080</td>
<td>0.622</td>
<td>2.658</td>
</tr>
<tr>
<td>$e_1$</td>
<td>$e_2$</td>
<td>$e_3$</td>
<td>$e_4$</td>
<td>$e_5$</td>
<td>$e_6$</td>
<td>$e_7$</td>
<td>$e_8$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-17.72</td>
<td>-15.82</td>
<td>-29.13</td>
<td>4.95</td>
<td>-24.19</td>
<td>-17.98</td>
<td>0.09</td>
<td>29.63</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RECONCILED VALUES (CS-I) $^a$$^b$</th>
<th>$x'_1$</th>
<th>$x'_2$</th>
<th>$x'_3$</th>
<th>$x'_4$</th>
<th>$x'_5$</th>
<th>$x'_6$</th>
<th>$x'_7$</th>
<th>$x'_8$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5.163</td>
<td>5.367</td>
<td>1.860</td>
<td>1.078</td>
<td>5.041</td>
<td>11.833</td>
<td>0.593</td>
<td>2.451</td>
</tr>
<tr>
<td>$c_1$</td>
<td>$c_2$</td>
<td>$c_3$</td>
<td>$c_4$</td>
<td>$c_5$</td>
<td>$c_6$</td>
<td>$c_7$</td>
<td>$c_8$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>14.42</td>
<td>-3.85</td>
<td>-3.43</td>
<td>-25.96</td>
<td>3.84</td>
<td>6.89</td>
<td>-3.81</td>
<td>19.54</td>
</tr>
</tbody>
</table>

$^a$Ratio of initial energy of system to that of final energy: $\frac{E_a}{E_f} = 1000$

$^b$The percentage error values $e_i$ were calculated as $100 \frac{(x'_i) - (x_i)}{(x'_i)}$
Figure 9. Performance of CS-II: Effect of step size on decrease in energy table.

(compare with values larger than 0.2–0.3 in the previous example). The use of different step sizes for the different search variables (MULTISTEP) instead of a constant step size for all variables, resulted in considerable improvements in the performance of the system.

In contrast to the situation highlighted by example 4, much is to be gained by using a multilayer system such as CS-II. In Figure 10, the significant improvement in convergence based on the use of 10 layers, versus 1 (CS-I) is illustrated. This can be attributed to the non-linear character of the response surface of the energy function. By making use of CS-II, movement through the state space of the system is accelerated along steeper attractor basin gradients, than is the case when CS-I is used.

Connectionist system CS-III displays the same less favourable convergence behaviour as was the case in the previous example, as shown in Figure 11. Exponential contraction of the range of the system results in somewhat better performance, compared to a linear reduction.

4.7. Computational efficiency of neural networks

Judging from the reduction in the objective function (equation 31) of the reconciliation problem, the results obtained with neural networks simulated on a computer appear to be comparable to those normally associated with traditional non-linear optimization methods, even though the neurodynamics used in these networks are relatively basic. If necessary the results can be improved upon by making use of more sophisticated neurodynamic functions. These functions could incorporate other stochastic procedures such as simulated annealing and its variants.
Figure 10. Performance of CS-II: Effect of number of layers on decrease in energy (example 5).

Figure 11. Performance of CS-III.
or hill climbing terms [30] to avoid entrapment in local minima, while moving the system through state space.

The real advantage of using neural networks for data reconciliation problems is the fact that they can be implemented in electronic hardware which could fully exploit the massively parallel architectures of the networks. By making use of analog devices [45], which typically converge in the characteristic time of the artificial neurons (in the order of $10^{-6}$ to $10^{-3}$ seconds), rapid computation is possible [46].

Since this investigation was based on the use of simulated neural networks, and not actual analog networks implemented in electronic circuits, no direct conclusions can be made with regard to the temporal aspects of the computational procedures. A rough estimate of the speedup is provided by Amdahl's law

$$ S = \frac{P}{P(1 - \Omega) + \Omega} \quad (39) $$

where $S$ is the speedup factor, $P$ the number of processors working on the task, and $\Omega$ the fraction of the programming code which can be executed in parallel.

### Table 8. Estimated speedup factors for examples 4 and 5.

<table>
<thead>
<tr>
<th>EXAMPLE</th>
<th>SYSTEM</th>
<th>NO OF PROCESSORS</th>
<th>SPEEDUP FACTOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Search for first local minimum)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>CS-I</td>
<td>21</td>
<td>10.50</td>
</tr>
<tr>
<td>2</td>
<td>CS-I</td>
<td>8</td>
<td>5.93</td>
</tr>
<tr>
<td>(Search for global minimum)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>CS-IIa</td>
<td>4200</td>
<td>19.91</td>
</tr>
<tr>
<td>1</td>
<td>CS-IIb</td>
<td>4200</td>
<td>19.91</td>
</tr>
<tr>
<td>2</td>
<td>CS-IIa</td>
<td>1600</td>
<td>19.77</td>
</tr>
<tr>
<td>2</td>
<td>CS-IIb</td>
<td>1600</td>
<td>19.77</td>
</tr>
<tr>
<td>1</td>
<td>CS-III</td>
<td>4200</td>
<td>0.42</td>
</tr>
<tr>
<td>2</td>
<td>CS-III</td>
<td>4200</td>
<td>0.42</td>
</tr>
</tbody>
</table>

The time consumed by computational overheads was estimated to be not more than approximately 5% for all three connectionist systems, and on this basis and the results of the optimization experiments, it was possible to estimate the speedup factors for the solution of the data reconciliation problems outlined in examples 4 and 5. These estimates are summarized in Table 8. Two different situations are highlighted in the table, namely the location of a solution (local minimum) of the problem, and secondly the location of a global solution or minimum.
to the problem (by combining a stochastic procedure with a gradient descent or direct search method). The gradient descent methods (CS-I and CS-II) performed significantly better than the direct search procedure (CS-III). As can be expected, the larger the problem, the more is gained by making use of these parallel strategies. According to equation 39 the speedup factor is also quite sensitive to the fraction of computer code that can be executed in parallel (estimated to be 95% in this investigation).

The quality of the solutions obtained with the simulated networks indicates however, that analog networks could be employed to considerable advantage to solve data reconciliation problems.

The problems posed in example 5 presented non-linearities of a higher degree than the problem discussed in example 4. This meant that smaller time steps had to be implemented to ensure a monotonic decrease in the energy of these systems, and as a result these systems took longer to converge than the bilinear two-product classification system. After approximately 10 iteration steps or cycles (see Figure 6) the energy of the bilinear system discussed in example 4 (nine variables) did not show further significant decreases for step sizes larger than 0.3. The energy or objective function of the system considered in example 5 (eight variables) on the other hand, decreased by approximately two orders of magnitude after 25 iterations (and showed a decrease of approximately three orders of magnitude after 40 iteration steps).

![Figure 12. Comparison of the performance of CS-II with other non-linear data reconciliation techniques.](image)

In Figure 12 the CS-II system is compared with two other non-linear procedures used for the solution of the problem posed in example 5, viz. that of Broyden
[18,47] and the constant derivative approach [48]. From this graph it can be seen that the CS-II system initially (steps 1 to 3) decreases the value of the energy or penalty function faster than the other two methods. In subsequent iteration steps it loses ground, but in the end (steps 11 and 12) the advantage gained by the methods of Broyden and constant derivatives is largely eradicated. It should be borne in mind that this comparison can serve as a rough guideline only, since the central processing unit (CPU) times associated with the execution of the iteration steps in the different algorithms cannot be compared directly. If anything, a comparison of actual CPU times could only be to the advantage of the CS-II system with its relatively simple computational procedures.

Another important factor that should not be overlooked is that in principle the efficiency of the CS-II system is not affected significantly by an increase in the dimensionality of the process system, while other non-parallel procedures such as those depicted in Figure 12 are usually sensitive to increases in the size of the problem. In large systems consisting of hundreds or even thousands of variables, the CS-II system can consequently be expected to perform significantly better than any other traditional procedure [49].

5. Conclusions

In this chapter the use of neural networks for processing of plant data prior to modelling was explored. This included the estimation of variance-covariance matrices associated with measurements, the detection and isolation of gross errors in the plant data, as well as the reconciliation of inconsistent data after elimination of gross errors.

With regard to variance estimation and gross error detection, these techniques follow much the same approach as methods based on classical statistics. They are thus also subject to the same prerequisites, viz. analytical redundancy of the process data and a knowledge of the distribution of errors in the data. An important point is the fact that the knowledge concerning these errors does not have to be explicit when neural networks are used (i.e. the usual assumption of a normal distribution with a certain variance and a zero mean), since the network can derive its own representation of this distribution when provided with sufficient suitable process data. Once the network has constructed a model of the random errors in the measurements (from examples represented to it in the training phase), it uses this representation as the exclusive standard against which errors are classified, or on which the estimation of variances are based.

As far as the reconciliation of process data is concerned, the main advantage of neural networks is the fact that they can be used as fast parallel processors which can deal with large problems, or with on-line systems requiring rapid calculation.

These techniques are still in an exploratory stage however, and more sophisticated techniques are currently being developed. Despite this, the potential of neural networks is clear, and they can be expected to play a significant role in process modelling and control in future.
References


**Nomenclature**

\[
\begin{align*}
A(m, n) & \quad m \times n \text{ Boolean incidence matrix of a process circuit} \\
b & \quad \text{vector of systematic errors or biases} \\
C & \quad \text{matrix of process constraint coefficients} \\
\text{cov}(-) & \quad \text{covariance} \\
d(-) & \quad \text{system constraints} \\
E & \quad \text{scalar energy or cost function serving as a measure of the overall state of a feedback neural network} \\
E(\cdot) & \quad \text{expected value} \\
e & \quad \text{random error vector or measurement}
\end{align*}
\]
residuals associated with observed variables \( x'' \)

\( e_F \) random error vector associated with observed process flow streams

\( e_f \) random error vector associated with observed component mass fractions

\( e_i \) error associated with the output of the \( i \)'th node in the output layer of a supervised neural network

\( D_x \) state space of set of variables

\( F \) flow rates of a process streams \( \{ F_i \} \)

\( f_i \) flow rate of \( i \)'th stream in a process circuit

\( F_{OBJ} \) objective function in general

\( f \) component mass fractions \( \{ f_{i,j} \} \) in process streams

\( f_{i,j} \) \( j \)'th component mass fraction in the \( i \)'th flow stream in a process circuit

\( f'_{i,j} \) \( j \)'th measured component mass fraction in the \( i \)'th flow stream in a process circuit

\( g(\cdot) \) transfer function of a node in a neural network

\( k_1, k_2 \) parameters

\( m_{ij} \) momentum term used in the adjustment of the weight connecting nodes \( i \) and \( j \) of a neural network during training

\( P \) number of processors in a parallel computer system

\( R^0_i \) initial search range associated with the \( i \)'th output node of a neural network

\( R^j_i \) search range associated with variable \( i \) at time \( j \)

\( r \) constraint residuals associated with observed variables \( x'' \)

\( r_p \) \( p \)'th constraint residual of a process system

\( S \) speedup factor

\( t \) time

\( \Delta t_i \) \( i \)'th time increment

\( u \) potential of a node in a neural network

\( u^i \) potential of \( i \)'th unit in the output layer of a neural network at time \( t \)

\( V \) variance-covariance matrix

\( V_{xz} \) variance-covariance matrix of variables \( x \)
$V_k$  
$k$'th exemplar of the training data set of a neural network

$V_{IN}^k$  
inputs of the $k$'th exemplar of the training data set of a neural network

$V_{OUT}^k$  
outputs of the $k$'th exemplar of the training data set of a neural network

$v$  
output state of a feedback neural network

$v'$  
current state of a feedback neural network

$v^0$  
initial state of a feedback neural network

$\text{var}(\cdot)$  
variance

$\nu_i$  
output state of $i$'th node in a feedback neural network

$v_j^*$  
output state of the $i$'th node at time $j$

$W_{i}$  
$i$'th feed stream in a process circuit

$W_{i}''$  
adjusted value of $i$'th process flow stream $W$

$W_{i}'$  
obscerved value of $i$'th process flow rate $W$

$x$  
true values of variables

$x''$  
obscerved values of variables

$x'$  
adjusted values of variables

$x_{ij}$  
$j$'th component mass fraction in the $i$'th stream of a process circuit

$x_{ij}''$  
adjusted value of $j$'th component mass fraction in the $i$'th process flow stream

$x_{ij}'$  
obscerved value of $j$'th component mass fraction in the $i$'th process flow stream

$z_i$  
i' th node in the output layer of a neural network

Greek letters

$\alpha_i$  
i' th variable in general

$\delta$  
incremental reduction of search range

$\epsilon$  
convergence criterion

$\phi_i^j$  
random number associated with the state of the $j$'th node at time $i$

$\sigma_p$  
standard deviation of $p$'th constraint residual

$\sigma_{W_{i}}$  
standard deviation of flow stream $W_{i}$

$\sigma_{x_{ij}}$  
standard deviation of component mass fraction $x_{ij}$

$\Omega$  
the fraction of the programming code which can be executed in parallel
This paper proposes a new methodology to model ill-defined processing operations using neural nets (NN's). A process with many variables (large dimensionality) cannot be modelled adequately if limited process data are available. This problem of multi-dimensionality is addressed and an approach suggested to reduce the dimensionality using NN's. The result of this modelling methodology is a combined model of phenomenological expressions and NN's. The model can successfully be incorporated in a dynamic simulator of the process. A new approach to conduct modelling on the basis of continuous data collected directly from an industrial processing unit, is also proposed. The modelling methodology is applied to a typical processing operation, the resin-in-pulp (RIP) process.

INTRODUCTION

Most processing operations in the chemical and metallurgical industry are ill-defined in some way. This means that the features of a process cannot be described uniformly through the whole process domain with a phenomenological model. In the search for modelling techniques to address this problem, various techniques have been investigated such as empirical modelling, traditional regression analysis, as well as KBS or expert systems. Some recent work by Reuter et al (1991) showed how KBS can be applied to simulate CIP (carbon-in-pulp) and CIL (carbon-in-leach) circuits. From batch reactor simulations, they constructed system variable profiles (such as gold concentration profiles) which could be coupled with the dynamics of the adsorption and leaching processes through a data base. Although KBS can be powerful modelling tools, constructing them is usually time-consuming and expensive.

Connectionist networks (neural nets) are mathematical regression tools which seem to be very promising for a wide range of modelling problems. Although still in its infancy, encouraging results of various applications have been published to date. Most research in neural nets (NN's) has focused on developing classifiers and tools for recognition. Also in the process industry NN's were used in this regard, where substantial work is in progress to use NN's for process control, fault diagnosis and optical classifiers.

Although process modelling forms an integral part of process engineering, little work has been done to employ NN's for modelling purposes. The main reason for this is the problems caused by high-dimensional variable spaces (many process variables determining the kinetics of the process), which are unique to process operations in general. This problem arises with any modelling problem, but becomes even worse if an "intelligent interpolator" such as an NN is to be used. Various authors such as Poggio et al (1990) and van der Walt et al (1991) have recently identified this problem and started investigating this matter. It can be
explained synoptically as follows: Consider a function $f$ which is depicted on an $m$-dimensional variable space $\mathbb{R}^m$. Let $Y_i$ be the average number of data points necessary within the $i_{th}$ dimension to represent the whole system sufficiently, then the minimum number of data points needed ($P$) will be

$$P = \prod_{j=1}^{m} Y_j$$

(1)

This shows that the amount of information (such as process data points) required to model a process adequately, increases exponentially with a rise in the number of process variables.

This paper will show how process identification can be conducted by analyzing data which are representative of the process, by means of an NN. The NN sorts the data and forms a "data base" that can be employed to identify some features of a process. Firstly the NN is employed to perform simple perturbation analyses in order to quantify the influences of the separate variables within different domains of the variable space. By doing so, less-influential variables can be eliminated from a specific domain of the variable space and the dimensionality reduced accordingly. In a next step, the "data base"-NN identifies mathematical relations, which in turn can be described through mathematical expressions, thus further reducing the "ill-defined" dimensionality. This results in a number of separate variable spaces or subspaces, each one containing a function surface within the subspace of reduced dimensionality. Some variables within a specific subspace will be related to the function surface through mathematical expressions. The relations between the function and the remaining dimensions are ill-defined and can be mapped by using an NN. The final product is a combination of variable subspaces, each one consisting of relational mathematical expressions, combined with an NN-representation within the ill-defined dimensional space of a specific subspace.

The Resin-in-pulp (RIP) process will be used here to illustrate the proposed methodology for NN-modelling. This process is a typical metallurgical processing operation used for gold recovery and is still poorly understood, and therefore ill-defined.

It will also be shown how process identification can proceed on the basis of continuous operating data. In practice a model compiled on the basis of batch data needs, almost without exception, to be adjusted considerably if it is to be incorporated in a simulator for a continuous system. For this reason process identification should preferably be conducted on the basis of continuous data.

**MODELLING ON THE BASIS OF CONTINUOUS DATA**

The dynamics of a continuous reactor can be described by performing material, momentum and energy balances over the reactor. Differential equations are the standard language for expressing such dynamics. Equation 2 represents a typical material balance equation for a substance $X$ in a CSTR:

$$\frac{\epsilon}{\tau} \frac{dX}{dt} = X_{in} - X_{out} + f$$

(2)

$f$ represents the reaction(s) which take place within the reactor. This reaction term can be expressed in terms of the other three terms of the latter equation:

$$f = \frac{\epsilon}{\tau} \frac{dX}{dt} - X_{in} + X_{out}$$

(3)

The terms on the right-hand side of equation 3 should be measurable so that $f$ can be calculated with this expression. At the same instant when $f$ is determined, some other process variables which may have an influence on $f$, are evaluated as well and combined with $f$ to form a process data point. These data points can be used by the process engineer to develop a model for $f$.

An adequate phenomenological model for $f$ can only be formulated if the chemistry and mechanics of the process are reasonably well understood. In the case of an NN modelling approach, the data set consisting of the abovementioned data points, is employed as a "training set" by an NN training program to learn the relations between the different process variables and functions.

**NEURAL NETS**

Recent success in connectionist network research is mainly attributed to the achievements in multilayer perceptrons with backpropagation. Various NN topologies have been proposed, and the most novel and exciting approach is the field of radial basis function networks. Sanger (1991) proposed a tree-structured adaptive network for approximating functions in high-dimensional spaces. The learning procedure compiles a tree, the structure of which depends upon both the input data and the function to be approximated. It can be used to reduce the number of required measurements in situations where there is a cost associated with sensing.

The NN used for the purposes of this paper is a three-layered feedforward NN (three-layer perceptron) which is trained through a backpropagation algorithm. The
weight matrix is adjusted through conjugate-gradient optimization with restart procedures during training, as described by Powell (1977). The activation function of the nodes in the input and output layer is linear, while the hidden layer's nodes are sigmoidal.

**DYNAMIC SIMULATION OF A TYPICAL PROCESSING OPERATION**

The NN modelling methodology proposed in this paper is done using a typical metallurgical processing operation as example. The Resin-in-pulp (RIP) process is a gold-recovery process which still holds numerous secrets concerning its reaction mechanism. Various phenomenological models have been developed for this processing operation. Although some of these models can be used successfully to predict the kinetics of the process within certain operating domains, their domain-specificity is a major drawback for fullscale industrial implementation.

In order to illustrate the mentioned methodology, artificial data points were generated by using process variable profiles together with the material balance equations 5 to 7. This procedure of data generation will be described in greater detail below.

In the next few paragraphs the RIP processing plant layout, the material balance equations representing the process dynamics, as well as the phenomenological model describing the process kinetics, are described.

**RIP plant layout with material streams**

In Figure 1 the RIP-cascade is illustrated schematically. It consists of a number of exchange reactors in series. Countercurrent flow of gold ore slurry, containing gold cyanide in solution, takes place through the cascade. There is a constant slurry stream down the cascade. Resin is transferred periodically upward, during which an upstream slurry flow also takes place. The volumetric flow rate of the slurry stream entering the cascade from the leaching section into the first column. The volumetric flow rate of the slurry which is transferred upstream during resin transfer periods, is expressed by $Q_s$. The volumetric flow rate of the total downflow slurry stream ($Q_{f5}$) can be calculated as follows:

$$Q_{f5} = Q_f + Q_{sp}$$  \[4\]

During the intervals when no resin transfer takes place, $Q_s$ is non-existent and $Q_{f5}$ equals $Q_f$ for all columns.

**Material balance equations**

The dynamic behaviour of the RIP-cascade can be simulated through a number of differential equations representing material balances for gold within the different phases. With the exception of the first and last reactors, the three equations 5 to 7 describe the gold mass balances over each intermediate reactor in the liquid, resin and ore phases. The gold concentration in the liquid phase is represented by $C$, while $q$ and $G$ are the average gold loading on the resin and the average gold grade in the ore respectively. The masses of ore and resin within a specific reactor are $M$ and $W$ respectively. The void fractions for liquid ($l$), ore ($o$) and resin ($r$) are expressed by $\varepsilon$.

**Gold balance in liquid phase for the $i^{th}$ stage:**

$$e_i V_i \frac{dC_i}{dt} = -\left(\frac{e_{bi}}{e_{bi} + e_{ao}}\right)Q_{b1} C_{i-1} + \left(\frac{e_{bi}}{e_{bi} + e_{ao}}\right)Q_{b1} C_{i-1} + e_{fi} (Q_{s1} + Q_{f1}) C_i + f_i$$ \[5\]

**Gold balance on resin for the $i^{th}$ stage:**

$$W_i \frac{dq_i}{dt} = -\left(\frac{e_{rt}}{e_{rt} + e_{ao}}\right)Q_{r1} p_r q_{r1} + e_{rt} Q_{s1} p_r q_{r1} + f_2$$ \[6\]
Gold balance on ore for the \( f \)th stage:

\[
\frac{dG_f}{dt} = - \frac{e_{a_1}}{(e_{a_1} + e_{a_2})} Q_{a_1} p_0 G_{a_1} + \frac{e_{a_3}}{(e_{a_4} + e_{a_5})} Q_{a_3} p_0 G_{a_3} - \frac{e_{a_7}}{(e_{a_8} + e_{a_9})} (Q_{a_7} + Q_{a_8}) p_0 G_f + f_3
\]  

If \( f_1 \), \( f_2 \) and \( f_3 \) are known, these differential equations can be solved by numerical techniques such as the fourth-order Runge Kutta integrator. These three reaction terms can be expressed in numerous ways through different modelling techniques. If a data base which captures the knowledge of the process kinetics (KBS) is available for \( f_1 \), \( f_2 \) and \( f_3 \), their values can be determined at each instant in time during a simulation run. The same holds for phenomenological, regression analysis or NN models for the reaction functions.

The phenomenological model

The phenomenological model employed by a phenomenological model simulator (dynamic simulator incorporating a phenomenological model) was used to generate artificial data points for the RIP-process. \( f_1 \), \( f_2 \) and \( f_3 \) in equations 5 to 7 are the kinetic reaction terms and can be described by the following equations according to the film diffusion model for exchange on the resin surface and an empirical expression for the process of leaching still taking place within the RIP-cascade:

\[
f_2 = \frac{6 \ k_r W_r}{p_r d_r} (C_f - C_n) \]

\[
f_3 = - k_r M_f (G_f - G_n)^2 \]

\[
f_1 = -(f_2 + f_3) \]

The equilibrium at the resin surface is described by the Freundlich isotherm of equation 11.

\[
C_n = \left(\frac{C_f}{A}\right)^n \]

A and \( n \) are the equilibrium parameters, specified to be 10 and 0.15 respectively for the specific resin. Further properties of the resin are a density \( (p) \) of 1100 kg.m\(^{-3}\) and an average particle diameter \( (d) \) of 1.0 mm. The adsorptive and leaching coefficients \( (k_r \ & k) \) are assumed to remain constant at \( 10^4 \text{m.sec}^{-1} \) and 1.2 kg\(_{\text{RIP}}\) \text{gAu}^{-1}\text{sec}^{-1} \) respectively. The grade of gold in the ore at infinity \( (G^\infty) \) was specified to be \( 8 \times 10^{-6}\text{gAu.kg}^{-1}\).

**CONSTRUCTING AN NN-PHENOMENOLOGICAL MODEL**

The procedure followed to compile an NN-phenomenological model is conducted through a number of steps. In the first instance a global NN is trained with all process data points. This global NN is used to divide the process variable space into subspaces. This division is done by performing a perturbation analysis on the training data set, whereafter these same perturbation results are used to identify less-influential variables within every individual subspace. These non-significant variables can then be eliminated and the dimensionality of the various subspaces reduced. During the next step an NN is trained for each subspace on data points containing only the values for the significant variables within a specific subspace. Such NN’s have less input nodes and should therefore be able to perform improved curve fittings due to a reduction in the degrees of freedom. Hence, the generalization or interpolation properties of the NN’s should be better as well. The trained NN of each subspace is now employed to identify simple mathematical relations between a function and certain variables within a subspace. The remaining dimensions with unknown relations to the function can now be mapped during a further step by an even simpler NN. The result will be an empirical model consisting of a combination of phenomenological expressions and small NN’s.

**Constructing an artificial data set**

During simulation runs with the phenomenological model simulator for the RIP-process, variable profiles were constructed which covered the process domain in which the simulation runs were completed as portrayed in the final part of this paper. Using these profiles together with the balance equations 5 to 7, \( f_1 \), \( f_2 \) and \( f_3 \) were calculated and the values of \( C, q, G, M \) and \( W \) registered together with the three reaction terms. This was done in order to illustrate how continuous data can be gathered for modelling purposes, as explained earlier. The values for \( M \) and \( W \) were varied between 400-500 tons and 12-22 tons respectively. All simulations were done at fixed reactor volumes of 750 m\(^3\) for all stages. The values of \( f_1 \), \( f_2 \) and \( f_3 \) of these data points (variable-function pairs with 5 input variables and 3 output functions) were then randomly corrupted with 15% Gaussian noise. The first training set thus contained 800 noisy data points which is
typical of an industrial system. After scaling the values of the data points, this training set was employed to train a NN which in turn was used to perform perturbation analyses on the training data set.

**Scaling the training data**

It is very important to scale the values of the variables and functions (input-output data pairs) appropriately. There are three principal reasons for this. Firstly the input node activities of the NN should be bounded so that the weights between the inputs of the sigmoidal hidden nodes are sensible (their absolute values should not be too large). This will cause extremely slow training convergence. Secondly it is imperative that the input values to the net are of the same absolute order. During training each weight is adjusted according to the overall smallest error-weight gradient. The error-weight gradient is directly proportional to the magnitude of the input node according to backpropagation. A very small input node will thus prevent the weight that connects it to the hidden layer, to make a significant contribution to the objective function and will slow down the training process. Thirdly, whilst learning is complicated if highly curved function surfaces should be found, NN's can easily learn smooth relations.

The input-output data pairs for the RIP-example were scaled as illustrated in Figure 2. The variables C, q, and G varied over a few orders of magnitude and were therefore scaled logarithmically. M and W were linearly scaled. All scaled values were transformed into the range -3.5 to 3.5. Figure 3 illustrates the distribution of the scaled variables. It shows that the number of scaled values for all variables C, q, G, M and W are more or less symmetrically distributed around zero.

**Training an NN**

The search for an adequate NN for a specific application is very important. A number of factors should be considered prior to training an NN. One of these factors is the number of hidden nodes to be specified. Although techniques have been proposed by Huang and Huang (1991) as well as Baum and Haussler (1989) to determine an optimal NN-configuration for NN's used as binary classifiers, no effective procedure exists to estimate this optimal architecture for NN's employed as function approximators. An excessive number of hidden nodes will cause bad generalization properties, so that the NN would not be able to interpolate effectively between adjacent training data points. As opposed to this, too few hidden nodes will limit the competence of the NN to locate an adequate mapping between functions and variables. The importance of scaling the input-output training pairs appropriately should again be emphasized.

The global NN was specified to have 10 hidden nodes. It was trained with the 5-dimensional training data set containing 800 data points. It has been proved that NN training algorithms with gradient descent optimization procedures converge more easily to a satisfactory weight matrix if noisy data are used. This was also the case for the NN trained here. This trained NN was used to perform perturbation analyses on the training data.

**Perturbation analysis**

Although some sensitivity analysis methods such as
factorial design are available, it is difficult to use them effectively on practical process operating data containing noise. Contrary to this, NN's approximate analysis of a single variable's influence on a specific effect effectively on practical process operating data smooth function curves which allow their convenient use in perturbation analysis for the whole system. Shootout data points are eliminated prior to data analysis.

Procedure for perturbation analysis: The perturbation analysis of a single variable's influence on a specific function at a set point in variable space is used to explain the procedure followed to analyze the process variable space. Figure 4 illustrates how the sensitivity of a function \( f \) towards a variable \( X_i \) can be quantified if an NN relates \( f \) with an \( m \)-dimensional variable space \( \{X_1, X_2, \ldots, X_m\} \). At a given variable space coordinate \( \{x_{1i}, x_{2i}, \ldots, x_{mi}\} \) of data point \( i \), the value of \( f \) is calculated by the trained NN (note that the scaled NN output value is scaled back to the function value). This predicted function value is the perturbation midpoint and the corresponding value for \( X_i \) the variable midpoint \( \langle x_i \rangle \). \( X_i \) is now adjusted to both sides within the \( f \) dimension with the same constant absolute value (perturbation interval \( PI_i \)) for \( x_{ji} \), while the values of the other variables \( \{X_1, X_2, \ldots, X_{j-1}, X_{j+1}, \ldots, X_m\} \) are kept fixed. \( PI \) is determined by a simple formula on the assumption that the function surface within the immediate vicinity of the coordinate is flat. This means that the function \( f \) can be approximated locally by a linear \( m \)-dimensional expression in terms of \( X_i \) to \( X_m \). The expression used to determine \( PI \) is

\[
PI_i = \beta \cdot x_{ji}
\]  

where \( \beta \) is an adjustment factor whose value is constant for all perturbation calculations during an exhaustive perturbation analysis for all variables within the total bounded variable space. The perturbations done at a specific coordinate are proportional to the magnitudes of the variables, so that the perturbation results which represent the quantified influences of all variables on \( f \) at a specific coordinate can be compared directly. The means used to compare these results are explained later.

A relatively small value for \( \beta \) should be selected due to two reasons. Firstly, the variable midpoint should not be adjusted too drastically into the inaccurate extrapolative domains of the NN. This could result in dubious perturbation values close to the bounded variable space margins. Furthermore, the analysis at every point in the variable space should be performed only within the more or less linear function domain.

The variable midpoint \( x_{ji} \) is adjusted to both sides with \( PI_i \). The values of \( f \) at these two new coordinates are now predicted by the NN. Using the absolute differences between the perturbation midpoint and the two discriminate perturbated values for \( f \) (\( A \) and \( B \) as portrayed in Figure 4), the preliminary perturbation values (ppv) are calculated as follows:

\[
ppv = \frac{(A + B)}{2}
\]  

The ppv's for all variables \( X_i \) to \( X_m \) are computed at the coordinates for every training data point.

The normalized perturbation values (npv's) are the criteria used to quantify the relative influence of each variable \( X_i \) on \( f \) within its dimension. This facilitates the detection of a changing influence on \( f \) through a specific dimension in the variable space. If the ppv's at a coordinate \( i \) are represented by a vector \( \langle p_{1i}, p_{2i}, \ldots, p_{mi} \rangle \), the npv (\( n_{ij} \)) of a ppv within the \( f \) dimension (\( p_{ij} \)) can be determined using the normalizing equation 14.

\[
n_{ij} = \frac{p_{ij}}{\sum_{k=1}^{m} p_{ik}} \times 100\%
\]  

The npv's at a coordinate \( i \) are also expressed as a vector \( \langle n_{1i}, n_{2i}, \ldots, n_{mi} \rangle \). The influences of the different variables on \( f \) are now quantified throughout the variable space so that the magnitudes of these average perturbated values can be compared directly. Table 1 summarizes perturbation results for three randomly selected data points. The npv's at a coordinate can be compared mutually which is also the case for the ppv's. It should be apparent from Figures 5 to 7 that the npv's can be employed as a criterium for identifying a

**FIGURE 4. Perturbation analysis with trained NN - analysis of \( X_i \) at the coordinates of a data point \( i \).**
ILL-DEFINED OPERATIONS USING CONNECTIONIST NETWORKS

TABLE 1. Perturbation results of three randomly selected training data points of the RIP-example - analysis performed on \( f_2 \) for all five variables.

<table>
<thead>
<tr>
<th>( C )</th>
<th>( q )</th>
<th>( G )</th>
<th>( M )</th>
<th>( W )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preliminary perturbation values (ppv's)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.09159</td>
<td>1.05399</td>
<td>0.07077</td>
<td>0.65825</td>
<td>1.23185</td>
</tr>
<tr>
<td>0.04424</td>
<td>0.00252</td>
<td>0.00556</td>
<td>0.00742</td>
<td>0.06038</td>
</tr>
<tr>
<td>0.00285</td>
<td>0.00003</td>
<td>0.00021</td>
<td>0.00020</td>
<td>0.00290</td>
</tr>
<tr>
<td>Normalized perturbation values (npv's)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25.56</td>
<td>25.67</td>
<td>1.72</td>
<td>16.03</td>
<td>30.00</td>
</tr>
<tr>
<td>36.83</td>
<td>2.10</td>
<td>4.63</td>
<td>6.18</td>
<td>50.27</td>
</tr>
<tr>
<td>46.01</td>
<td>0.56</td>
<td>3.40</td>
<td>3.19</td>
<td>46.83</td>
</tr>
</tbody>
</table>

changing influence of a variable on the function within its dimension. The ppv's are not suitable to do so.

For the RIP-example, perturbation analyses were performed at all coordinates for the 800 training data points. \( B \) was selected to be 0.01. In this process, the orders of magnitude of the variables \( C \) and \( q \) are the same through the RIP-cascade, so that the relative tendencies of variable influences on the reaction functions \( f_1, f_2 \) and \( f_3 \) through the RIP variable space can be portrayed in Figures 5, 6 & 7 respectively. Note that these figures do not show the npv's of the variables within the dimension of \( C \). It is impossible to plot the npv's of all variables in a five-dimensional variable space and it will be unsuitable to show a complete set of graphs for the npv's within single dimensions in the limited space available.

Dividing the variable space into subspaces: The perturbation results can now be used to identify boundaries between subspaces on the basis of eliminating different non-significant variables from neighbouring subspaces.

Eliminating less-significant variables: Figure 5 portrays the perturbation results of \( f_1 \) and shows that all five variables play significant roles as variables of \( f_1 \). Notable is the strong influence of \( q \) on \( f_2 \) within the high \( C \)-region and the drop in significance of \( q \) towards the region of lower \( C \)-values within the variable space. This corresponds with the Freundlich isotherm of equation 11. According to this, the variable space is divided into two subspaces at approximately \( C = 0.2 \). It can also be noticed that \( G \) plays an increasingly important part at lower \( C \)-values, condoning the role played by the
leaching term in the phenomenological model for \( f_1 \).

It can be observed from Figure 6 that \( C \) and \( W \) play relatively strong roles in determining \( f_2 \) throughout the variable space. Also notable is the strong influence of \( q \) on \( f_2 \) within the high \( C \)-region and the drop in significance of \( q \) towards the region of lower \( C \)-values within the variable space. The \( npv \)'s for \( G \) and \( M \) remain relatively small through the RIP variable space, so that these two variables can be completely eliminated as variables for \( f_2 \). These observations correspond with the phenomenological expression for \( f_2 \) (equation 8).

The perturbation results in Figure 7 confirm the dependence of \( f_2 \) on \( G \) and \( M \). It shows that the dimensions for \( C, q \) and \( W \) can be eliminated, thus reducing the dimensionality of the variable space for \( f_2 \) drastically. The higher \( npv \)'s for \( G \), if compared with those of \( M \), are attributed to the quadratic relation between \( f_2 \) and \( G \). The variable space for \( f_2 \) is not subdivided.

The variable spaces of \( f_1 \) and \( f_2 \) have now been subdivided into two subspaces each. The first subspace of \( f_1 \) (for \( C > 0.2 \) g.m\(^{-2}\)) is 5-dimensional, while the dimensionality of the second subspace (\( C < 0.2 \) g.m\(^{-2}\)) has been reduced to four. In the case of \( f_2 \), its first subspace contains three dimensions (for \( C, q \) and \( W \)). This subspace is neighboured by the second subspace with dimensions for \( C \) and \( W \) only. The undivided variable space for \( f_3 \) has two dimensions (\( G \) and \( M \)).

Identifying mathematical relations

During the following step, different simplified NN’s are trained for the different subspaces of each function.

These NN’s (2 for \( f_1 \), 2 for \( f_2 \) and a single one for \( f_3 \)) are employed to identify mathematically definable relations between the functions and their variables. The procedure is explained by using the NN for \( f_3 \). At different domains within the variable space, the value of \( W \) is varied, while \( G \) is kept constant. \( f_3 \) is calculated at these different coordinates. Figure 8 exhibits the predicted curves at three distinct values of \( G \). It confirms the direct proportionality between \( f_2 \) and \( M \). The curves can be forced through the origin, so that \( f_3 \) can be expressed as follows:

\[
f_3 = M.r_3,
\]

where \( r_3 = a(G) \).

The same procedure was performed for \( f_2 \) and \( W \), and equation 16 displays the simplified functions.

\[
f_2 = W.r_2,
\]

where

\[
r_2 = \beta(C, q) \quad \text{if} \quad C > 0.2
\]

and

\[
r_2 = \beta(C) \quad \text{if} \quad C < 0.2.
\]

Although the relevant NN’s for \( f_1 \) predicted linear relations between \( f_1, W \) as well as \( f_1, M \), the mathematical relations could not be isolated from the "ill-defined" dimensional space. This is due to the presence of two terms in equation 10. A new NN-topology is currently investigated which should be able to extend the amount of information captured from the data set, after identifying variables such as \( M \) and \( W \) as orthogonal within the variable space of \( f_1 \). This will hopefully have the same beneficial effect as to "reduce" the ill-defined dimensionality of a subspace.

In the meantime a different approach is followed for this example. If the assumption can be made that some a priori knowledge about the RIP-process is available, such knowledge can be applied to simplify the modelling of \( f_1 \). It should be emphasized that this NN approach for modelling does not tend to replace existing modelling techniques, but should contribute to the modelling toolkit. If it is known, for example, that the gold mass balance in the liquid phase is dependent on an adsorption and leaching process, equations 15 & 16 can be combined to describe \( f_1 \).

\[
f_1 = M.r_3 - W.r_2
\]

The result: an NN-phenomenological model

According to the relations of the model equations 15 to 17, an NN-phenomenological model with three NN’s (two nets for each \( C \)-interval for \( r_2 \) and a net for \( r_3 \)) was compiled. These NN’s were trained on exactly the same training data as the NN’s trained earlier. The NN’s of \( r_2 \) and the lower \( C \)-range subspace of \( r_3 \) had only one input node (the bias node ignored), while the NN for \( r_3 \) within the higher \( C \)-range contained two input

![FIGURE 8. Linear relation between \( f_3 \) and \( W \) as predicted by a simplified, 2-dimensional NN for \( f_3 \).](image-url)
nodes only. The dimensionality of the "ill-defined" parts of the system has thus been reduced considerably from a 5-dimensional one to an ill-defined dimensionality of only 2 (function \( f_2 \) for \( C > 0.2 \)) and 1 (functions \( f_2 \) for \( C < 0.2 \) and \( f_3 \)) by means of the techniques described above. This new model was used to replace the phenomenological model in a dynamic RIP-cascade simulator. At each time step, \( f_1 \), \( f_2 \) and \( f_3 \) could now be calculated by equations 15 to 17 during a simulation run.

A second NN-model was constructed for \( f_1 \), \( f_2 \) and \( f_3 \) using the five-dimensional variable space. This model consisted of three NN’s (one net for each of \( f_1 \), \( f_2 \) and \( f_3 \)) which were trained with the same data points used to develop the NN-phenomenological model. An NN with only one output node will more easily learn the mapping between this single function and its variables as opposed to an NN with more output nodes. This was the reason for training separate NN’s for the different reaction functions. Each one of these NN’s had five input nodes for the five variables involved. This NN-model was developed to show the inadequacy of implementing an NN-model of high dimensionality in a straightforward fashion, as compared to the effectiveness of the model proposed in this paper.

COMPARING SIMULATION RESULTS

A dynamic simulator was developed which solved the differential mass balance equations 5 to 7 by incorporating the three different models for \( f_1 \), \( f_2 \) and \( f_3 \). The results of the phenomenological model simulator can be used as a basis of comparison. Figure 9 illustrates some results of simulation runs completed with the three different models under the same conditions. It should be apparent from this figure that the NN-phenomenological model is much more accurate than the NN-model of high dimensionality. Although the concentration profiles of Figure 9 represent only a small part of the simulation results, the same conclusions can be drawn from the simulated variable profiles of each RIP-reactor.

The superiority of the simulation results with the NN-phenomenological simulator (reduced dimensionality) to those of the NN-model simulator is mainly due to the large reduction in "ill-defined" dimensionality and an associated improvement of the model.

CONCLUSIONS

It has been explained how process data can be collected directly from an industrial operating unit, and a new technique to conduct process identification on the basis of such continuous data was proposed. A novel modelling approach using neural nets (NN’s) has been developed and a typical metallurgical process, the resin-in-pulp (RIP) process for gold recovery, was used to demonstrate how modelling can be conducted along this line.

In mineral processing, process identification usually deals with problems caused by multi-dimensionality. This problem becomes even worse in the case of NN-modelling. A simple method which employs an NN to perform perturbation analyses for a function through its variable space was also suggested. This forms part of the broader methodology for modelling on the basis of connectionist networks. The perturbation results can be used to subdivide a variable space into subspaces, after which less-significant variables could be eliminated, thus reducing the dimensionality of a subspace accordingly. NN’s were then employed to identify mathematically definable relations between a function and certain variables. During the final step, NN’s with small dimensionalities could be trained to capture the ill-defined relations between a function and the remaining variables for each subspace. This model (a combination of phenomenological expressions and NN’s) was incorporated in a dynamic simulator for the RIP-process. The performance of this process was successfully predicted by the simulator.

Further work is in progress to refine and automate the proposed methodology for reducing dimensionality. This also includes the design of a new NN-topology which should be able to reduce dimensionality, even within the ill-defined variable space.

REFERENCES


LIST OF SYMBOLS

A Freundlich isotherm parameter.
C Gold concentration [g.m$^{-3}$].
d Average diameter of solid particles [m].
f Reaction term in mass balances.
G Grade of gold in the ore [g.kg$^{-1}$].
k$_i$ Film diffusion constant for adsorption [m.sec$^{-1}$].
k$_r$ Leaching parameter [kg.g$^{-1}$.sec$^{-1}$].
M Mass of gold ore in a single column [kg].
n Freundlich isotherm parameter.
n Normalized perturbation value.
p Preliminary perturbation value.
p Density [kg.m$^{-3}$].
Q Volumetric flow rate of slurry [m$^3$.sec$^{-1}$].
q Gold loading on the resin.
t Time [sec].
V Volume of RIP-column [m$^3$].
W Resin mass in a single column [kg].

Greek symbols

θ Adjustment factor for perturbation analysis.
ε Void fractions.
φ Function symbol.
ICHM'92
Proceedings of The Second International Conference on Hydrometallurgy
October 23–26, 1992, Changsha, China
VOLUME 1

Editor in Chief: Chen Jiayong
Co–editors in Chief: Yang Songqing, Deng Zuoqing

Sponsored by
The Nonferrous Metals Society of China
The Chinese Nuclear Society

Hosted by
Central South University of Technology

Cosponsors:
The Institution of Mining and Metallurgy (IMM)
The Mining and Materials Processing Institute of Japan (MMIJ)
Gesellschaft für Deutscher Metallhütten und Bergleute (GDMB)
THE SIMULATION OF ILL-DEFINED HYDROMETALLURGICAL PLANTS USING ARTIFICIAL INTELLIGENCE

J.S.J. van Deventer & T.J. van der Walt
(Department of Metallurgical Engineering, University of Stellenbosch, Stellenbosch, South Africa)
M.A. Reuter
(Institut für Metallhüttenwesen und Elektrometallurgie, RWTH Aachen, Intzestrasse 3, 5100 Aachen, Germany)

ABSTRACT

Artificial intelligence techniques such as knowledge-based systems, expert systems and neural networks are useful for the simulation of ill-defined unit operations. An object-oriented knowledge-based system (KBS) is proposed here for the dynamic simulation and fault-diagnosis of hydrometallurgical plants. In this approach the dynamic behaviour of a system is finger-printed by the gradient of change of a state variable, as determined from batch kinetic data. Moreover, it is explained how a simple dynamic equation can be used to capture the mechanistic characteristics (independent of the time variable) of a unit operation from continuous dynamic data. A neural network can be used instead of a KBS to relate the gradient of change of a state variable to the prevalent process conditions in a unit process. It is also shown how the dimensionality of a neural network model can be reduced on the basis of a perturbation analysis. These techniques have been applied successfully to the simulation of batch and continuous leaching of gold, and to the adsorption of gold onto activated carbon or ion-exchange resins in countercurrent cascades.

INTRODUCTION

Despite significant advances in the modelling of hydrometallurgical unit operations and the computer simulation of processing circuits over the last decade, many unit operations are still ill-defined in the sense that few can be described by phenomenological models. Existing models are usually incapable of incorporating the effects of pH, ionic strength, adsorbent deactivation, zeta potential, and ore mineralogy. In many cases empirical models are used owing to the complexity of mineral/solution/reagent/equipment interactions. It is, however, not always possible to prescribe a priori the form of such empirical equations.

Knowledge-based systems (KBS) and expert systems have recently found increasing application for treating such ill-defined problems, especially in the chemical industry.1-8 These techniques of artificial intelligence (AI), including neural networks (NN), have also been applied in the mineral industry,9-13 as is evident from the increasing number of papers on AI at the APCOM series of symposia.18 Most KBS and expert systems have been used as management support systems and training tools, while NN's have been used mainly as classifiers, e.g. in fault diagnosis and image processing. The dynamic simulation of processing operations by means of a hybrid of simple differential equations and AI methods is a very recent development11-16,18, which will be reviewed briefly below.

An expert or KBS system is a computer programme that behaves like an expert in a usually narrow field of application.1-10 Features of a KBS include the separation of knowledge from the techniques that are used to think about this knowledge, and its capability to deal with uncertainty and incompleteness. A limitation of an object oriented KBS is the large knowledge base and accompanying rules which are necessary to describe a particular system. NN's are inherently parallel and have the capability to learn non-linear and ill-defined relationships
between a set of inputs and outputs\textsuperscript{10,18}. However, existing NN's are tedious to train for problems of high dimensionality. Therefore, a need exists for a type of connectionist network relating outputs to inputs in which the dimensionality is reduced by identifying sub-spaces within the training data.

It is the objective of this paper to broaden the concepts of dynamic simulation by AI, with special emphasis on hydrometallurgical processes such as the leaching of gold, and adsorption of gold cyanide onto activated carbon or ion-exchange resins in countercurrent cascades. It will be shown how the dynamic behaviour of a system can be finger-printed by the gradient of change of a state variable. It will also be explained briefly how process identification can be conducted. Neural networks will be used to model dynamic data for problems of high dimensionality. Owing to lack of space, examples of application will not be discussed.

**TRANSFORMATION OF KINETIC DATA**

A generalised process independent knowledge base requires a process independent generalised kinetic model. In the methodology explained below knowledge representation is based on concentration-time data, which represent the deep knowledge regarding the process. The rest of the knowledge is structured in such a way that it has access to these data. Although the generalised model given below does not attempt to suggest a reaction mechanism, it is a fingerprint of that reaction. Consequently, the model can describe a reaction under all possible conditions. This basic fingerprint curve and its associated shallow and deep level knowledge are termed the pivot-data, or the standard condition. These pivot-data serve as a reference with which other curves are compared, and which can subsequently be used for process identification.

Any condition that differs in whatever way from this standard condition or the pivot-data, is considered to be non-standard. Consequently, any change in the rate variables should be considered as the net effect of different deviations from the standard condition, and therefore as taking all interactions into account. The changes in the leaching, adsorption or other reactions may be caused by changes in the mineralogy and chemistry of the ore, deactivation of the adsorbent, or changes in the concentration of reagents.

For leaching the rate variable $k[g(t)]$ is defined here to be only a function of the grade $g(t)$, and is determined from a batch leaching curve. For the solution phase:

$$\frac{dC}{dt} = k[g(t)]g(t) = r_1$$

For the solids phase:

$$\frac{dg}{dt} = -\beta k[g(t)]g(t) = r_2$$

where:

$$k[g(t)] = 2\left(C_{n+1}-C_n\right)/\delta t(g_{n+1}+g_n)$$

$$\beta = \frac{(M_l/M_s)}{D}$$

Oxygen and cyanide are consumed as a result of the leaching reaction. This causes the leaching environment in the reactor to be changing continuously. It is therefore necessary to adjust the leaching condition as time proceeds. The depletion of oxygen and cyanide can be simulated by a simple first order rate equation, where the rate "variable" evolves with changes in the operating conditions.\textsuperscript{12}

The rate variable $k[y(t)]$ for adsorption is defined here to be a function of only the loading $y(t)$, and is used to predict the change in concentration and carbon loading at any carbon concentration $M_s$, and at any initial concentration at the prevailing chemical process conditions. Reuter gives more details.\textsuperscript{16}

$$\frac{dC}{dt} = -(\alpha_2/\alpha_4)k[y(t)]C = r_3$$

$$\frac{dy}{dt} = \alpha_3 k[y(t)]C = r_4$$
where: \[ k[y(t)] = -2 \frac{(C_{m1}-C_n)}{\delta t (C_{m1}+C_n)} \] (5)
\[ \alpha_3 = \left( \frac{M_i}{M_e} \right)_3 \] (6)
\[ \alpha_4 = \left( \frac{M_i}{M_e} \right)_4 \]

MODELS FOR CONTINUOUS FLOW REACTORS

The kinetic equations (1) to (4) can be incorporated into suitable process models which describe continuous flow systems. Non-ideal flow can be compensated for by estimating the degree of short-circuiting \( c_{c.t} = \phi C_{1.e}+(1-\phi) C_{c.t} \) and dead volume \( V_{act} = V_{total} - V_{dead} \) from tracer tests, or by comparing real plant behaviour with that predicted from an ideal flow model. An object orientated KBS can be used to relate the degree of non-ideality to the nature of the reactor system. The corresponding material balance equations for a combined leaching and adsorption system, such as carbon-in-leach (CIL) and resin-and-leach (RIL), are:

\[
\frac{dC_i}{dt} = \frac{v_{p,in}}{V_{act}} C_{i-1} - \frac{v_{p,ou}}{V_{act}} C_i + r_1 + r_3 \tag{7}
\]

\[
\frac{dy_1}{dt} = \frac{m_{c,c}}{V_{act}} [y_{i+1} - y_i] + \frac{m_{c,c}}{V_{act}} [y_{i-1} - y_i] + r_4 \tag{8}
\]

\[
\frac{dg_i}{dt} = \frac{v_{p,in}}{V_{act}} g_{i-1} - \frac{v_{p,ou}}{V_{act}} g_i + r_2 \tag{9}
\]

Depending on \( m_{c,c}, M_{c,c}, m_{c,t}, M_{c,t} \), any type of adsorbent transfer mode, ranging from a low periodicity to continuous, co-current to counter-current or a combination of these may be simulated. Reuter and Van Deventer11 showed how the philosophy of integrating KBS and dynamic material balance equations could be used to simulate the adsorption of gold or silver cyanide on activated carbon in fixed beds or periodic countercurrent columns. The material balance equations for the countercurrent leaching of gold ore in a column were explained previously.12

Clearly, \( r_1 \) and \( r_3 \), and \( r_2 \) and \( r_4 \), are related by the mass ratio of the phases in contact. As explained earlier, the intrinsic kinetic characteristics of such a system can be represented by discretising equations (7) to (9), and by applying these equations to continuous plant data.18 In this way, \( r_1 \) or \( r_3 \), and \( r_2 \) or \( r_4 \), and therefore \( k[y(t)] \) and \( k[y(t)] \), can be estimated at different positions in the space of variables determining the intrinsic kinetics of the system. This implies that the time variable is eliminated from the dynamic behaviour of the system.

OVERALL STRUCTURE OF THE KBS

Figure 1 illustrates the overall structure of the KBS, which consists mainly of: (1) A data-base for editing of all relevant experimental and heuristic information; (2) An inference procedure for utilising the data in the database to perform process identification and simulation; (3) A working memory which contains the current input and status of the specific problem being solved. The different modules of the system communicate via files in order to increase the working memory; (4) A user-friendly interface used to communicate with the above three components.

An object-oriented approach was applied in the KBS proposed here in order to define the structure of the knowledge-base. Each object (frame) is structured so as to include all relevant information regarding a particular process variable and its effect on the process, or information regarding a particular unit operation.15,16,19 Rules (production rules) could include search rules, procedural rules and adjustment rules. An example of the latter is:
if pyritic ore type B is leached by cyanide in batch and cyanide level is 160 p.p.m. and level of oxygen is 6.0 p.p.m. and temperature is 25°C and pH = 12.9 and average particle size is 74 μm and mass % solids in slurry is 47 and rate of agitation is 150 r.p.m. and time of solid-liquid contact is 2 hours then concentration gradient factor = 0.88

Neural nets\textsuperscript{18} may be used effectively instead of objects and rules to relate adjustment parameters and rate variables to conditions in the process space.

---

**Figure 1** Overall structure of simulation procedure by KBS and neural networks.
FAULT-DIAGNOSIS AND PROCESS IDENTIFICATION

Process identification of a system is conducted by a search through both shallow and deep knowledge in the KBS, subject to if-then rules and a comparison with the pivot-data. Neural networks are natural classifiers, and could be used to relate processing problems and symptoms to process conditions. Reuter gives algorithmic details about the search procedure. Once the adjustment parameters have been determined, the backward chaining inference engine determines the appropriate operating conditions from the appropriate objects, which may include neural networks. The results of such a fault-diagnosis are saved in the system so as to facilitate learning of the KBS and an improvement of the expert capabilities of the system.

For example, if sufficient knowledge about a CIL plant and other similar operations has been saved in the KBS, a fault-diagnosis could have identified reasons for an unexpectedly high level of gold in the tailings. The KBS can also be used for the identification of ores, slurries, reagents or adsorbents by comparing the behaviour of an unknown substance with that of known substances at similar operating conditions. It is essential to set a sensitivity level for the search through the data base.

NEURAL NETWORKS

The application of neural nets to processing problems has been discussed before. It is well-known that the amount of information required to model a process satisfactorily increases exponentially with an increase in the number of process variables used. This is especially true for NN, which renders the training process difficult and time-consuming. Frequently, a three-layer perceptron NN trained with a backpropagation algorithm is used. Although the activation function of the nodes in the input and output layers is usually linear, the nodes of the hidden layer are sigmoidal, this does not correspond to reality. Current research is aimed at defining activation functions which are representative of the functional forms in process engineering. It is very important to scale the values of the variables and functions appropriately, so that the input node activities should be bounded and of the same order. Furthermore, separate NN's should be trained for different reaction functions, because an NN with only one output node will learn the mapping between this single function and its variables more easily than an NN with multiple output nodes.

A procedure to divide the variable space for an adsorption-in-leach system into sub-spaces has been described in an earlier paper. Firstly, a global NN is trained with all process data points, followed by a perturbation analysis to identify and eliminate less influential variables in the individual sub-spaces. Consequently, the dimensionality of the various sub-spaces is reduced and the population density of the training points increased accordingly. During the next step an NN is trained for each sub-space. These NN's have fewer input nodes and therefore give improved fittings of the data, as well as improved interpolation properties. The trained NN of each sub-space can then be used to identify simple mathematical relationships between a function and certain variables within the sub-space. The remaining dimensions with unknown relations to the function can now be mapped during a further step by an even simpler NN. The objective is to formulate a hybrid model consisting of a combination of expressions, differential equations and small NN's.

Van der Walt et al. presented an example of how the formulation of an NN-phenomenological model (versus an NN model of high dimensionality) could lead to increased accuracy when compared with a phenomenological model for CIL.

CONCLUSIONS

A generalised non-linear first order rate expression in interaction with a KBS and/or neural network was formulated to place concentration-time data for ill-defined systems on a more fundamental basis. Whereas the KBS system with its inference engine and system of interactive menus is convenient to manage the simulation and diagnostic functions, the neural network models are in most cases more suitable to correlate input and output data for ill-defined hydrometallurgical processes. In most processes the dimensionality of a problem should be reduced by a perturbation analysis on the variable space before acceptable accuracy could be achieved with a neural network.
LIST OF SYMBOLS

\( \alpha, \beta \) Adjustment parameters

\( C, C_i, C_n \) Concentration in a batch reactor, concentration in stage \( i \) and data point \( n \) respectively [ppm].

\( g, g_i, g_n \) Ore grade in a batch reactor, grade in stage \( i \) and grade data point \( n \) respectively [g/ton].

\( k \) Rate variable as a function of \( y(t) \) and \( g(t) \) [h\(^{-1}\)].

\( M_e, M_i, M_s \) Mass concentration of carbon, liquid and solids respectively [g/l].

\( m_e \) Mass flow rate of the carbon counter-current and \( m_e^{cc} \) for co-current flow [kg/h].

\( m_{c,t} \) Mass of carbon being transferred counter-current and \( m_{c,t}^{co} \) co-current [kg].

\( r \) Rate equation [ppm/h].

\( t \) Time variable [h].

\( V_{act} \) Active stage volume [m\(^3\)].

\( v_{p,in}, v_{p,out} \) Pulp flow rate in and out of a stage [m\(^3\)/h].

\( Y, Y_i, Y_n \) Carbon loading in a batch reactor, loading in stage \( i \) and loading data point \( n \) [ppm].

\( \delta t \) Time increment.

\( \phi \) Fraction of pulp short-circuiting reactor stage.

REFERENCES

Recent Advances in the Simulation of Mineral Processing Circuits Using Artificial Intelligence

C Aldrich, J S J van Deventer, T J van der Walt and M A Reuter

ABSTRACT
Owing to their complexity, many metallurgical processes are difficult to model fundamentally, and as a result more attention has been focused on the use of artificial intelligence techniques to simulate these processes. In this paper recent advances in the use of neural nets to model processes not amenable to conventional methods are discussed. These include the application of a new modelling methodology based on the use of artificial neural nets for the simulation of metallurgical processes, the use of conventional techniques in conjunction with neural nets to simulate flowsheet circuits, as well as the detection of gross errors in variable measurements by means of a neural net representation of the relation between measurement and constraint residuals.

INTRODUCTION
Since the early 1980s progressively more attention has been focused on the use of artificial intelligence in the simulation of mineral processing circuits. The branches of artificial intelligence related to knowledge-based systems and object oriented programming have already established a firm foothold in the modelling of problems where the exhaustive search of variable spaces or numerical procedures have proved inadequate. Since knowledge-based systems make use of explicit rules and symbolic manipulation, they are not always effective as far as ill-defined problems are concerned. As a result, a considerable part of the artificial intelligence research effort in the mineral processing industry has been diverted to connectionist systems or artificial neural nets during the past few years. Besides their ability to learn by example, and to form internal representations of complex processes, these systems also have an inherently parallel nature, which makes them suitable for implementation in parallel computational procedures (and the potential to process huge amounts of data). In this paper recent advances made in the use of connectionist systems to simulate mineral processing systems are summarised. The use of neural nets at the University of Stellenbosch are discussed. These include the application of a new modelling methodology based on the use of neural nets for the dynamic simulation of metallurgical processes, the use of conventional techniques in conjunction with neural nets to simulate mineral separation process circuits, as well as the detection of gross errors in variable measurements by means of a neural net representation of the relation between measurement and constraint residuals.

CONNECTIONIST SYSTEMS
Connectionist systems are computational structures consisting of interconnected arrays of computational elements capable of processing large volumes of data simultaneously. Artificial neural nets constitute the most important class of connectionist systems and generally consist of large numbers of primitive computational elements or nodes interconnected on a massive scale, so that the nodes collectively confer certain attractive features to the nets, such as the ability to represent functional relationships, regardless of the complexity of the relationship.

In layered networks, these processing elements are typically divided into disjoint subsets, called layers, which are usually categorised as either input, hidden or output layers, to denote the way in which they interact with the information environment of the net.

The back propagation neural net models used in the majority of process modelling applications are feedforward networks which can be trained by repeatedly presenting them with exemplars of inputs and desired outputs (Bhat et al., 1990; Bhat and McAvoy, 1990; Hocht-Nielsen, 1990; Hinton, 1989; Hornik et al., 1989; Leonard and Kramer, 1990; Lippman, 1987, 1989; Rumelhart et al., 1986; Wasserman, 1989). Training, which entails the modification of the connection strengths between elements in the net (ie the weight matrix of the net), is accomplished by means of learning algorithms designed to fit an objective function, such as the least mean squares (LMS) error function to the data. The weights of the net can be adjusted on the basis of process information propagated back through the net during the learning phase and in back propagation neural nets the states of the process units in lower levels or layers of the net are usually updated before units in layers further down in the net, so that these nets are said to be feedforward and synchronous. The rules by which process units are updated, such as the generalised delta rule, are typically of the form

\[ z(t + 1) = \phi \left[ \sum w_{ij} z_j(t) - \Theta_i \right] \]

where \( \sum w_{ij} z_j(t) - \Theta_i \) designates the potential of a process unit at time t, ie the difference between the weighted sum of all the inputs to the unit and the unit bias.

The form of the transfer function \( \phi \) may vary, but it could be a linear, step or sigmoidal transfer function, among others, with a continuous derivative and a domain typically much smaller than that of the potential of the process unit, ie

\[ \phi(z) = z \quad \text{linear} \]
\[ \phi(z) = (1 + e^{-z})^{-1} \quad \text{sigmoidal} \]
\[ \phi(z) = (1 - e^{-z})^{-1} [1 + e^{-z}] \quad \text{hyperbolic tangent} \]

The training of back propagation neural nets (ie the search for an optimal weight matrix) is an iterative process involving the alteration of the weights of the net, typically by means of a gradient descent method, in order to minimise an error criterion, that is

\[ w_{ij}(t + 1) = w_{ij}(t) + \Delta w_{ij} \]

\[ w_{ij} = -\alpha e_i \sum \partial w_{ij} \]

where \( T \) is the learning rate and \( e^t \) the error criterion, ie

\[ e^t = \frac{1}{2} \sum (d_{jk} - z_{jk})^2 \]

based on the difference between the desired and the actual outputs of the unit. It is through these training procedures that neural nets form internal representations of the relationships between inputs and outputs presented to them, and which make them such powerful tools for pattern recognition, classification and modelling of especially ill-defined processes.

DYNAMIC MODELLING OF CARBON-IN-LEACH PROCESSES BY MEANS OF ARTIFICIAL NEURAL NETS
Effective modelling is often severely hampered by the high dimensionality of large process systems. The reason for this is that the minimum amount of information required for adequate

1. Department of Metallurgical Engineering, University of Stellenbosch, Stellenbosch, 7600, South Africa.
modelling of a process increases exponentially with an increase in the number of process variables, i.e. \( P = \prod y_i \), where \( P \) is the number of data points needed to model the process and \( y_i \) is the number of points required to represent the whole system sufficiently in the ith dimension. Several methods have been proposed to overcome the burden of high dimensionality, such as task dependent clustering (Poggio and Girosi, 1990), non-parametric multivariate regression analyses (Cherkassky et al., 1991) and local variable selection techniques, such as CART (Breiman et al., 1984), MARS (Friedman, 1991), ID3 (Quinlan, 1983) and C4 (Quinlan, 1987).

A new modelling methodology implemented by means of neural nets has been developed by Van der Walt et al (1992) to address the problem of dimensionality. This methodology is based on the use of neural nets to deal with the local reduction of dimensionality in an external fashion. A typical metallurgical process is modelled to illustrate the construction of a kinetic model with this new method.

**Dynamic simulation of a carbon-in-leach (CIL) process**

The carbon-in-leach (CIL) process is a metallurgical processing operation in which gold is recovered from cyanided gold ores. Despite the development of various phenomenological models, the fundamentals of the process are poorly understood and it is for this reason that non-parametric regression analysis such as represented by artificial neural nets appears to be an attractive alternative to fundamental modelling. The configuration of a typical CIL cascade is depicted in Figure 1. The process can be described as follows: Consider a CIL cascade which consists of a series of sequentially connected adsorption reactors through which an aqueous cyanide and gold ore slurry flows in a countercurrent mode. Carbon is periodically transferred upstream at a volumetric flow rate of \( Q_t \) in the presence of a continuous slurry feed to the cascade with volumetric flow rate \( Q_s \). The resultant volumetric flow rate of the total down flow slurry stream \( Q_s \) is calculated as follows:

\[
Q_s(i) = Q_t \cdot Q_{s(i+1)}, \quad i = 1, 2, \ldots, N
\]  

The dynamic behaviour of the CIL cascade can be simulated through any of ordinary differential equations which represent the process of mass transfer of gold within the liquid, carbon and ore. The average gold loading on the carbon is denoted by \( q \), the average gold grade in the ore by \( G \) and the gold concentration in the liquid phase by \( C \). The mass of ore and carbon within a particular reactor is represented by \( M \) and \( W \) respectively.

Gold balance in liquid phase in stage 1

\[
e_i \frac{dC_i}{dt} = \frac{e_1}{(e_1 + e_{o1})} Q_s, C_2 + \frac{e_2}{(e_2 + e_{o2})} Q_s, C_1 + f_1 (7)
\]

Gold balance in liquid phase in stage \( i = 2, 3, \ldots, N \)

\[
e_i \frac{dC_i}{dt} = \frac{e_{i+1}}{(e_{i+1} + e_{o_i})} Q_s, C_{i+1} + \frac{e_i}{(e_i + e_{o_i})} Q_s, C_i + f_i (8)
\]

Gold balance on carbon for stages \( i = 1, 2, \ldots, N \)

\[
W_i \frac{dq_i}{dt} = \frac{e_{X1}}{(e_{X1} + e_{o_i})} Q_s, p_i + Q_i + f_2 (9)
\]

Gold balance on ore for stage 1

\[
M_1 \frac{dG_1}{dt} = \frac{e_{o1}}{(e_{o1} + e_{o2})} Q_s, p_o, G_2 + \frac{e_{o2}}{(e_{o2} + e_{o3})} Q_s, p_o, G_3 - \frac{e_{o3}}{(e_{o3} + e_{o4})} Q_s, C_1 + f_3 (10)
\]

Gold balance on ore for stages \( i = 2, 3, \ldots, N \)

\[
M_i \frac{dG_i}{dt} = \frac{e_{o_i}}{(e_{o_i} + e_{o_i+1})} Q_s, p_o, G_{i+1} + \frac{e_{o_i+1}}{(e_{o_i+1} + e_{o_i+2})} Q_s, p_o, G_{i-1} - \frac{e_{o_i+2}}{(e_{o_i+2} + e_{o_i+3})} Q_s, G_i + f_3 (11)
\]

If the kinetic reaction terms \( f_1, f_2 \) and \( f_3 \) are known, the above sets of differential equations can be solved by means of a suitable numerical technique, such as a fourth order Runge-Kutta integration routine. Moreover, these reaction terms can be expressed in various ways, depending on the modelling technique being used. For example, if the relevant knowledge with regard to the process kinetics is embodied in a knowledge-based system, the values of \( f_1, f_2 \) and \( f_3 \) can be determined from the system’s data base at each instant during simulation. The same is valid for regression analysis and phenomenological or connectionist models of the reaction functions.

**Kinetic and equilibrium models**

The kinetic reaction terms \( f_1, f_2 \) and \( f_3 \) can be described by the equations derived from the film diffusion model for adsorption onto the carbon surface (Equation 12), as well as the leaching process taking place within the CIL cascade (Equation 13).

\[
f_2 = 6 \cdot k_y \cdot W_i \cdot (C_i \cdot C_{2i}) \cdot p_o \cdot d_e (12)
\]

\[
f_3 = -k_i \cdot M_i \cdot (G_i \cdot G_i) \cdot G_i (13)
\]

\[
f_1 = (-f_2 + f_3) (14)
\]

Equation 14 is valid, assuming that mass transfer occurs only between the liquid, ore and carbon, with insignificant interaction between the carbon and the ore. The equilibrium conditions at the
carbon surface can be described by the well-known Freundlich isotherm for a single-component system, ie
\[
C_2 = \frac{q}{A} f^\alpha
\]
where \(A\) and \(n\) are equilibrium parameters. This phenomenological model was used to simulate typical process data.

**Generation and pre-processing of typical plant data**

During simulation runs with the phenomenological model simulator, variable profiles were computed which covered the process domain at fixed reactor volumes. These profiles were used to generate data sets of the form \([C_2,C_w,G,M,W] \rightarrow f_1(f_2,f_3)\]. The training data, which consisted of 800 vectors, were corrupted with 15 per cent Gaussian noise to simulate a typical industrial system. The perturbed data were scaled to be within suitable range (approximately [-3,3]) for the connectionist network.

**Construction of the hybrid subspace model**

As a first step, various artificial neural nets with different configurations were trained with the five-dimensional training data set. The modelling performance of each trained network was evaluated with the test data set. The most successful of these nets was a net consisting of an input layer with five nodes (one for each input variable), a hidden layer with ten hidden nodes and output layer with three nodes (one for each kinetic reaction term, \(f_1, f_2\) or \(f_3\)). This network was referred to as the global neural net.

In order to reduce the dimensionality of the global variable space, the global network model was used to perform perturbation analyses at the co-ordinates of each training sample. By doing so, the relative influence of each variable on the different kinetic reaction terms could be quantified. The criterion used for this purpose was the relative perturbation values (RPVs), as defined by Van der Walt et al (1992). The RPVs for the five variables of \(f_2\) are portrayed in Figure 2. From this figure it can be seen that variables \(G\) and \(M\) exert little influence on \(f_2\), so that these variables can be eliminated from the global variable space of \(f_2\) with negligible consequences. Variables \(C\) and \(W\) play relatively important roles in determining \(f_2\) throughout the variable space. The strong influence of variable \(q\) on \(f_2\) within high \(C\)-regions and the drop in significance of \(q\) towards lower \(C\)-values can also be noted from Figure 2. These observations correspond to the dual-rate expression for adsorption, as represented by Equations 12 and 15.

![Relative perturbation values for reaction function \(f_2\).](image)

**ADVANCES IN SIMULATION OF MINERAL PROCESSING CIRCUITS**

Owing to the empirical nature of gravity concentration technology, fundamental modelling of gravity separation circuits is not feasible at present, and as a result most models are of an empirical or semi-empirical nature (Lowest and Sutherland, 1985; Laplante and Shu, 1988). Spiral gravity concentrator circuits can be modelled and optimised by making use of neural nets (representing the requisite empirical knowledge of the system) embedded in conventional computational procedures. In the example discussed in this paper, simulation is based on two linear programming models and an artificial neural net representing the performance characteristics of the separators under various operating conditions. These concentrators each separate a feed (u) stream composed of a valuable element (h), gangue (g) and water (w), into a concentrate (\(y\)), middlings (z) and tailings stream (m).

The neural net is trained to generalise the relation between the process conditions, viz the total flow rate (\(d_1\)), the dry solids flow rate (\(d_2\)) and the feed grade (\(d_3\)), and the concentrate-middlings (ym² = y/m²) and middlings-tailings (zm² = z/m²) separation

![Gold concentration profiles in reactor 1 of the CIL-cascade, as predicted by the CIL-simulator incorporating three different models.](image)
factors for each of the three elements $k$ in the circuit, as shown in Figure 4.

This distributed representation of the experimental data can then be used in conjunction with the two linear programming models to simulate and optimise the gravity separation circuit.

Simulation procedure

The strategy used to simulate the gravity separation circuit entails linearisation of the model equations (reflecting the material conservation requirements of the system) for each concentrator in the circuit, which facilitates optimisation by means of linear programming techniques. The highly non-linear character of the process is retained through the incorporation of an artificial neural network previously trained to represent the separation of a feed stream $u^*$ with a given composition of elements $h$, $g$ and $w$ into three product streams $y^*$, $z^*$ and $m^*$. The global optimisation scheme is iterative and optimisation is guided by the neural network in terms of an ill-defined constraint relaxation process, whereby the results obtained by the linear program models are forced to satisfy the process constraints represented by the neural net.

By using the two linear programming models (Anthony et al., 1991) sequentially (the one a subset of the other), the now of the valuable element followed by optimisation of the concentrate grade by minimising the gangue in the concentrate streams. The circuit configuration on which the mass balance equations are based, is shown in Figure 4, which illustrates the steady state flow of the valuable element (h) between two concentrator units. The flow of the gangue (g) and water (w) is similar to that of the valuable element. Both linear programming models are derived from a material balance around the general circuit model depicted in Figure 4. No explicit restrictions are specified and all constraints are derived from experimental data.

Linear programming model I

The model which is described in more detail elsewhere (Anthony et al., 1991; Reuter et al., 1988; Reuter and Van Deventer, 1990) is formulated by considering all possible process constraints

$$\sum ry^*_{ij} + \sum rz^*_{ij} + \sum rm^*_{ij}$$

**SEPARATOR BANK i**

$u^*_{h,i}$

$y^*_{h,i}$

$z^*_{h,i}$

$m^*_{h,i}$

**SEPARATOR BANK j**

$y^*_{h,j}$

$z^*_{h,j}$

$m^*_{h,j}$

$a^*_{j}$

imposed on the material conservation equations of the system shown in Figure 4.

Mass balance constraints

$$u^*_{ij} + \sum ry^*_{ij} + \sum rz^*_{ij} + \sum rm^*_{ij} = 0 \quad (16)$$

$$m^*_{ij} \cdot b_{ij} = 0$$

Separation factors for each of the three elements $k$ in the circuit.

Separation, external and recycle constraints

The separation factors used in the model are specified in terms of upper and lower bounds, for each component or element $k$ as follows:

$$ym^*_{ij} \leq \rho_m^*$$

$$ym^*_{ij} \geq \rho_m^*$$

Further constraints are reflected in the bounds of the system as specified in model I.

Objective function

The aim of the objective function of model I is to maximise the recovery of valuable elements, subject to the constraints derived from the mass balance streams, i.e.

$$\text{Max: } OBJ = a^*_1 + a^*_2 + \ldots + a^*_N$$

where $a^*_i$ represents the recovery of the valuable element $h$ from concentrator unit $i$ in the circuit.

LP model II

Model II minimises the flow of the gangue and water in the concentrate recovery streams $a^*_i$ and is constrained by the flow configuration determined by model I, i.e.

$$\text{Min: } \text{GRADE} = a^*_1 + a^*_2 + \ldots + a^*_N$$

with $k = g$ and $w$, and $N$ the number of spiral concentrators in the circuit.

The separation factors applicable to the gangue and the water are appropriately restricted to reflect the operability limits of the plant:

$$ym^*_{ij} \leq \rho_m^*$$

$$ym^*_{ij} \geq \rho_m^*$$

$$y^*_{ijk} \leq \rho_{y_{ijk}}$$

$$y^*_{ijk} \geq \rho_{y_{ijk}}$$

$$y^*_{ijk} \leq \rho_{y_{ijk}}$$

$$y^*_{ijk} \geq \rho_{y_{ijk}}$$

with $k = g$ and $w$, and $N$ the number of spiral concentrators in the circuit.

Figure 4 - Steady states of valuable element $h$ separator banks $i$ and $j$.
Realistic values are assigned to the flow variables, based on the operational limitations of the circuit.

**Neural net representation of separation process**

A back propagation neural net with an input layer with three computational elements (one for d1, d2 and d3), a hidden layer with twelve computational elements and an output layer with six computational elements (one for each ymj and znm, j = 1,2,3) is used. The net is subsequently trained with a set of exemplars of the form [d1,d2,d3,ymj,znm]. The exemplars are generated from experimental data obtained from a commercial plant, based on the assumption that the only factors influencing the separation factors are the process conditions d1, d2 and d3. The behaviour of all gravity concentrators are thus considered to be identical. In more sophisticated analyses these assumptions can be modified to take the behaviour of individual process units into account. Presentation of these data enables the net to learn the ill-defined relationship between the separation factors and the process conditions.

**Results**

The results of the optimisation of a flow circuit containing four gravity concentrator banks are shown in Figure 5. The simulated separation factors which are modelled with the neural net, satisfy the experimental data as shown in Figure 6. Discrepancies between the simulated and experimental data can be attributed to experimental errors, the influence of other less significant parameters not accounted for in the model, as well as the somewhat uneven distribution of the plant data. This modelling methodology can be applied to many other mineralogical separation processes which are difficult to describe fundamentally, such as hydrocyclone classification, heavy medium separation and flotation. As with any model, the success of the procedure depends on the accuracy of the assumptions on which the linear programming models (or other numerical computational routines) are based, as well as the availability of a large body of reliable process data.

**THE DETECTION OF GROSS ERRORS IN CONSTRAINED VARIABLE MEASUREMENTS**

The detection of gross errors in material and energy balance data constitutes an essential part of the process of reconciliation of these data, which are generally inconsistent with process constraints. Failure to detect and eliminate these types of errors (which can be the result of instrument failure or miscalibration, etc.) can lead to a severely distorted picture of the process being monitored.

**Problem statement**

Typically the process constraints or conservation equations, such as those of a mineral processing system, are described by (Crowe, 1989).

\[ A_y = 0 \quad (21) \]

where \( y \) is the \((p \times 1)\) vector of true values of the state variables and \( A \) an \((m \times p)\) constraint matrix of full row rank \( m \) (\( p > m \)). If \( y' = y + e \)

\[ y' = y + e \quad (22) \]

constitutes the \((p \times 1)\) vector of measurements of the true values \( y \), with \((p \times 1)\) error vector \( e \), then the measured values of the process variables generally violate the process constraints.

\[ A \left(y + e\right) = r \quad (23) \]

and if the constraints are linear

\[ A \left(y + e\right) = A_y + A_e = r, \quad ie \]

\[ A_e = r \quad (24) \]

Under the null hypothesis that no systematic errors are present, \( r \) is a multivariate normal with a zero mean (Madron et al., 1977; Romagnoli and Stephanopoulos, 1981; Mah and Tamhane, 1982; Tamhane and Mah, 1985), ie

\[ E(r) = E(A_e) = A_e E(e) = 0 \quad (25) \]

The alternative hypothesis is that the expected value of \( r \) is not zero, ie

\[ E(r) = b = 0 \quad (26) \]
which indicates the presence of an error with a bias of magnitude b. These two hypotheses are subsequently evaluated and rejected or accepted on the basis of standard statistical criteria.

**Gross error detection with neural nets**

A back propagation neural net is used to form an internal representation of the relationship between the distributions of measurement errors and the residuals of process constraints. This representation can subsequently be used to classify errors in the measurement data. The net consists of an input layer with one sigmoidal computational element for each measured process variable, as well as one element for each constraint residual. The output layer consists of one computational element for each measured variable, as indicated in Figure 7. Hidden layers may be used in order to characterise complex relationships between measurement and constraint residuals, but in most cases the inclusion of hidden layers has been found to be unnecessary.

**Training data**

Exemplars are constructed synthetically by corrupting a consistent set of process data with random, as well as gross errors. Typical characteristics of these errors (i.e. a bias b in the gross error and a relative variance of approximately 0.0134) are shown in Figure 8. By labelling each type of error numerically (0 for the absence of a gross error in a measurement, and one for its presence, for example), the exemplar can take the form \([y(x) \rightarrow L]\), i.e. the measurements y and the constraint residuals r are mapped to the labels L. After presentation of the appropriately scaled data, the net forms a distributed representation of the features of the different errors and their classes. By presenting the trained net with the constraint residuals of the measurement data, it is then able to classify the measurement errors associated with each variable, in terms of the numerical labels it had been trained to assign.

A comparison of the performance of a back propagation neural net (BPNN) used to detect gross errors in a set of variable measurements associated with the metallurgical grinding circuit portrayed in Figure 9 (Seth et al., 1987) and which is subject to non-linear process constraints, and that of a conventional statistical procedure, the modified iterative measurement test (MIMT) method is summarised in Table 2. The expected values of the measurements corrupted by the gross errors referred to in Table 2 are 100 per cent larger than those corrupted by small random errors. Although the modified iterative test method is designed to accommodate process systems subject to non-linear process constraints (Seth et al., 1987), it is clearly not as effective as the neural net for the detection of gross errors in the variable measurements. The results summarised in Table 2 indicate the outstanding potential of the use of neural nets in gross error detection schemes.

The reason for the superior capability of neural nets to classify errors in measurement data subject to constraints is that, unlike procedures involving Gaussian maximum likelihood techniques and other statistical methods, no assumptions with regard to the underlying error distributions are made. Neural nets are known

---

**Table 1**

Flow rates [kh] in optimised gravity separator bank (see Figure 5).

<table>
<thead>
<tr>
<th>a_1</th>
<th>a_2</th>
<th>a_3</th>
<th>a_4</th>
<th>e_1</th>
<th>e_2</th>
<th>e_3</th>
<th>e_4</th>
<th>r_1^h</th>
<th>r_2^h</th>
<th>r_3^h</th>
<th>r_4^h</th>
</tr>
</thead>
<tbody>
<tr>
<td>125.53</td>
<td>122.41</td>
<td>136.59</td>
<td>55.51</td>
<td>63.20</td>
<td>93.87</td>
<td>61.34</td>
<td>201.80</td>
<td>31.89</td>
<td>36.04</td>
<td>62.87</td>
<td>29.84</td>
</tr>
<tr>
<td>717.32</td>
<td>699.46</td>
<td>942.01</td>
<td>444.05</td>
<td>4.09</td>
<td>6.07</td>
<td>5.26</td>
<td>9.35</td>
<td>1594.25</td>
<td>1801.80</td>
<td>1343.32</td>
<td>1889.09</td>
</tr>
</tbody>
</table>

**Table 2**

Comparison of neural net with MIMT method to detect gross errors in selected variables.

<table>
<thead>
<tr>
<th>Variable Number</th>
<th>Description</th>
<th>Number of Gross Error Occurrences Detected</th>
<th>% of Gross Errors MIMT^1 BPNN^2 MIMT^1 BPNN^2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2nd smallest flow</td>
<td>10</td>
<td>22</td>
</tr>
<tr>
<td>2</td>
<td>smallest flow</td>
<td>12</td>
<td>25</td>
</tr>
<tr>
<td>3</td>
<td>largest flow</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>4</td>
<td>2nd smallest composition</td>
<td>11</td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td>largest composition</td>
<td>14</td>
<td>20</td>
</tr>
<tr>
<td>6</td>
<td>smallest composition</td>
<td>15</td>
<td>23</td>
</tr>
</tbody>
</table>

^1 Modified iterative measurement test

^2 Back propagation neural net
for their capability to model non-linear processes successfully (Lippman, 1987) and are thus ideally suited for the classification of errors with unknown or strongly non-Gaussian distributions. The MIMT method with which the net was compared, is handicapped by its assumption that the residuals of the constraints follow a Gaussian distribution, (which it tries to compensate for by testing the assumption and rejecting the results of hypothesis tests, when the assumption is found to be invalid). This method is clearly not as efficient as one in which the particular error distribution is learned and incorporated directly into the classification process, as the neural net does in effect.

The major advantage of using neural nets is that the process constraints need not be linear, as they need to be for more traditional methods. Once a net has been trained over an appropriate range of measurement values and error parameters, it can be used directly in various schemes to detect biased errors. When so desired, the net could be retrained or trained adaptively, in order to accommodate modifications in the process being monitored.

CONCLUSIONS
Over the last few years the attention of the mineral processing community has increasingly shifted towards the use of neural nets in process simulation and modelling. This rising interest is to a considerable extent attributable to a growing realisation of the limitations of traditional algorithmic procedures and heuristic methods used in the simulation of metallurgical processes. Neural

![Random Error](image1)

**OUTPUT: CLASSIFIED ERRORS**

![Input: Measurement Errors and Constraint Residuals](image2)

Fig 7 - Back propagation neural net used to identify gross errors in a process circuit.

![Frequency](image3)

**Fig 8 - Artificial errors introduced into the training and test sets of examples for error classification.**

![Metallurgical Grinding Circuit](image4)

**Fig 9 - Metallurgical grinding circuit used as a basis for comparing neural net with MIMT procedure.**

XVIII International Mineral Processing Congress
Sydney, 23 - 28 Mar. 1993

535
Neural nets hold great promise as useful and practical tools of analysis for mineral processing operations. These nets are often easier to use than conventional techniques and neural net modelling is attractive in cases where the development of models from first principles is not cost effective (Venkatasubramanian and McAvoy, 1992). Neural nets can also, for example, be used in conjunction with standard numerical procedures such as linear programming to optimise ill-defined processes, which are very difficult to simulate by conventional means, as was shown with a gravity separation circuit.

The majority of metallurgical plants are furthermore burdened with copious amounts of process data, but very little information. Neural nets such as the hybrid subspace model discussed in this paper, can provide an efficient means for extracting useful information from these plant data.

The use of neural nets that can be implemented in various computational strategies to provide a means (not equalled by current statistical methods) for detecting gross errors in measurement data subject to non-linear constraints has moreover been highlighted. These applications demonstrate the potential of neural nets as engineering tools for modelling and simulation, as well as their practical use in the mineral processing industry.

**SYMBOLS**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>constant coefficient matrix</td>
</tr>
<tr>
<td>$a_{ij}^k$</td>
<td>concentrate recovery of element $k$ from concentrator $j$</td>
</tr>
<tr>
<td>b</td>
<td>vector of biases in gross errors</td>
</tr>
<tr>
<td>$b_{ij}^k$</td>
<td>tailings recovery of element $k$ from concentrator $j$</td>
</tr>
<tr>
<td>$C_i$</td>
<td>gold concentration in liquid phase in reactor $i$</td>
</tr>
<tr>
<td>$C_{o}$</td>
<td>concentration of gold at carbon surface</td>
</tr>
<tr>
<td>$d_{ij}^l$</td>
<td>total flow rate</td>
</tr>
<tr>
<td>$d_{ij}^y$</td>
<td>dry solids flow rate</td>
</tr>
<tr>
<td>$d_y^l$</td>
<td>feed grade</td>
</tr>
<tr>
<td>$d_e$</td>
<td>particle diameter of carbon</td>
</tr>
<tr>
<td>$d_{ij}$</td>
<td>desired or target output of computational element $j$ in output layer of neural net</td>
</tr>
<tr>
<td>$E(\cdot)$</td>
<td>expected value</td>
</tr>
<tr>
<td>$e$</td>
<td>measurement error vector</td>
</tr>
<tr>
<td>$f_i$</td>
<td>kinetic reaction term</td>
</tr>
<tr>
<td>$G_i$</td>
<td>average gold grade of ore in reactor $i$</td>
</tr>
<tr>
<td>$G_{ij}^-$</td>
<td>grade of gold in ore in reactor $i$ at infinite leaching time</td>
</tr>
<tr>
<td>g</td>
<td>gangue</td>
</tr>
<tr>
<td>h</td>
<td>valuable element</td>
</tr>
<tr>
<td>$k_r$</td>
<td>adsorption coefficient</td>
</tr>
<tr>
<td>$k_l$</td>
<td>leaching coefficient</td>
</tr>
<tr>
<td>L</td>
<td>vector of labels for classification of measurement errors</td>
</tr>
<tr>
<td>$M_i$</td>
<td>mass of ore in reactor $i$</td>
</tr>
<tr>
<td>$m_{ij}^k$</td>
<td>flow rate element $k$ of tailings stream from gravity concentrator $i$</td>
</tr>
<tr>
<td>N</td>
<td>number of reactors in a process circuit or plant</td>
</tr>
<tr>
<td>n</td>
<td>thermodynamic equilibrium parameter</td>
</tr>
<tr>
<td>P</td>
<td>number of data points in a data set</td>
</tr>
<tr>
<td>$Q_i$</td>
<td>continuous slurry feed rate to CIL cascade</td>
</tr>
<tr>
<td>$Q_{j(o)}$</td>
<td>volumetric flow rate of total downflow slurry stream from unit $i$ in CIL cascade</td>
</tr>
<tr>
<td>$Q_{j(o)}$</td>
<td>upstream volumetric flow rate of carbon from reactor $i$ in a CIL cascade</td>
</tr>
<tr>
<td>$q_i$</td>
<td>average gold loading on carbon in reactor $i$</td>
</tr>
<tr>
<td>r</td>
<td>vector of process constraint residuals</td>
</tr>
<tr>
<td>$r_{p}^k$</td>
<td>recycle of element $k$ in tailings stream from concentrator $j$ to $i$</td>
</tr>
<tr>
<td>$r_{j}^k$</td>
<td>recycle of element $k$ in concentrate stream from concentrator $j$ to $i$</td>
</tr>
<tr>
<td>$r_{j}^k$</td>
<td>recycle of element $k$ in middlings stream from concentrator $j$ to $i$</td>
</tr>
<tr>
<td>t</td>
<td>time</td>
</tr>
<tr>
<td>$u_i^{x,x}$</td>
<td>flow rate of element $k$ in feed to gravity concentrator $i$</td>
</tr>
<tr>
<td>W</td>
<td>mass of carbon in reactor $i$</td>
</tr>
<tr>
<td>w</td>
<td>water</td>
</tr>
<tr>
<td>$w_{ij}(t)$</td>
<td>weight at time $t$, associated with connection from computational element $i$ to computational element $j$ in neural net</td>
</tr>
<tr>
<td>y</td>
<td>vector of observed process variables</td>
</tr>
<tr>
<td>$y_{i}^{x,x}$</td>
<td>flow rate of element $k$ concentrate stream from gravity concentrator $i$</td>
</tr>
<tr>
<td>$y^*$</td>
<td>vector of actual states of process variables</td>
</tr>
<tr>
<td>$y_i$</td>
<td>process variable $i$</td>
</tr>
<tr>
<td>$y_{m}^{x,k}$</td>
<td>lower limit of separation factor $y_{m}^{x,k}$</td>
</tr>
<tr>
<td>$y_{m}^{x,k}$</td>
<td>upper limit of separation factor $y_{m}^{x,k}$</td>
</tr>
<tr>
<td>$y_{n}^{x,k}$</td>
<td>concentrate-tailings separation factor for element $k$ in concentrator $j$</td>
</tr>
<tr>
<td>$x_{i}^{j,k}$</td>
<td>flow rate of element $k$ middlings stream from gravity concentrator $i$</td>
</tr>
<tr>
<td>$z(i)$</td>
<td>state of computational element $i$ at time $t$ in neural net</td>
</tr>
<tr>
<td>$z_{m}^{x,k}$</td>
<td>lower limit of separation factor $z_{m}^{x,k}$</td>
</tr>
<tr>
<td>$z_{m}^{x,k}$</td>
<td>upper limit of separation factor $z_{m}^{x,k}$</td>
</tr>
<tr>
<td>$z_{m}^{x,k}$</td>
<td>middlings-tailings separation factor for element $k$ in concentrator $j$</td>
</tr>
<tr>
<td>$z_{o}^{j}$</td>
<td>actual output of computational element $j$ in output layer of neural net</td>
</tr>
</tbody>
</table>

**GREEK LETTERS**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e_1$</td>
<td>void fraction of carbon</td>
</tr>
<tr>
<td>$e_i$</td>
<td>void fraction of liquid</td>
</tr>
<tr>
<td>$e_o$</td>
<td>void fraction of ore</td>
</tr>
<tr>
<td>$e_\theta$</td>
<td>error criterion</td>
</tr>
<tr>
<td>$\theta$</td>
<td>bias of computational element $i$ in neural net</td>
</tr>
<tr>
<td>$\Phi$</td>
<td>transfer function</td>
</tr>
<tr>
<td>$\rho$</td>
<td>density</td>
</tr>
<tr>
<td>$\rho$</td>
<td>density</td>
</tr>
<tr>
<td>$\tau$</td>
<td>learning rate coefficient</td>
</tr>
</tbody>
</table>
REFERENCES


ARTIFICIAL INTELLIGENCE METHODS FOR PROCESS MONITORING AND MODELLING OF METALLURGICAL PLANTS

C Aldrich* & J S J van Deventer

Department of Metallurgical Engineering, University of Stellenbosch, Private Bag X5018, Stellenbosch, 7599, South Africa.
Fax 27-21-8082059.

*Author to whom all correspondence should be addressed.

ABSTRACT

Although artificial neural networks have become established as useful instruments for the modelling of complex processes in the chemical and metallurgical industries, little has hitherto been published with regard to their use on actual plant data. In this paper connectionist models of a gold reduction and a phosphate flotation plant are shown to afford better representations than regression models presently in use on the plants. It is furthermore shown that neural nets can be used to detect systematic changes in the behaviour of industrial plants, regardless of the complexity of the plant models or the stochastic processes underlying the behaviour of the plants.

INTRODUCTION

The majority of chemical and mineral processing plants are burdened with copious amounts of process data, which makes it difficult to identify the essential features of the processes involved in plant operations. The development of process models based on these data is often not cost effective and the data are usually analyzed by means of multiple linear or non-linear regression techniques. Since these techniques require explicit process models, they are not always suitable for modelling of the complex behaviour that industrial plants so often exhibit. In contrast, neural nets do not suffer from this drawback and (provided they are presented with enough representative data) constitute an efficient means for the construction of implicit models of ill-defined processes. In spite of these well-known attributes, very little has been published in the chemical engineering literature with regard to the use of neural nets in this way. In this paper feedforward neural nets with sigmoidal and hyperbolic tangent process units in one or more hidden layers are used for the prediction of gold losses on a gold reduction plant, as well as the consumption of various additives on a phosphate flotation plant. The residuals generated by these (and other) models are moreover shown to constitute a general basis for the detection of systematic changes in the behaviour of chemical and metallurgical plants, regardless of the distributions of the residuals.

ARTIFICIAL NEURAL NETS

A back propagation neural network consists of a large number of massively interconnected processing elements which are typically arranged in at least an input and an output layer, and possibly one or more hidden layers separating the input and output layers. Connections between the elements in the various layers are characterized by adjustable numeric values or weights which can in principle assume values enabling inputs to the net to be mapped to functionally related outputs to an arbitrary degree of accuracy [1].

Computation in back propagation neural nets is feedforward and synchronous, i.e. the states of the process units in lower levels or layers of the net are updated before units in layers further down in the net. The activation rules of process units are typically of the form

\[ z_i(t+1) = \varphi(u_i(t)) \]  

(1)

where \( u_i(t) \) designates the potential of a process unit at time \( t \), i.e. the difference between the weighted sum of all the inputs to the unit and the unit bias

\[ u_i(t) = \sum_j w_{ij} z_j(t) - \Theta_i \]  

(2)
The training of back propagation neural nets is an iterative process involving the changing of the weights of the net, typically by means of a gradient descent method, in order to minimize an error criterion, that is

$$ w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij} $$  \hspace{1cm} (3) $$

where

$$ \Delta w_{ij} = -\tau \varepsilon \frac{\partial E}{\partial w_{ij}} $$  \hspace{1cm} (4) $$

and where \( \tau \) is the learning rate and \( \varepsilon \) the error criterion, i.e.

$$ \varepsilon = \frac{1}{2} \sum (d_{kj} - z_{kj})^2 $$  \hspace{1cm} (5) $$

based on the difference between the desired and the actual outputs of the unit. The change in the error with changes in the weights of the net can be expressed as

$$ \frac{\partial E}{\partial w_{ij}} = z_i(t)z_j(t)[1-z_j(t)]f_k(t) $$

for output units and

$$ \frac{\partial E}{\partial w_{ij}} = \sum_k w_{jk}(z_k(t)[1-z_k(t)])f_k $$

for hidden units.

**CONNECTIONIST PLANT MODELS**

Assuming the process system to be modelled to be acyclic, the problem concerned with the construction of a circuit or plant model is to relate the matrix \( Y \) to some function of matrix \( X \), in order to predict \( Y \) from \( X \), where \( y_{ik} \) (\( i = 1, 2, \ldots, p \) represent \( p \) variables, dependent on \( m \) causal or independent variables \( x_{jk} \) (\( j = 1, 2, \ldots, m \)), based on \( n \) observations (\( k = 1, 2, \ldots, n \)). The variables \( y_{ik} \) are usually parameters which provide a measure of the performance of the plant, while the \( x_{jk} \) variables are the plant parameters on which these performance variables are thought to depend.

The simplest approach, and a method often used on mineral processing plants, is to assume a linear relationship between \( X \) and \( Y \), i.e.

$$ Y = Xk_1 + k_2 $$

and to find the coefficient vectors \( k_1 \) and \( k_2 \) by ordinary least squares methods, that is

$$ k_1 = (X^TX)^{-1}X^TY \quad \text{and} \quad k_2 = Y - Xk_1, $$

provided that the elements of the columns \( X_j \) of matrix \( X \) are not correlated and that the number of observations is larger than the number of coefficients that has to be estimated (i.e. \( n > m \)). If not, other techniques, such as partial least square methods [2] can be used to obviate the problem. Should the assumption of multilinear relationships between the variables prove to be inadequate, they can be extended by the addition of suitable non-linear terms [3], the incorporation of spline methods [4], or replaced by non-linear regression methods [5].

The main advantage of modelling techniques based on the use of neural nets, is that a priori assumptions with regard to the functional relationship between \( x \) and \( y \) are not required. The net learns this relationship instead, on the basis of examples of related \( x-y \) vector pairs or exemplars. The following examples illustrate the construction of connectionist plant models.

**MODELLING OF LOSSES ON A GOLD REDUCTION PLANT**

The efficiency of gold reduction plants is often interpreted in terms of the gold lost during the recovery process, since the recovery of gold (which commonly exceeds 97%) is too insensitive a parameter to use [5],[6]. It is convenient to distinguish between gold losses in a dissolved form, as well as losses in solid residues. These losses cannot be explained in terms of a fundamental model of the plant and are often predicted in practice by means of
linear regression models. These models relate the dissolved gold losses ($y_1$) and the undissolved gold losses ($y_2$) to a number of empirical parameters, namely the head grade of the ore ($x_1$), residual grade of the ore ($x_2$), solution tonnage ($x_3$), treated tonnage ($x_4$), filter feed rate ($x_5$), filter wash ($x_6$), solids duty ($x_7$), filter ARLA ($x_8$), solution duty ($x_9$), entering solution ($x_{10}$), filter flocculation ($x_{11}$), filter vacuum ($x_{12}$), sodium cyanide agitator I ($x_{13}$), and sodium cyanide agitator II ($x_{14}$).

A back propagation net with an input layer and one hidden layer, both comprised of fourteen processing elements or artificial neurons, and an output layer comprised of two processing elements was used to model the gold losses. The fourteen elements in the input layer corresponded to the fourteen input parameters used to correlate the gold losses ($x_1, x_2, \ldots, x_{14}$), while the number of elements in the hidden layer was chosen arbitrarily. The input layer did not process the data, but merely served to distribute the data to the hidden layer. The output of the two processing elements in the output layer corresponded with the predicted values of the two output variables, namely the dissolved ($y_1$) and undissolved gold loss ($y_2$).

The layers were connected in a feedforward manner, i.e. no layer was connected to any layer preceding it, and all layers consisted of elements with hyperbolic tangent translation functions, to ensure that low-valued and high-valued outputs were treated equally, i.e. $\phi(u) = (e^u - e^{-u})/(e^u + e^{-u})$. The output of the net after training with the generalized delta rule [7],[8] is compared with the predicted outputs based on a linear regression analysis used on the plant. The results are depicted graphically in figures 3 and 4. Based on the root mean square values of the correlation errors, the nets performed significantly better than the existing plant models (approximately 51% for the undissolved gold losses and 87% for the dissolved gold losses).

**MODELLING OF RECOVERY AND REAGENT CONSUMPTION ON A PHOSPHATE FLOTATION PLANT**

The ore feed to a phosphate flotation plant is analyzed hourly and these data, as well as those representing other parameters in the plant are averaged on a daily basis and used to predict the consumption of three reagents in the plant, viz. water glass ($y_1$), polyglycol ether ($y_2$) and fatty acid ($y_3$). The water glass or sodium silicate serves as a dispersant and depresses diopside, iron silicates and olivine. The fatty acid acts as a collector for apatite and contributes to the frothing characteristics of the flotation cells, while the neural nets during training (weight adjustment of the nets), while the test set was used to monitor the performance of the nets subsequent to training. This procedure is essential to ensure that the net generalizes the relationships between parameters correctly, instead of just learning to reproduce the data presented to it.
polyglycol ether (nonyl phenol tetruglycol ether) is a non-ionic surfactant and emulsifier, serving as a froth modifier and a depressant for iron minerals and calcite. These reagents are expensive (totalling approximately 87% of the direct operating costs of the plant) and inadequate control of their consumption can have a major impact on plant economics [9]. More specifically, the variables \( y_1, y_2 \) and \( y_3 \) are related to the mass fractions of apatite \( x_1 \), phosphogip (\( x_2 \)), lizardite \( x_3 \), magnetite \( x_4 \), diopside \( x_5 \), calcite \( x_6 \), dolomite \( x_7 \) and forsterite \( x_8 \) in the feed, as well as the feed pulp density \( x_9 \), feed flow rate \( x_{10} \), the phosphate \( P_2O_5 \) concentration in the feed \( x_{11} \), iron content of the feed \( x_{12} \) and the tailings \( x_{13} \) and concentrate \( x_{14} \) flow rates.

The data used in the investigation consisted of 438 sets of \( \{x_1, x_2, \ldots, x_{14}, y_1, y_2, y_3\} \) vectors, which were subdivided into a training set consisting of 408 vectors, and a test set consisting of 30 vectors. In contrast with the gold plant models, the training and test sets were time sequential, so that the performance of the neural nets on the test sets can be regarded as a measure of the ability of these nets to extrapolate historic trends in the behaviour of the plant. The nets used to model the consumption of the different additives were all back propagation neural nets with sigmoidal processing elements, and as before the delta rule [7] was used to train the nets.

**Water glass consumption \( y_1 \)**

A single hidden layer consisting of six hidden units was used between the input and the output layers of the net. The input layer was fully connected to both the hidden layer and the output layer of the net.

Instead of training the net to a certain output error tolerance on the training exemplars, use was made of a cross validation method in which the performance of the net was periodically checked against the test data set, until improvement in the performance of the net became marginal. Note that this approach has no effect on the adjustment of the weights of the net during training, but merely serves as a guide to an appropriate neural net structure.

**Polyglycol ether \( y_2 \) and fatty acid \( y_3 \) consumption**

The nets used to model the polyglycol ether and fatty acid consumption had identical configurations and were comprised of two hidden layers each. The first hidden layer consisted of six hidden elements, while the second had three. The input layer was fully connected to the first hidden layer only, while the first hidden layer was fully connected to both the second hidden layer and the output element. As before, a cross validation technique was also used to determine the convergence of the nets.

![Figure 5](image1.png)  
Run numbers are time sequential

**Figure 5** Prediction of water glass consumption \( y_1 \)

![Figure 6](image2.png)  
Run numbers are time sequential

**Figure 6** Prediction of polyglycol ether \( y_2 \)

The ability of the nets to generalize the trends in the data is depicted in figures 5 to 7, where the predictions corresponding to the test data sets are highlighted. As can be seen from these results the nets were able to generalize the data better than the regression models.

**DETECTION OF SYSTEMATIC CHANGES IN PLANT BEHAVIOUR**

Once an accurate connectionist or other model of the plant or process is developed, systematic changes...
in the behaviour of the plant can be monitored by periodically comparing plant data with the output of the plant model. This is accomplished through analysis of the distribution of the residuals generated when actual plant data are compared with the data predicted by the neural net. If no significant changes occur in the system, a relatively small residual with magnitude dependent on random measurement and modelling errors can be expected. If the system changes systematically however, the magnitude of the residuals would also be affected by a bias in the expected values of the dependent variables.

These changes can generally not be tested statistically, since the distributions of the residuals are unknown. Explicit knowledge of the distribution of the residuals in the variables is not a prerequisite to the use of neural net methods however, since the nets can learn the distributions a priori.

To use a neural net to detect the presence of these biases in the data, a set of exemplars is first constructed by corrupting a representative subset of the set of dependent data \( Y \) from which the plant model \( Y = f(X) \) was derived to generate residuals and to label the residuals accordingly. This set of training exemplars generally consists of \( k \) training vectors \( V_k \), each of which can be partitioned into a subvector \( V_k^{\text{IN}} \) containing all the inputs to the net, and a sub-vector \( V_k^{\text{OUT}} \) which contains the desired outputs associated with the corresponding inputs, i.e. 

\[
V_k = [V_k^{\text{IN}}, V_k^{\text{OUT}}]
\]

The inputs to the net generally consist of some function of the residuals of the system, while the outputs consist of some label to identify the type or class of error. That is

\[
v_i^{\text{IN}} = f(y_{\text{pred}} - y_{\text{exp}}) \quad (i = 1, 2 \ldots N) \quad (6)
\]

where

\[
f(y_{\text{pred}} - y_{\text{exp}}) = \frac{y_{\text{pred}} - y_{\text{exp}}}{\sigma}
\]

and where only two events need to be recognized

\[
v_i^{\text{OUT}} = 0, \quad \text{if } v_i^{\text{IN}} \text{ is associated with no change in the process } (i = 1, 2 \ldots N), \text{ or}
\]

\[
v_i^{\text{OUT}} = 1, \quad \text{if } v_i^{\text{IN}} \text{ is associated with a systematic change in the process } (i = 1, 2 \ldots N)
\]

Since these outputs are binary vectors, the output of continuous-valued neural nets (such as back propagation neural nets) have to be postprocessed prior to interpretation. In this investigation continuous-valued outputs were rounded off to the nearest binary class indicator, i.e.

IF \(|z_{0,i} - v_i^{\text{OUT}}| < 0.5,
\]

THEN \(z_{0,i} = v_i^{\text{OUT}}
\]

Example

The previously discussed connectionist model of the phosphate and gold reduction plants, and a non-linear regression model of a uranium leach plant [10] as well as the plant data on which these models were based.
were used to construct sets of training exemplars as explained above.

The output variables of these models were corrupted by various biases and in each case a simple sigmoidal back propagation neural net with no hidden layers was trained to classify the normalized magnitudes of the residuals into two classes, depending on whether the residuals were associated with a bias in the process or not.

As shown in Figure 8, where the ability of these nets to correctly classify biased residuals (subject to less than 2% misclassification of unbiased residuals) is depicted, the nets were able to detect process shifts of more than four standard deviations with an accuracy of more than 80%.

![Graph showing the ability of neural nets to detect systematic changes in plant behaviour](image)

**Figure 8** Ability of neural net to detect systematic changes in plant behaviour

**CONCLUSIONS**

- Based on an analysis of industrial plant data, it can be concluded that connectionist systems provide a convenient means of constructing ill-defined plant models.

- Simple sigmoidal back propagation neural nets can be used to model probability distributions of plant model residuals through perturbation, and

- As a result these nets constitute a general means of identifying systematic changes in the behaviour of chemical and metallurgical processes that cannot be detected readily by other methods.

**REFERENCES**


NOMENCLATURE

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>d_{o,j}</td>
<td>desired output value of j'th process element in output layer of artificial neural net</td>
</tr>
<tr>
<td>k_{1,k2}</td>
<td>vectors of constant parameters</td>
</tr>
<tr>
<td>u_i(t)</td>
<td>potential of i'th process unit</td>
</tr>
<tr>
<td>v_i^{OUT}</td>
<td>i'th feature of an exemplar</td>
</tr>
<tr>
<td>v_i^{IN}</td>
<td>i'th class of an exemplar</td>
</tr>
<tr>
<td>v_k^{OUT}</td>
<td>k'th exemplar</td>
</tr>
<tr>
<td>v_k^{IN}</td>
<td>features of k'th exemplar</td>
</tr>
<tr>
<td>w_{ij}</td>
<td>connection weight between process elements i and j in a neural net</td>
</tr>
<tr>
<td>x</td>
<td>a set of independent variables</td>
</tr>
<tr>
<td>X</td>
<td>an array of independent variable sets</td>
</tr>
<tr>
<td>x_{i,j}</td>
<td>j'th observation of the i'th independent variable</td>
</tr>
<tr>
<td>X_j</td>
<td>j'th column of matrix X</td>
</tr>
<tr>
<td>\phi</td>
<td>transfer function of a neural net</td>
</tr>
<tr>
<td>y</td>
<td>a set of dependent variables</td>
</tr>
<tr>
<td>Y</td>
<td>an array of dependent variable sets</td>
</tr>
<tr>
<td>Y_{exp}</td>
<td>observed value of a dependent variable</td>
</tr>
<tr>
<td>Y_{ij}</td>
<td>j'th observation of the i'th dependent variable</td>
</tr>
<tr>
<td>Y_{pred}</td>
<td>predicted value of a dependent variable</td>
</tr>
<tr>
<td>z_i(t)</td>
<td>the state of the i'th process element in a neural net</td>
</tr>
<tr>
<td>z_{o,j}</td>
<td>the state of the j'th process element in the output layer of a neural net</td>
</tr>
<tr>
<td>\beta_j(t)</td>
<td>error criterion propagated back through net during learning</td>
</tr>
<tr>
<td>\epsilon</td>
<td>an error criterion</td>
</tr>
<tr>
<td>\Theta_j</td>
<td>bias of j'th process element in neural net</td>
</tr>
<tr>
<td>\sigma</td>
<td>standard deviation of plant model residuals</td>
</tr>
<tr>
<td>\tau</td>
<td>a learning parameter</td>
</tr>
</tbody>
</table>
IPMM'97

AUSTRALIASIA-PACIFIC FORUM

ON

INTELLIGENT PROCESSING & MANUFACTURING OF MATERIALS

Volume 1: Intelligent Systems Applications

Editors

T. Chandra, S.R. Leclair, J.A. Meech

B. Verma, M. Smith, B. Balachandran
IDENTIFICATION OF HYDROMETALLURGICAL PROCESSES USING NEURAL NETWORKS

J.S.J van Deventer

Department of Chemical Engineering, The University of Melbourne, Parkville, Victoria 3052, Australia. E-mail: jsj.van_deventer@chemeng.unimelb.edu.au

ABSTRACT

Some hydrometallurgical problems cannot be formulated using conventional parametric mathematical models owing to a lack of phenomenological understanding. Neural networks provide one way of mapping the relations between process variables and functions for such ill-defined problems. A carbon-in-leach process is used as a case study to show how simple neural models could be formulated for sub-processes and hence sub-spaces within the more general data space, provided that such sub-processes could be estimated explicitly using a dynamic material balance equation. In the elution of gold it is more difficult to estimate such sub-processes from dynamic process data. It will be explained briefly how a neural net could be used to relate the equilibrium loading on the carbon in terms of the process conditions of elution. The problem is often that a trained neural net is required in the solution of a system of differential equations in order to describe system dynamics, while training data are not available prior to solution of the differential equations. This means that it is impossible to use standard neural nets in such an inverse problem.

INTRODUCTION

Many hydrometallurgical processes are complex and poorly understood, so that reliable phenomenological models are not available. Moreover, few of the existing fundamental models have been implemented for design or on-line control decisions in industry. The design, monitoring and control of many plants are consequently often conducted on an informal heuristic basis, where plant operators attempt to maintain optimal operating conditions based on their experience of the
behaviour of the plant. As a result, these systems are usually controlled suboptimally owing to human error, inadequate training and lack of experience. It is often difficult to formulate an exact mathematical expression for a process owing to its ill-definedness. A non-parametric method such as a neural network could be used instead to relate input and output variables. The main advantage is that no functional form needs to be specified a priori, and even semi-quantitative data could be included as inputs.

Neural nets have been applied recently to a variety of mineral processing operations, such as carbon-in-pulp plants [1,2], elution of carbon [3], the liberation of gold [4], pyrometallurgical applications [5,6], the detection of gross errors in plants [7], the modelling of rare earth solvent extraction [8], and the characterisation of flotation froths [9,10]. The literature on neural nets is substantial [11], even in materials processing, so that no basic concepts will be explained below. Excellent neural network packages are available commercially which conduct scaling and training in a user-friendly manner. A neural net consists of simple computational elements called neurons or processing elements which are interconnected, and the collective behaviour of these neurons determines the characteristics of the net. Of the numerous network architectures developed to date the Sigmoidal Backpropagation Neural Net (SBNN) remains the most widely used for the modelling of ill-defined process systems.

In this paper a neural network technique called the hybrid subspace method is combined with a set of dynamic equations in order to simulate a carbon-in-leach process on the basis of sparse data. The incorporation of a neural net in a set of differential equations is also described briefly with a view to model the elution of gold from activated carbon. It will be explained that new theory needs to be developed for the simultaneous training of a neural net embedded in a set of differential equations.

HYBRID SUBSPACE MODELLING

Hybrid subspace modelling [1,2] is a neural network methodology based upon the elimination of unnecessary ill-defined dimensionality through local variable selection and the identification of simple mathematical relations between the predictor and response variables. Relying on the fact that SBNN's with a relatively small number of free model parameters approximate the geometric form of the function adequately, SBNN's are used both during the model building process and as basis functions for the ill-defined parts of the final hybrid subspace model.

The procedure used to construct a hybrid subspace model is conducted through a number of steps. In the first instance a global SBNN is trained with all process data points. This global network is used to divide the process variable space into subspaces. This division is done by performing a perturbation analysis on the training data set, whereafter the perturbation results are used to identify less significant variables within every individual subspace. A relative perturbation value (rpv) is calculated for each variable in order to give its relative importance throughout the variable space [1,2]. The perturbation results can now be used to identify boundaries between subspaces on the basis of eliminated different less-significant variables from neighbouring subspaces. A "cutoff" value for each rpv can be specified in order to divide the global variable space into subregions. A
A subregional boundary is identified within the variable space at the points where the \( r_{pv} \) of a specific dimension becomes smaller than the \( r_{pv} - \text{cutoff} \).

During the next step an SBNN is trained for each subspace on data points containing only the values for the significant variables within a specific subspace. Such SBNN "basis functions" have few input nodes and should therefore be able to perform improved curve fittings due to a reduction in the number of local model parameters. Hence, the generalisation of interpolation properties of the SBNN's for the different subspaces should be better as well. The SBNN model of each subspace is now employed to identify simple mathematical relations between a function and certain variables within a subspace. The remaining dimensions with unknown relations to the function can now be mapped during a further step by an even simpler SBNN. The result will be an empirical model consisting of a combination of phenomenological expressions and SBNN basis functions for the different subspaces.

**DYNAMIC SIMULATION OF THE CIL-PROCESS USING HYBRID SUBSPACES**

The hybrid subspace modelling strategy proposed above is illustrated using the carbon-in-leach (CIL) process for gold recovery as an example. The dynamic reactor model used here is similar to that described elsewhere [12], except that one component only (gold) is considered. For a single-component CIL-process, three kinetic reaction terms can be used in the material balance equations to describe the processes of mass transfer due to adsorption and leaching within the three relevant phases (i.e. \( f_c \) - carbon phase, \( f_o \) - ore phase and \( f_l \) - liquid phase). The net mass transfer \( f_i \) in the liquid phase over a single stage \( i \) in a CIL cascade is

\[ f_i = V \frac{dC}{dt} = Q C_{i+1} - Q C_i - f_c + f_o \]

subjected to a certain equilibrium relationship. \( V \) is the volume of liquid in a stage, \( C \) is the liquid phase concentration, \( Q \) is the volumetric flowrate of liquid and \( t \) is the time variable. Similar dynamic material balance equations could be formulated for the ore phase and carbon phase, which should be dependent on the mode of countercurrent transfer of carbon between stages.

If \( f_i, f_c \), and \( f_o \) are known under specific process conditions, the differential material balance equations of the reactor model can be solved by numerical techniques. These three reaction terms can be expressed in numerous ways through models constructed by means of different modelling techniques. If an adequate model for the process kinetics is available, values for \( f_i, f_c \), and \( f_o \) can be determined at each point in time during a simulation run. It is possible to obtain plant data where either \( f_c \) or \( f_o \) is dominant, so that it is possible to back-calculate \( f_i, f_c \) and \( f_o \) from plant profiles of \( C \), \( q \) (the loading on the activated carbon) and \( C \) (the grade of the ore) using discretised dynamic equations [1] such as Eq. (1). The values for \( M \) (mass of ore in a specific stage) and \( W \) (mass of carbon in a specific stage) are also known.

The relationship between \( f_i, f_c \) or \( f_o \) and the process conditions is described by a phenomenological model if adequate knowledge exists about the rate-controlling mechanisms. Usually this is not the case, and it is for this reason that a hybrid subspace neural net is used to capture such relationships. In effect the neural net models relate the values of each of \( f_i, f_c \), or \( f_o \) to the process conditions within a specific region. A modified first order rate equation can also be used to capture this relationship,
where the rate "coefficient" is not a constant but a time-independent neural net function of the process conditions at that stage [13]. This generalised neural net kinetic model has been shown to be useful in the simulation of a wide range of ill-defined hydrometallurgical and pyrometallurgical rate phenomena. As will be explained in the last section of this paper, this approach can be used only when the values of functions such as $f_i$, $f_c$, or $f_0$ can be estimated explicitly from dynamic profiles. When that is not the case, it is not possible to train the neural nets independent of the solution of the differential equations.

In total 800 data points with values of $M$ between 400-500 tonnes, $W$ between 12-22 tonnes and $C$ between 0.001 and 5 p.p.m. were used for training, while 400 data points were used as a test set. It was estimated that the data contained 15% Gaussian noise. The data values of $f_i$, $f_c$, $f_0$, $C$, $q$ and $G$ were scaled logarithmically, while $M$ and $W$ were scaled linearly so that all scaled values were transformed into the range -3.5 to 3.5. A global SBNN consisting of 10 hidden nodes was trained, which in turn was used to perform the perturbation analysis. Since it is impossible to plot the $rpv$'s of all variables in a five-dimensional variable space, it will be unsuitable to show a complete set of graphs for the $rpv$'s within single dimensions. Only the $rpv$'s of the reaction function $f_c$ as a function of $C$ (ignoring all other variables) are therefore shown in Figure 1. Although it should be possible to determine an $rpv$-cutoff in an automated way, a suitable cutoff ($rpv < 10\%$) was selected arbitrarily in order to illustrate this technique.

![Figure 1](image)

**Figure 1.** $Rpv$ perturbation results for the reaction kinetics function $f_c$ of the CIL-process [2].

Evidently the reaction term $f_c$ is significantly affected by both $C$ and $W$ throughout the variable space. Also notable is the strong effect of $q$ on $f_c$ within the high $C$-region and the drop in significance of $q$ in the region of lower $C$-values. The $rpv$'s for $G$ and $M$ remain relatively small through the predictor variable space, so that these two variables can be eliminated as variables for $f_c$. 

---

264
These observations are in agreement with predictions of a phenomenological model. Similar results were found from rpv-plots for \( f_i \) and \( f_0 \). As expected, all five variables are significant predictors of \( f_i \). A strong effect of \( q \) on \( f_i \) within the high \( C \)-region and the drop of significance of \( q \) towards the lower \( C \)-values enabled the division of the variable space of \( f_i \) into two subspaces at approximately \( C = 0.2 \text{ p.p.m.} \). The perturbation results of \( f_0 \) confirmed its dependence on \( G \) and \( M \). The variable spaces of \( f_i \) and \( f_0 \) have now been subdivided into two subspaces each. The first subspace of \( f_i \) (for \( C > 0.2 \text{ p.p.m.} \)) is five-dimensional, while the dimensionality of the second subspace (\( C < 0.2 \text{ p.p.m.} \)) has been reduced to four. In the case of \( f_0 \), its first subspace contains three dimensions (\( C, q \) and \( W \)), which is adjacent to the second subspace with dimensions for \( C \) and \( W \) only. The undivided variable space for \( f_0 \) has two dimensions (\( G \) and \( M \)).

During the following step, different simplified SBNN's were trained for the adjacent subspaces of each function. These subspace networks (two for \( f_i \), two for \( f_0 \) and a single one for \( f_6 \)) were used to identify simple mathematical relations between the functions and their variables [1,2]:

\[
f_i = W \cdot r_2 \quad \text{where} \quad r_2 = \beta(C,q) \text{ if } C > 0.2 \text{ p.p.m.} \quad (2)
\]

\[
r_2 = \beta(C) \text{ if } C < 0.2 \text{ p.p.m.}
\]

\[
f_0 = M \cdot r_3 \quad \text{where} \quad r_3 = \alpha(G) \quad (3)
\]

\[
f_i = M \cdot r_3 - W \cdot r_2 \quad (4)
\]

The nonparametric nature of the hybrid subspace model is further reduced in the sense that \( r_2 \) and \( r_3 \) are the only nonparametric (neural net) subdivisions of the modelling problem. The dependencies of \( f_i, f_2 \) and \( f_6 \) on \( M \) and \( W \) are similar to those used in a phenomenological model [2,12]. As shown in Figure 2, a hybrid subspace model containing five SBNN's (two nets for each \( C \)-interval for \( r_2 \) and a net for \( r_3 \)) is more accurate than a five-dimensional SBNN model and approximates a phenomenological model.

Figure 2. Gold concentration profiles in reactor 1 of the CIL-cascade as predicted by the CIL-simulator [12] and two neural net models [2].
NEURAL NET MODELLING OF GOLD CYANIDE ELUTION

With the assumption of equilibrium throughout an elution column under aggressive pre-soaking conditions, a gold balance is written over the j'th section from the inlet of a column consisting of N sections of height Δh each:

\[ S \Delta h p_c (1 - \varepsilon) \frac{dq}{dt} + S \Delta h \{ p_c V_p (1 - \varepsilon) + \varepsilon \} \frac{dC}{dt} = V(C^{j+1} - C) \]  

(5)

where all the variables refer to the j'th section, except for the concentration \( C^{j+1} \) which refers to the previous section. \( S \) is the flow area of the column, \( V \) is the volumetric flow rate, \( \varepsilon \) is the void fraction in the carbon bed, \( p_c \) is the apparent density of the carbon, \( V_p \) is the specific pore volume of the carbon, \( q \) is the equilibrium loading of gold on the carbon, \( t \) is the time variable and \( h \) is the height variable.

It is often difficult to formulate an explicit functional form for the complex relationship between \( q \) and process conditions. Therefore, a non-parametric method such as a neural network should be used instead to relate \( q \) to all the possible variables such as solution phase concentrations, temperature, pre-soaking conditions and the loading history of the activated carbon. Especially in multi-component elution it becomes very difficult to estimate independently the dependence of the various metals on process conditions; when neural nets are used this step-wise estimation of parameters becomes unnecessary. The 10 input nodes in the neural nets used here are related to the solution phase concentrations \( C_0 \) and the initial loadings \( q_{i,0} \) of elements \( i \), i.e. gold, silver, copper and nickel, the concentration of potassium in solution \( C_K \), and the loading of cyanide decomposed or adsorbed on the carbon \( q_N \). The four output nodes refer to the loadings \( q_i \) of elements \( i \), i.e. gold, silver, copper and nickel. A training set of 850 vectors and a test set of 200 vectors were used. Three or four hidden nodes were required to give an adequate representation of the data. Hyperbolic tangent processing elements were used, so that low valued and high valued loadings were treated equally. A net having the same input and hidden nodes but only a single output node yielded similar results, but was slightly easier to train. As a result of the non-linearity of the equilibrium relationships, these nets required at least 50 000 iterations to stabilise and were therefore relatively slow to train. A hybrid subspace model could also have been used. Satisfactory agreement was obtained between measured and predicted gold loadings on the carbon.

THE INVERSE PROBLEM IN DYNAMIC MODELLING

Usually only concentration \( C \) versus time \( t \) data are available to estimate parameters, or to train a neural net embedded in a system of differential equations such as Eqs. (1) and (5). The problem is that the \( C - q \) equilibrium relationship to be simulated using a neural net is often unknown, but indeed required to solve the differential equations. In Eq. (1) the data required to train such a neural net can be back-calculated from discretised dynamic data, but in the case of Eq. (5) that is not possible. In fact, the trained neural net is required to solve the host differential equations, and vice versa, the neural net cannot be trained independently from the solution of the differential equations. This means that an inverse problem exists which has not been addressed in the literature on neural nets. It could have been argued that it is easier to apply a single neural net to the entire set of \( C - t \) data without formulation of a set of differential equations, but this would
unnecessarily increase both the dimensionality and the non-linearity of the neural net. It has been attempted to incorporate the differential equations into the architecture of a modified recurrent net and then to solve this inverse problem. Nevertheless, this has not been found to be a viable option owing to the complexity of training such a recurrent net. If this problem could be solved, it will allow the identification of complex hydrometallurgical processes using neural nets within the structure of more fundamental equations.

CONCLUSIONS AND SIGNIFICANCE

The CIL process and the elution of gold in a column of activated carbon were used as case studies to demonstrate how neural nets could be incorporated in an overall dynamic material balance. In the CIL model dimensionality was reduced by defining a hybrid subspace model, which consists of mathematical relations and SBNN models for adjacent variable subspaces with different combinations of dimensionality. This model showed a substantial improvement in its generalisation properties over its SBNN counterpart of full dimensionality. In contrast with the CIL process, it is more difficult to estimate sub-processes from dynamic process data in the case of gold elution. The problem is often that a trained neural net is required in the solution of a system of differential equations in order to describe plant dynamics, while training data are not available prior to solution of the differential equations. This means that it is not feasible to train standard neural nets in such an inverse problem. Nevertheless, this paper shows that neural nets, if applied innovatively, can be used as tools for the modelling of ill-defined systems in hydrometallurgical processes.

REFERENCES

<table>
<thead>
<tr>
<th>Session</th>
<th>Topic</th>
</tr>
</thead>
</table>
| 0       | Final Program  
List of Participants  
General Information |
| 1       | SESSION 1:  
Grinding and Separation |
| 2       | SESSION 2:  
Iron- and Steelmaking |
| 3       | SESSION 3:  
Continuous Casting and Rolling |
| 4       | SESSION 4:  
General Applications |
| 5       | SESSION 5:  
New Methods |
| 6       | Invited Lectures |
THE SIMULATION OF ILL-DEFINED METALLURGICAL PROCESSES USING A NEURAL NET TRAINING PROGRAM BASED ON CONJUGATE-GRADIENT OPTIMIZATION

T. J. van der Walt, J. S. J. van Deventer

Department of Metallurgical Engineering, University of Stellenbosch, Stellenbosch, 7600, South Africa

E. Barnard

Computer and Electronics Engineering Department, University of Pretoria, Pretoria, 0001, South Africa

Abstract. Most metallurgical processes are ill-defined in some way owing to their complex natures. This causes difficulty to construct adequate phenomenological models for such processes. It is shown in this paper how neural nets (NN's) can be used to model ill-defined reaction systems by using a typical metallurgical process, such as carbon-in-leach (CIL).

Multi-dimensionality (a high number of system variables), which causes problems for modelling, is investigated. It is then illustrated how a NN trained with noisy data can be employed to reduce dimensionality by following two approaches. Firstly, the influences of the system variables are quantified with a perturbation analysis technique. Less significant variables are then eliminated from a specific part of the operating domain. Secondly, the NN is used successfully to identify mathematical relations between system variables and functions. If a relation is known, it can be expressed mathematically. The ill-defined process domain thus shrinks further. Eventually the non-defined variable space has a low dimensionality. The relations between functions and these variables can more easily be learned with NN's. Such a simplified NN-model is then incorporated in a dynamic simulator to predict the performance of the reactor system, and compares extremely well with a phenomenological model simulator. A new approach to conduct process identification on the basis of continuous data is also proposed.

Keywords. Adsorption, multi-dimensionally, neural nets, perturbation analysis.

INTRODUCTION

The problem with most unit operations and metallurgical processes is that they are ill-defined in some way. This places an obligation on the modelling engineer to construct adequate mathematical models to be fully representative of the whole system. In weakly-defined systems, such models are usually domain-specific and unreliable. The carbon-in-leach (CIL) process will be used here to illustrate the proposed theory of neural net (NN) modelling.

Recently KBS has been applied increasingly in treating ill-defined problems, especially in the chemical industry. Reuter, Van Deventer and Van der Merve (1991) explained the application of KBS to the simulation of CIP- and CIL-circuits. From batch reactor simulations, they constructed system variable profiles (such as gold concentration and gold-loading-on-carbon profiles) which were coupled with the kinetics of the adsorption and leaching processes through a data base.

The search for a technique which has the ability to analyze a "black-box" system effectively is still continuing. Such a technique should be able to construct a mapping between system variables and parameters so that this relational mapping can be implemented as a "black-box" model. Most importantly, it should have the ability to learn relations within those system domains which cannot readily be modelled by means of conventional techniques. The connectionist network-approach studied here seems to be a promising modelling tool to cover the abovementioned gaps in the metallurgical modelling toolkit.

A further major problem for any modelling expert is the question of multi-dimensionality. The amount of information (such as system data points) required to model a process adequately, increases exponentially with a rise in the number of process variables, which in turn has an influence on the process performance. In order to explain this, consider a function f to be depicted on an m-dimensional variable space $R^m$. If $y_j$ is the average number of data points necessary within the jth dimension to represent the whole system sufficiently, the minimum number of data points needed will be

$$P = \prod_{j=1}^{m} y_j$$  \hspace{1cm} (1)

It should now be clear that the addition of every single dimension will cause an exponential decay in information contained by a specific set of information for a system. This problem will also be addressed using the CIL-example. Also, the process of training a NN-model is much more complex for a system of higher dimensionality, owing to an increase in complexity of the total mapping with every additional variable. It will be shown how automated perturbation analysis can be performed by means of a trained NN and its training data set in order to quantify the degree of influence of the separate process variables on the different function surfaces. Having done
so, variables which are identified as "low-key" variables within a certain domain of the system, can be eliminated and dimensionality is reduced accordingly.

**DIRECT MODELLING OF A CONTINUOUS REACTOR**

In practice a model compiled on batch data needs, almost without exception, to be adjusted considerably if it is to be incorporated in a simulator for a continuous system. Hence, it is proposed here that process identification should be conducted preferably on the basis of continuous data.

The dynamics of a continuous reactor can be described by performing material, energy and momentum balances over the reactor. A typical material balance equation for substance X in a CSTR is:

\[
\tau \frac{dX}{dt} = X_{in} - X_{out} + f
\]  

(2)

\(f\) represents the reaction(s) which take place within the reactor. This reaction term can be expressed in terms of the other three terms of the latter equation:

\[
f = \tau \frac{dX}{dt} - X_{in} + X_{out}
\]  

(3)

The terms on the righthand side of equation 3 should be measurable so that \(f\) can be calculated with this expression. At the same instant when \(f\) is determined, some other process variables which might probably have an influence on \(f\), are evaluated as well and combined with \(f\) to form a system data point. All such data points are put together to make up an information data set, which can be used by the modelling engineer to develop a model in some way or another. In the case of a NN modelling approach, this data set is employed as a "training set" by a NN training program to learn the relations between the different process variables and functions.

**NEURAL NETS**

Various NN-topologies have been proposed. However, recent success in connectionist network research is mainly attributed to the achievements in multilayer perceptrons with backpropagation training algorithms. The architecture and training algorithm of the three-layer perceptron used here will now be discussed briefly.

**NN-architecture**

Figure 1 illustrates the NN used for the purpose of this paper. The three-layer perceptron consists of three layers of nodes, viz. (1) an input layer, (2) a single hidden layer and (3) an output layer. Nodes of adjacent layers are connected and these connections quantified with weights.

The weight matrix of a trained NN contains the information about a system under investigation. Each node is characterized by an activation function (linear or sigmoidal) and a bias node with a constant value of 1, which provides extra degrees of freedom which enables a fitted curve to be moved up or down in the variable space. The activities of the input nodes are calculated by their activation functions, which are linear in this case. They simply take the values of the system variable inputs. The inputs to the hidden and output nodes are calculated by adding the products of the activities of all nodes in the previous layer with the corresponding weights of the connections which attach the separate nodes in the previous layer to the specific node. The activities of the hidden nodes are calculated with the sigmoidal squashing function \((\text{act} = 1/(1 + \exp(-\text{input})))\), while the outputs of the NN are equal to the activities of the output nodes.

**Training a Neural Net**

The training process of NN's is an important part of NN-research. It involves optimizing a weight matrix of a NN so that the NN will memorize the mapping between a number of system variables and functions. The backpropagation algorithm, as described by Rumelhart, Hinton and Williams (1986), is used by the NN training program, which incorporates the conjugate gradient (CG-) optimization algorithm with restart procedures of Powell (1977) as optimization method. During each training iteration all weights of the NN are adjusted in such a way as to decrease the value of an objective function. Most NN training programs employ the LMS-error function as objective function and the net used here is no exception.

Mainly two steps can be identified during each iteration of training, viz.

1. determining the error-weight gradients of each weight in the network by presenting all training examples once to the net;
2. adjusting the weight matrix by means of an optimization method.

The first step in calculating the error-weight gradients (step 1) is performed in the following substeps, for each training example respectively:

(i) A training example is fed to the input layer. The activities of all nodes in the net are calculated during a feedforward step as described in previous paragraphs.

(ii) The value of the error function at each output node is calculated.

(iii) During a backward sweep the error-weight gradients throughout the whole weight space are determined via the procedure of backpropagation.

It is vitally important that an effective optimization method be used during step 2. The CG-method is usually able to
locate a minimum of a multivariate function much faster than the momentum algorithm which is customarily employed with backpropagation. Furthermore, its memory usage is in the order of \(N\) (number of weights) locations. Also important to note is that the CG-technique eliminates the choice of critical parameters, such as the learning rate and momentum parameters of the momentum algorithm. On the other hand, like all gradient descent optimization techniques, the CG-algorithm can converge into local minima, which is a major drawback.

**MODELLING A TYPICAL METALLURGICAL PROCESS**

A known phenomenological model for the CIL-process was used to generate data points during a single simulation run of a continuous CIL simulator for a 5-stage cascade. These data points were determined at consecutive time steps for all 5 stages in the cascade, and the data points represented a very broad and practical range of the CIL-variable space. It constituted a training set which could be used to train a NN-model for the CIL-system. It is shown here how a NN-model for a specific ore and carbon type can be developed and refined by analyzing the training set, quantifying the influence of the different variables and by reducing the dimensionality accordingly.

![Fig. 2: Schematic diagram of the CIL-cascade](image)

The mass balances of the gold in the different phases are described by equations 4 to 6 and were constructed for the cascade illustrated in Fig. 2. With the exception of the first and last reactor stages, these balance equations are applicable for each intermediate stage in the CIL-cascade, as portrayed schematically in Fig. 2. \(C\) is the gold concentration in water, \(q\) represents the gold loading on the carbon and \(G\) the degree of gold in the ore. Volumetric fractions for water \((I)\), carbon \((c)\) and ore \((o)\) are expressed by \(\varepsilon\).

**Gold balance in liquid phase for the \(i^{th}\) stage:**

\[
V_i \frac{dC_i}{dt} = \frac{e_{o,i}}{(e_{o,i} + e_{h,i})} Q_{i,h} P_0 q_{i,h} + \frac{e_{h,i}}{(e_{h,i} + e_{w,i})} Q_{i,w} C_{i,w} - \frac{e_{w,i}}{(e_{w,i} + e_{c,i})} (Q_{i,c} + Q_{i}) C_i + f_i
\]

**Gold balance on carbon for the \(j^{th}\) stage:**

\[
W_j \frac{dQ_j}{dt} = \frac{e_{o,j}}{(e_{o,j} + e_{h,j})} Q_{j,h} P_0 q_{j,h} - \frac{e_{h,j}}{(e_{h,j} + e_{w,j})} Q_{j,w} C_{j,w} + \frac{e_{w,j}}{(e_{w,j} + e_{c,j})} (Q_{j,c} + Q_{j}) C_j + f_{j}
\]

**Gold balance on ore for the \(i^{th}\) stage:**

\[
M_i \frac{dG_i}{dt} = \frac{e_{o,i}}{(e_{o,i} + e_{h,i})} Q_{i,h} P_0 G_{i,h} + \frac{e_{h,i}}{(e_{h,i} + e_{w,i})} Q_{i,w} G_{i,w} - \frac{e_{w,i}}{(e_{w,i} + e_{c,i})} (Q_{i,c} + Q_{i}) G_i + f_{i}
\]

The volumetric flow rate of the overflow slurry stream \((Q_{o,3})\) can be calculated as follows:

\[
Q_{o,3} = Q_i + Q_{o,0}
\]

\(Q_i\) is the volumetric flow rate of the slurry input to the first stage. During the intervals when no carbon transfer takes place, \(Q_o\) is non-existent and \(Q_{o,0}\) equals \(Q_i\) for all stages.

\(f_1, f_2\) and \(f_3\) are the kinetic reaction terms and are described by the following equations according to the film diffusion model for adsorption and an empirical expression for the process of leaching:

\[
l_2 = -\frac{6 k_1 W_i}{P_c} (C_i - C_s)
\]

\[
l_3 = -k_1 M_i (G_i - G_s)^2
\]

\[
l_1 = -(l_2 + l_3)
\]

A Freundlich isotherm describes equilibrium at the carbon surface, so that

\[
C_i = \left(\frac{Q_{o,3}}{A}\right)^{1/n}
\]

where \(A\) and \(n\) are the equilibrium parameters, specified to be 6 and 0.2 respectively for the specific carbon. Further specifications of the carbon is a density \((\rho_c)\) of 900 kg m\(^{-3}\) and an average particle diameter of 1.4 mm. The adsorptive and leaching coefficients \((k_1\) and \(k_2\) are assumed to remain constant at \(10^{-5}\) m sec\(^{-1}\) and \(1.2 \text{ kg}_{\text{ore}}/\text{gAu} / \text{sec}\) respectively. The grade of gold in the ore at infinity \((G^\infty)\) was specified to be \(6 \times 10^{-6}\text{ gAu}/\text{gore}\).

During a typical simulation run with the phenomenological model simulator, \(l_1, l_2\) and \(l_3\) were calculated and the values of \(C, q, G, M\) (mass of ore in the reactor) and \(W\) (mass of carbon in the reactor) registered together with the three reaction terms. At random values for \(M\) (400-500 tons) and \(W\) (between 10 and 25 tons) were generated at each time step for every stage. This was done at fixed reactor volumes of 750m\(^3\) for all stages. The values of \(l_1, l_2\) and \(l_3\) of these data points (variable/function pairs with 5 input variables and 3 output functions) were then randomly corrupted with 20% Gaussian noise. The first training set thus contained noisy data which is typical of an
industrial system. After scaling the values of the data points, this training set was employed to train a NN which in turn was used to perform perturbation analyses on the training data set.

Training the Neural Net

It is important largely for three reasons to scale the input and output values of the variable and function values. Firstly the input node activities of the NN should be bounded so that the weights between the inputs of the sigmoidal hidden nodes are sensible (their absolute values should not be too large). This will cause extremely slow training convergence. Secondly it is imperative that the input values to the net are of the same absolute order. During training each weight is adjusted according to the overall smallest error-weight gradient. The error-weight gradient is directly proportional to the magnitude of the objective function and will slow down the training process. Thirdly, whilst learning is complicated if highly curved function surfaces should be found, NN's can easily learn smooth relations.

For these reasons, the values of the input-output pairs are scaled as illustrated in Fig. 3. The variables C and Q varied over 3 to 4 orders of magnitude (±0.001 to ±10) and were effectively scaled logarithmically. In order to do comparable perturbation analyses the range of all scales variables was specified to be between approximately 4 and -4. NN's with activation functions such as this seem to control such input ranges effectively. All variables were scaled accordingly.

Six NN's with 12 hidden nodes each and different random weight matrices at the start of the training runs, were trained on a 2000 noisy data point training set constructed as described earlier. The training results are displayed in Table 1. From these results it is clear that NN's handle such input ranges well. The average absolute errors (normalized as a % of 4) when compared to the "true" scaled phenomenological model calculations for the input variable values in the training set, show that the trained neural nets' predictions of $f_1$, $f_2$ and $f_3$ are much closer to the desired scaled values as is the case for the noisy scaled function values. Note that these errors were calculated as the average absolute differences between the specific scaled function values and the model-predicted scaled function values ("true" scaled values) for all training data points and over all three output nodes. The number of training iterations is also significantly lower than would have been the case for a training program with a momentum optimization algorithm.

One of these NN's was used to complete a perturbation analysis on the training set. This net will be referred to as NN (A) below.

Automated perturbation analyses

It is very difficult and sometimes impractical to perform a perturbation analysis on noisy data points gathered from an industrial system. Some techniques such as factorial design have been developed to analyse data points of an ideally defined system. Noisy data limit the applicability of such linear perturbation techniques.

A NN automatically identifies smooth function curves which enable its convenient use in perturbation analysis on the whole system. This is done in the following manner: For a given variable space coordinate the value of a function is calculated by the trained NN (note that the scaled NN output value is scaled back to the function value). The predicted value is the perturbation midpoint. The scaled value of a specific variable is now adjusted to both sides of the midpoint with the same constant absolute value, while the other variables are kept constant. The values of the function at these two new coordinates are predicted by the NN. The absolute differences between the midpoint and the two discriminate perturbated function values are divided by the absolute midpoint value, and the average of these two relative values calculated. This average is expressed as a percentage (see Table 2). This is done for all variables (dimensions) for a specific data point. At a specific location within a system's variable space the influences of the different variables on the separate functions are now quantified so that the magnitudes of these average perturbated values can be compared directly. Non-significant variables can now be eliminated from the variable space within the relevant domains.

For this specific CIL-system where the input variables were scaled between 4 and -4, a low constant value of 0.1 was selected to adjust a certain variable value to both sides of the midpoint. The normalization constant is chosen to be 0.1, which will prevent the NN from being used in its highly inaccurate extrapolative domain. The importance of

<table>
<thead>
<tr>
<th>Number of training iterations</th>
<th>Average absolute error normalized as % of 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noisy data</td>
<td>NN-predictions</td>
</tr>
<tr>
<td>1567</td>
<td>3.34</td>
</tr>
<tr>
<td>1106</td>
<td>3.22</td>
</tr>
<tr>
<td>3455</td>
<td>3.56</td>
</tr>
<tr>
<td>2825</td>
<td>4.21</td>
</tr>
<tr>
<td>3214</td>
<td>3.21</td>
</tr>
<tr>
<td>1918</td>
<td>3.67</td>
</tr>
<tr>
<td>2349*</td>
<td>3.54*</td>
</tr>
</tbody>
</table>

TABLE 1: Training Results of 6 NN's converged upon by a Training Set with 20% Gaussian noise on $f_1$, $f_2$ & $f_3$ - (all nets have 12 hidden nodes, but different random weight matrices at the start of the training runs)

Fig. 3: NN (A) trained on noisy data for the CIL-process
TABLE 2 Perturbation results of five randomly selected data points, using NN (A)

<table>
<thead>
<tr>
<th>Data point coordinates</th>
<th>C [g m(^{-3})]</th>
<th>q [kg g(^{-1})]</th>
<th>G [g kg(^{-1})]</th>
<th>M [tons]</th>
<th>W [tons]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3340</td>
<td>7290</td>
<td>5.0</td>
<td>450</td>
<td>18</td>
</tr>
<tr>
<td>2</td>
<td>750</td>
<td>4610</td>
<td>3.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>125</td>
<td>1500</td>
<td>2.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>26</td>
<td>337</td>
<td>1.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>91</td>
<td>1.3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Average relative perturbation values for \(f_2\):  

<table>
<thead>
<tr>
<th>Data point coordinates</th>
<th>(f_2) [%]</th>
<th>(f_3) [%]</th>
<th>(f_4) [%]</th>
<th>(f_5) [%]</th>
<th>(f_6) [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>33.64</td>
<td>14.07</td>
<td>1.57</td>
<td>3.12</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3.65</td>
<td>1.62</td>
<td>0.52</td>
<td>2.00</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1.14</td>
<td>3.01</td>
<td>0.14</td>
<td>2.90</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.65</td>
<td>11.45</td>
<td>0.48</td>
<td>2.97</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2.10</td>
<td>35.93</td>
<td>1.84</td>
<td>8.15</td>
<td></td>
</tr>
</tbody>
</table>

Average relative perturbation values for \(f_3\):  

<table>
<thead>
<tr>
<th>Data point coordinates</th>
<th>(f_3) [%]</th>
<th>(f_4) [%]</th>
<th>(f_5) [%]</th>
<th>(f_6) [%]</th>
<th>(f_7) [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.74</td>
<td>1.99</td>
<td>0.43</td>
<td>1.62</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.47</td>
<td>1.99</td>
<td>0.43</td>
<td>1.62</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.62</td>
<td>1.99</td>
<td>0.43</td>
<td>1.62</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.47</td>
<td>1.99</td>
<td>0.43</td>
<td>1.62</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.62</td>
<td>1.99</td>
<td>0.43</td>
<td>1.62</td>
<td></td>
</tr>
</tbody>
</table>

A simplified NN-model

As was expected, the perturbation results showed that \(f_1\), \(f_2\) and \(f_3\) are functions of the following variables for specific ore and carbon types:

\[
\begin{align*}
\text{for } q \leq 1 \text{ g kg}^{-1}: \\
f_1 &= \phi(C, q, G, M, W) \\
f_2 &= \phi(C, q, W) \\
\end{align*}
\]

(12) (13)

\[
\begin{align*}
\text{for } q > 1 \text{ g kg}^{-1}: \\
f_1 &= \phi(C, G, M, W) \\
f_2 &= \phi(G, W) \\
\end{align*}
\]

(14) (15)

Identifying mathematical relations. NN (A) was also used to confirm the linear relationships between \(f_2 \& W\), as well as \(f_2 \& M\). In the case of \(f_2 \& W\) the first three data points of Table 2 are used to illustrate how this is done. For each of these data points the value of \(W\) was varied within its data range, while the values of the other 4 variables remained constant. \(f_2\) was calculated with NN (A) at different \(W\)-values for each data point. These NN-predictions of \(f_2\) were plotted against \(W\) for each data point in Fig. 4. This figure confirms the direct proportionality between \(f_2\) and \(W\). The linear curves can be forced through the origin, so that \(f_2\) can be expressed as follows:

\[
\begin{align*}
f_2 &= W_{0}\rho_2, \\
\text{where } \rho_2 &= \alpha(C, q) \text{ if } q > 1.0 \ \\
\text{and } \rho_2 &= \alpha(C) \text{ if } q < 1.0.
\end{align*}
\]

The same procedure was performed for \(f_3\) and \(M\), and equation 18 displays the simplified functions.

\[
\begin{align*}
f_3 &= M_{0}\rho_3, \\
\text{where } \rho_3 &= \beta(G).
\end{align*}
\]

(18)

It was assumed that some a priori knowledge about the CIL-system is available. It should be highlighted that the NN approach that is proposed here does not tend to replace all existing modelling techniques and available a priori knowledge about the system. This modelling approach should contribute to the modelling toolkit. If it is a known fact that the gold mass balance in the liquid phase is dependent on an adsorption and leaching process, equations 17 and 18 can be combined to describe \(f_1\):

\[
\begin{align*}
f_1 &= M_{0}\rho_3 - W_{0}\rho_2.
\end{align*}
\]

(19)
According to these relationships, a simplified NN-model (NN-model I) with three NN's (two nets for each q-interval for $r_2$ and a net for $r_3$) was constructed. These NN's were trained on exactly the same training data with the exception that the data set did not contain noise. The NN's of $r_3$ and $r_2$ (lower q-range) had only one input node (the bias node ignored), while the NN for $r_2$ within the higher q-range was specified to have two input nodes. The dimensionality of the "ill-defined" parts of the system has been reduced considerably from a 5-dimensional one to an ill-defined dimensionality of only 2 (function $r_2$ for $q > 1.0$) and 1 (functions $r_2$ for $q < 1.0$ and $r_3$) by means of the two techniques described above. NN-model I was used to replace the phenomenological model in a dynamic CIL-cascade simulator. At each time step, $t_1$, $t_2$ and $t_3$ were calculated by equations 17, 18 and 19.

Comparing simulation results. Another NN-model (NN-model II) consisting of three NN's (one net for each of $t_1$, $t_2$ and $t_3$) was trained with the same data points used for training NN-model I. Each one of these NN's had five input nodes for the five variables as explained. The reason for splitting up the different output features of NN (A) was to simplify the training processes to find suitable mappings between $t_1$, $t_2$ and $t_3$ and the 5-dimensional variable space more easily. The training sets for these NN's were also noiseless. The convergence errors for these trained NN's compared well to the convergence errors of the NN's in NN-model I. NN-model II was also incorporated in the simulator and the simulation results of the two NN-model simulators compared to the results of the phenomenological model-simulator. All NN's of NN-model I and II were defined to have 7 hidden nodes. The number of NN-weights (degrees of freedom) of the two NN-models are thus the same, so that their simulation results can be compared as well. Some simulation results of reactor stage 1 for the three models referred to, are shown in Fig. 5.

The simulation results with NN-model I are highly superior to the results of simulations with NN-model II, if compared to the simulation results of the phenomenological model-simulator, which are viewed to be the standard. This is mainly due to a large reduction in dimensionality and an associated improvement of the model.

CONCLUSIONS

A new approach to conduct process modelling on the basis of continuous data collected directly from an industrial reactor, was proposed and applied to modelling a CIL-process. Multi-dimensionality as a problem to any modelling technique, was addressed with different approaches which utilized a NN trained on noisy data. The dimensionality was reduced and a very accurate NN-model could be constructed for the CIL-process. This model was incorporated in a dynamic simulator which predicted the performance of the CIL-process successfully.

Critics of NN's as modelling technique have valid doubts if the dimensionality-problem is not solved adequately. Further work is conducted to refine and automate the proposed techniques for reducing dimensionality. This also includes the design of a new NN-topology which should be able to reduce dimensionality, even within the ill-defined variable space.

REFERENCES


LIST OF SYMBOLS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Gold concentration [g.m$^{-3}$]</td>
</tr>
<tr>
<td>d</td>
<td>Average diameter of solid particle [m]</td>
</tr>
<tr>
<td>f</td>
<td>Reaction term in mass balance</td>
</tr>
<tr>
<td>G</td>
<td>Gold grade in the ore [g.kg$^{-1}$]</td>
</tr>
<tr>
<td>k$_h$</td>
<td>Adsorption constant [m.sec$^{-1}$]</td>
</tr>
<tr>
<td>k$_l$</td>
<td>Leaching constant [kg.g$^{-1}$.sec$^{-1}$]</td>
</tr>
<tr>
<td>M</td>
<td>Mass of ore in a single reactor [kg]</td>
</tr>
<tr>
<td>P</td>
<td>Density [kg.m$^{-3}$]</td>
</tr>
<tr>
<td>Q</td>
<td>Volumetric slurry flow rate [m$^3$.sec$^{-1}$]</td>
</tr>
<tr>
<td>q</td>
<td>Loading of the carbon [g.kg$^{-1}$]</td>
</tr>
<tr>
<td>t</td>
<td>Time [sec]</td>
</tr>
<tr>
<td>V</td>
<td>Reactor volume [m$^3$]</td>
</tr>
<tr>
<td>W</td>
<td>Mass of carbon in a single reactor [kg]</td>
</tr>
</tbody>
</table>

Greek symbols

c | Void fractions |
i | Residence time of reactor stream [sec] |

Subscripts

c | Carbon |
l | Liquid phase |
i | $i^{th}$ reactor or stage |
in | Reactor entrance stream |
o | Ore |
out | Reactor exit stream |