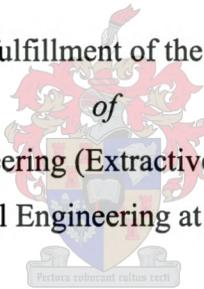


# **MODELLING AND CONTROL OF AN AUTOGENOUS MILL USING A STATE SPACE METHODOLOGY AND NEURAL NETWORKS**

*by*

**JACOBUS WILLEM DE VILLIERS GROENEWALD**

Thesis submitted in partial fulfillment of the requirements for the Degree  
*of*  
Master of Science in Engineering (Extractive Metallurgical Engineering)  
in the Department of Chemical Engineering at the University of Stellenbosch



*Supervised by*

Prof. C. Aldrich

Prof. L. Lorenzen

Mr. J.J. Eksteen

STELLENBOSCH

DECEMBER 2002

## **DECLARATION**

I, the undersigned, hereby declare that the work contained in this thesis is my own original work and that I have not previously in its entirety or in part submitted it at any university for a degree.

Jacobus Willem De Villiers Groenewald

15 September 2002

## SYNOPSIS

Metallurgical processes are often high dimensional and non-linear making them difficult to understand, model and control. Whereas the human eye has extensively been used in discerning temporal patterns in historical process data from these processes, the systematic study of such data has only recently come to the forefront. This resulted predominantly from the inadequacy of previously used linear techniques and the computational power required when analysing the non-linear dynamics underlying these systems. Furthermore, owing to the recent progress made with regard to the identification of non-linear systems and the increased availability of computational power, the application of non-linear modelling techniques for the development of neural network models to be used in advanced control systems has become a potential alternative to operator experience.

The objective of this study was the development of a non-linear, dynamic model of an autogenous mill for use in an advanced control system. This was accomplished through system identification, modelling and prediction, and application to control. For system identification, the attractor was reconstructed based on Taken's theorem making use of both the Method Of Delays and singular spectrum analysis. Modelling consisted of the development of multi-layer perceptron neural network, radial basis function neural network, and support vector machine models for the prediction of the power drawn by an autogenous mill. The best model was subsequently selected and validated through its application to control. This was accomplished by means of developing a neurocontroller, which was tested under simulation.

Initial inspection of the process data to be modelled indicated that it contained a considerable amount noise. However, using the method of surrogate data, it was found that the time series representing the power drawn by the autogenous mill clearly exhibited deterministic character, making it suitable for predictive modelling. It was subsequently found that, when using the data for attractor reconstruction, a connection existed between the embedding strategy used, the quality of the reconstructed attractor, and the quality of the resulting model. Owing to the high degree of noise in the data it was found that the singular spectrum analysis embeddings resulted in better

quality reconstructed attractors that covered a larger part of the state space when compared to the method of delays embeddings; the data embedded using singular spectrum analysis also resulting in the development of better quality models.

From a modelling perspective it was found that the multi-layer perceptron neural network models generally performed the best; a multi-layer perceptron neural network model having an appropriately embedded multi-dimensional input space outperforming all the other developed models with regard to free-run prediction success. However, none of the non-linear models performed significantly better than the ARX model with regard to one-step prediction results (based on the  $R^2$  statistic); the one-step predictions having a prediction interval of 30 seconds. In general the best model was a multi-layer perceptron neural network model having an input space consisting of the FAG mill power ( $x_1$ ), the FAG mill load ( $x_2$ ), the FAG mill coarse ore feed rate ( $x_3$ ), the FAG mill fine ore feed rate ( $x_4$ ), the FAG mill inlet water flow rate ( $x_7$ ) and the FAG mill discharge flow rates ( $x_9, x_{10}$ ).

Since the accuracy of any neural network model is highly dependent on its training data, a process model diagnostic system was developed to accompany the process model. Linear principal component analysis was used for this purposes and the resulting diagnostic system was successfully used for data validation. One of the models developed during this research was also successfully used for the development of a neurocontroller, proving its possible use in an advanced control system.

## OPSOMMING

Metallurgiese prosesse is gewoonlik hoogs dimensioneel en nie-lineêr, wat dit moeilik maak om te verstaan, modelleer, en te beheer. Alhoewel die menslike oog alreeds wyd gebruik word om temporale patrone in historiese proses data te onderskei, het die sistematiese studie van hierdie tipe data eers onlangs na vore gekom. Dit is hoofsaaklik na aanleiding van die onvoldoende resultate wat verkry is deur van voorafgaande lineêre tegnieke gebruik te maak, asook die beperkende berekenings vermoë wat beskikbaar was vir analise van onderliggend nie-lineêre dinamiese stelsels. 'n Verder bydraende faktor is die onlangse vordering wat gemaak is met betrekking tot die identifikasie van nie-lineêre stelsels en die toename in beskikbaarheid van rekenaar-vermoë. Die toepassing van nie-lineêre modellerings tegnieke vir die ontwikkeling van neurale netwerke om gebruik te word in gevorderde beheerstelsels, het 'n potensiële alternatief geword tot operateur ondervinding.

Die doelwit van hierdie studie was die ontwikkeling van 'n gevorderde beheerstelsel vir 'n outogene meul gebaseer op 'n nie-lineêre, dinamiese model. Dit is bereik deur middel van stelsel-identifikasie, modellering en voorspelling, en laastens implementering van die beheerstelsel. Vir stelsel-identifikasie is die atraktor van die stelsel bepaal soos gebaseer op Taken se teorema deur gebruik te maak van beide die metode van vertraging en enkelvoudige spektrum analise. Modellering van die stelsel vir die voorspelling van krag-verbruik deur die outogene meul het bestaan uit die ontwikkeling van multilaag-perseptron-neurale netwerke, radiaalbasisfunksie-neurale netwerke, en steunvektor-masjien-modelle. Die beste model is daarna gekies vir validasie deur middel van toepassing vir beheer. Dit is bereik deur 'n neurobeheerder te ontwikkel en te toets deur middel van simulasie.

Die aanvanklike inspeksie van proses data wat gebruik sou word vir modellering het egter getoon dat die data 'n aansienlike hoeveelheid geraas bevat. Nietemin, deur die gebruik van 'n surrogaat-data-metode, is dit bevind dat die tyd-reeks wat die krag verbruik van die outogene meul beskryf, duidelik deterministiese karakter toon en dat dit dus wenslik is om 'n nie-lineêre voorspellings-model, soos 'n neurale netwerk te gebruik. Gevolglik is gevind dat, wanneer die data vir atraktor hersamestelling gebruik

word, 'n verband bestaan tussen die ontvouing-strategie wat gebruik word, die kwaliteit van die gerekonstrueerde atraktor, en die kwaliteit van die daaropvolgende model. As gevolg van die geraas in die data is gevind dat die ontvouing gebaseer op enkelvoudige spektrum analise 'n beter kwaliteit atraktor hersamestelling lewer. So ook is gevind dat 'n groter deel van die toestandruimte gedek word in vergelyking met die metode van vertraging-ontvouing. Deur gebruik te maak van enkelvoudige spektrum-analise, het die dataontvouing ook beter kwaliteit modelle opgelewer.

Vanuit 'n modellerings-perspektief is gevind dat die multilaag-perseptron-neurale netwerk-modelle in die algemeen die beste gevaar het. 'n Multilaag-perseptron-neurale netwerk met 'n gepaste ontvoude multidimensionele invoer-spasie het die beste gevaar van al die ontwikkelde modelle met betrekking tot vryloop-voorspellings. Geen van die nie-lineêre modelle het egter beduidend (op 'n  $R^2$  basis) beter gevaar as die ARX model wanneer daar na die eenstap-voorspellings (oor 'n 30 sekonde interval) gekyk word nie. Die multilaag-perseptron-neurale netwerk met 'n invoer-spasie bestaande uit die meul krag-verbruik ( $x_1$ ), die meul lading ( $x_2$ ), die meul growwe-erts voertempo ( $x_3$ ), die meul fyn-erts voertemp ( $x_4$ ), die meul inlaat-water vloeitempo ( $x_7$ ) en die meul uitlaat vloeitempo's ( $x_9, x_{10}$ ) het in die algemeen die beste gevaar.

Aangesien die akkuraatheid van enige neurale netwerk afhanklik is van die data waarmee dit aanvanklik opgestel is, is 'n diagnostiese proses model ontwikkel om die proses-model te vergesel. Lineêre hoofkomponent analise is vir hierdie doel aangewend en die gevolglike diagnostiese stelsel is suksesvol aangewend vir data-validasie. Een van die modelle ontwikkel gedurende hierdie navorsing is ook suksesvol gebruik vir die ontwikkeling van 'n neurobeheerder wat dien as bewys dat die model goed gebruik kan word in 'n gevorderde beheerstelsel.

## **ACKNOWLEDGEMENTS**

My Heavenly Father, for having given me the strength with which to complete this undertaking.

Prof. Chris Aldrich, Prof. Leon Lorenzen and Mr. Jacques Eksteen, my supervisors, for their guidance and support.

J.P. Barnard, for use of his QuickIdent toolbox, developed for non-linear time series analysis and dynamic modelling.

Alex Conradie, for use of the SANE algorithm as implemented by him, and his assistance with the development of a neurocontroller.

Anglo Platinum, and in particular Leon Coetzer, for providing me with the opportunity to further my studies.

My family, for their patience and for believing in me.

All my friends, for their encouragement.

Everything should be made as simple as possible, but not simpler.

*Albert Einstein (1879-1955)*

To explain all nature is too difficult a task for any one man or even for any one age. 'Tis much better to do a little with certainty, and leave the rest for others that come after you, than to explain all things.

*Isaac Newton (1642-1727)*

## TABLE OF CONTENTS

<b>SYNOPSIS</b>	<b>i</b>
<b>OPSOMMING</b>	<b>iii</b>
<b>ACKNOWLEDGEMENTS</b>	<b>v</b>
<b>TABLE OF CONTENTS</b>	<b>vii</b>
<b>LIST OF TABLES</b>	<b>xi</b>
<b>LIST OF FIGURES</b>	<b>xi</b>
<b>LIST OF KEY SYMBOLS USED</b>	<b>xvii</b>
<b>LIST OF KEY ABBREVIATIONS USED</b>	<b>xix</b>
<b>1 MODELLING AND CONTROL OF COMMINUTION CIRCUITS</b>	<b>1</b>
1.1 Mineral beneficiation	1
1.2 Economics of comminution	3
1.3 Importance of milling	4
1.4 Control of a milling circuit	6
1.5 Intelligent control	7
1.6 A literature survey on autogenous mill modelling	11
1.6.1 Black box modelling	11
1.6.2 Summary	14
1.7 Dynamic non-linear modelling	15
1.8 Thesis objectives	16
<b>2 CONCENTRATION OF PGMS AT ANGLO PLATINUM'S AMANDELBULT PLANT</b>	<b>18</b>
2.1 UG2 Reef mineralogy	19
2.2 UG2 concentrator process overview	21
2.3 UG2 FAG mill process variable classification and instrumentation identification	22
2.3.1 What is the objective to be achieved?	23
2.3.2 What are the controllable variables?	24

2.3.3	What sensing instruments may be used to detect changes in circuit operation?	24
2.3.4	How must the controllable variables be altered to compensate for these changes?	26
2.3.5	Miscellaneous milling circuit information	27
<b>2.4</b>	<b>Summary</b>	<b>28</b>
<b>3</b>	<b>SYSTEM IDENTIFICATION</b>	<b>29</b>
<b>3.1</b>	<b>Identification through state space reconstruction</b>	<b>29</b>
3.1.1	Method of delays	30
3.1.2	Singular spectrum analysis	37
3.1.3	Quality of state space reconstruction	38
<b>3.2</b>	<b>Surrogate data</b>	<b>44</b>
<b>3.3</b>	<b>Modelling</b>	<b>46</b>
3.3.1	Multi-layer perceptron neural networks	47
3.3.2	Radial basis function neural networks	52
3.3.3	Least squares support vector machines	56
<b>3.4</b>	<b>Prediction</b>	<b>61</b>
<b>4</b>	<b>ANALYSIS OF PLANT DATA</b>	<b>63</b>
<b>4.1</b>	<b>Data inspection</b>	<b>64</b>
4.1.1	Raw data analysis	64
4.1.2	Subset data analysis	65
4.1.3	Results discussion	72
<b>4.2</b>	<b>Fourier analysis</b>	<b>72</b>
4.2.1	Fourier power spectrum	73
<b>4.3</b>	<b>Statistics for non-linear time series analysis</b>	<b>74</b>
4.3.1	Linear Autocorrelation statistic	75
4.3.2	Average Mutual Information statistic	79
4.3.3	False Nearest Neighbours criterion	82
4.3.4	Results discussion	84
<b>4.4</b>	<b>Data classification</b>	<b>85</b>
<b>4.5</b>	<b>Summary</b>	<b>88</b>
<b>5</b>	<b>MODELLING RESULTS</b>	<b>90</b>
<b>5.1</b>	<b>Process dataset</b>	<b>90</b>
5.1.1	Linear ARX model	92
5.1.2	One-dimensional time series modelling of the FAG mill power using a MLP neural network	95

5.1.3	Multi-dimensional time series modelling of the FAG mill power using a MLP neural network	106
5.1.4	Modelling of the FAG mill power using LS-SVM's	121
5.1.5	Modelling of the FAG mill power using a RBF neural network	127
5.1.6	Application of multi-dimensional time series models of the FAG mill power ( $x_1$ ) to free-run prediction	137
5.1.7	Prediction of FAG mill rougher feed 2 flow rate ( $x_{12}$ )	142
<b>5.2</b>	<b>Wipfrag and Malvern datasets</b>	<b>146</b>
5.2.1	Embedding the data	147
5.2.2	Attractor reconstruction	147
5.2.3	Model structure and validation	149
5.2.4	Use of graphical residual analysis for model validation	151
5.2.5	Use of surrogate data for model validation	152
5.2.6	Free-run prediction	154
5.2.7	Results discussion	155
<b>5.3</b>	<b>Summary</b>	<b>156</b>
<b>6</b>	<b>ON-LINE MODEL VALIDATION</b>	<b>158</b>
<b>6.1</b>	<b>Process model diagnostic system</b>	<b>158</b>
6.1.1	Use of PCA for a process model diagnostic system	160
<b>6.2</b>	<b>Validating new process data to be used on an existing empirical model</b>	<b>168</b>
<b>6.3</b>	<b>Application to process monitoring</b>	<b>175</b>
<b>6.4</b>	<b>Summary</b>	<b>185</b>
<b>7</b>	<b>AUTOMATED MODEL-BASED MILL CONTROL</b>	<b>187</b>
<b>7.1</b>	<b>Non-linear neurocontrol</b>	<b>187</b>
7.1.1	Neurocontroller fundamentals	187
7.1.2	Amandelbult FAG mill neurocontroller	194
<b>7.2</b>	<b>Summary</b>	<b>196</b>
<b>8</b>	<b>CONCLUSIONS</b>	<b>197</b>
<b>9</b>	<b>REFERENCES</b>	<b>202</b>
<b>10</b>	<b>APPENDIX A</b>	<b>220</b>
<b>10.1</b>	<b>Comminution</b>	<b>220</b>
10.1.1	Comminution theory	220
10.1.2	Tumbling mills	222
10.1.3	Autogenous milling	223
<b>10.2</b>	<b>Process variable classification</b>	<b>226</b>

<b>10.3</b>	<b>Lyapunov spectrum</b>	<b>228</b>
<b>10.4</b>	<b>Correlation and correlation coefficients</b>	<b>228</b>
<b>10.5</b>	<b>Fourier analysis</b>	<b>229</b>
<b>11</b>	<b>APPENDIX B – ADDITIONAL TABLES AND FIGURES</b>	<b>232</b>
<b>11.1</b>	<b>Model validation test datasets</b>	<b>232</b>
<b>11.2</b>	<b>Datasets used for multivariate statistical process monitoring</b>	<b>233</b>
<b>11.3</b>	<b>Additional figures</b>	<b>234</b>
11.3.1	Fourier power spectrum plots	234
11.3.2	Linear autocorrelation plots	237
11.3.3	Average Mutual Information plots	241
11.3.4	False Nearest Neighbours plots	245
11.3.5	Correlation Dimension plots	255
11.3.6	LS-SVM modelling plots	263
11.3.7	RBF neural network modelling plots	267
11.3.8	Free-run prediction results obtained from multi-dimensional time series models	273
11.3.9	Wipfrag and Malvern modelling plots	277

## LIST OF TABLES

Table 1.1: Energy required for size reduction (Hayes, 1993).....	4
Table 2.1: Average chemical analysis of individual PGMs, Au, Cu, and Ni for the UG2 layer in the northwestern parts of the Bushveld Complex.....	19
Table 2.2: Average size distribution of PGM grains.....	20
Table 4.1: Measured variables for the Amandelbult UG2 FAG mill.....	64
Table 4.2: Summary of statistics calculated for the measured variables.....	65
Table 4.3: Summary of statistics calculated for data subsets.....	66
Table 4.4: Pearson's correlation coefficient matrix (based on the bivariate method) calculated for the Process data subset. Correlations given in <i>italics</i> are significant at the 0.01 level (2-tailed) with correlations shaded in grey discussed earlier in the text.....	71
Table 4.5: Linear Autocorrelation (LA), Average Mutual Information (AMI) and False Nearest Neighbours (FNN) statistics calculated for reduced subset data in table 4.3.....	75
Table 5.1: FAG mill variable classification.....	91
Table 5.2: Process dataset embedding parameters.....	92
Table 5.3: Summary of $R^2$ [%] statistics for validation and independent test datasets.....	93
Table 5.4: One-dimensional time series modelling parameters.....	97
Table 5.5: Summary of $R^2$ [%] statistics for validation and independent test datasets.....	98
Table 5.6: Lyapunov spectrum for the FAG mill power ( $x_1$ ) time series.....	104
Table 5.7: FAG mill variable classification.....	107
Table 5.8: Multi-dimensional time series modelling parameters.....	110
Table 5.9: Summary of $R^2$ [%] statistics for validation and independent test datasets.....	111
Table 5.10: LS-SVM modelling parameters.....	123
Table 5.11: Summary of $R^2$ [%] statistics for validation and independent test datasets.....	123
Table 5.12: RBF modelling parameters.....	128
Table 5.13: Summary of $R^2$ [%] statistics for validation and independent test datasets.....	129
Table 5.14: Summary of $R^2$ [%] statistics for validation and independent test datasets.....	129
Table 5.15: Summary of $R^2$ [%] statistics for validation and independent test datasets.....	144
Table 5.16: Embedding parameters.....	147
Table 5.17: Modelling parameters and results.....	149
Table 5.18: Lyapunov spectrum for the FAG mill power ( $x_1$ ) time series.....	154
Table 6.1: Upper confidence limits for respective charts.....	169
Table 6.2: Upper confidence limits for respective charts.....	176
Table 11.1: Summary of statistics calculated for test dataset 1 (6571 observations).....	232
Table 11.2: Summary of statistics calculated for test dataset 2 (6091 observations).....	232
Table 11.3: Summary of statistics calculated for test dataset 3 (4501 observations).....	233
Table 11.4: Summary of statistics calculated for reference dataset (1071 observations).....	233
Table 11.5: Summary of statistics calculated for test dataset 1 (1080 observations).....	233
Table 11.6: Summary of statistics calculated for test dataset 2 (1091 observations).....	233
Table 11.7: Summary of statistics calculated for test dataset 3 (1201 observations).....	234

## LIST OF FIGURES

Figure 2.1: Geological sketch map of the Bushveld Complex showing the location of RPM Amandelbult (Knoper, 2001).....	18
Figure 2.2: Amandelbult UG2 FAG mill section flow diagram.....	22
Figure 3.1: (a) The nearly periodic Mackey-Glass series, (b) the frequency spectrum showing primarily one frequency component, (c) the attractor (Kanjilal, 1995).....	42
Figure 3.2: (a) The quasiperiodic Mackey-Glass series, (b) the frequency spectrum showing primarily two bands of frequencies, (c) the attractor (Kanjilal, 1995).....	43
Figure 3.3: (a) The chaotic Mackey-Glass series, (b) the frequency spectrum showing relatively widely spread frequency components, (c) the attractor (Kanjilal, 1995).....	44
Figure 3.4: Typical linear transfer function (Demuth and Beale, 2001).....	48
Figure 3.5: Typical tan-sigmoid transfer function (Demuth and Beale, 2001).....	48
Figure 3.6: Structure of a typical MLP neural network.....	49
Figure 3.7: Typical radial basis transfer function (Demuth and Beale, 2001).....	53
Figure 4.1: Plot of FAG mill power ( $x_1$ ) and load ( $x_2$ ).....	67
Figure 4.2: Plot of FAG mill coarse feed rate ( $x_3$ ) and fine feed rate ( $x_4$ ).....	67

Figure 4.3: Plot of FAG mill total screen feed rate ( $x_5$ ) and feed rate ( $x_6$ ).....	68
Figure 4.4: Plot of FAG mill inlet water flow rate ( $x_7$ ) and motor reference speed ( $x_8$ ) .....	68
Figure 4.5: Plot of FAG mill discharge 1 flow rate ( $x_9$ ) and discharge 2 flow rate ( $x_{10}$ ).....	69
Figure 4.6: Plot of FAG mill rougher feed 1 flow rate ( $x_{11}$ ) and rougher feed 2 flow rate ( $x_{12}$ ) .....	69
Figure 4.7: Plot of FAG mill feed size distribution % -31.5mm ( $x_{13}$ ) and product size distribution % - 75 $\mu\text{m}$ ( $x_{14}$ ) .....	70
Figure 4.8: Fourier power spectrum for FAG mill coarse feed rate ( $x_3$ ) and FAG mill fine feed rate ( $x_4$ ) .....	74
Figure 4.9: Autocorrelation for FAG mill coarse feed rate ( $x_3$ ) .....	76
Figure 4.10: Autocorrelation for FAG mill rougher feed 2 flow rate ( $x_{12}$ ).....	77
Figure 4.11: Autocorrelation for FAG mill fine feed rate ( $x_4$ ) .....	78
Figure 4.12: Autocorrelation for FAG mill product size distribution % -75 $\mu\text{m}$ ( $x_{14}$ ) .....	79
Figure 4.13: AMI plot of FAG mill load ( $x_2$ ).....	80
Figure 4.14: AMI plot of FAG mill power ( $x_1$ ).....	81
Figure 4.15: AMI plot of FAG mill discharge 1 flow rate ( $x_9$ ) .....	81
Figure 4.16: FNN plot of FAG mill power ( $x_1$ ) with $k = 37$ .....	83
Figure 4.17: FNN plot of FAG mill fine feed rate ( $x_4$ ) with $k = 9$ .....	84
Figure 4.18: Correlation dimension curve for FAG mill power ( $x_1$ ; solid line) and its corresponding surrogate data sets (broken lines) with $k = 37$ and $m = 8$ .....	87
Figure 4.19: Correlation dimension curve for FAG mill fine feed rate ( $x_4$ ; solid line) and its corresponding surrogate data sets (broken lines) with $k = 9$ and $m = 12$ .....	87
Figure 5.1: One-step prediction of FAG mill power ( $x_1$ ) using <i>linearARX</i> ( $R^2 = 84.24\%$ ).....	93
Figure 5.2: Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using <i>linearARX</i> .....	94
Figure 5.3: Reconstructed attractor of FAG mill power ( $x_1$ ) based on MOD embedding parameters....	96
Figure 5.4: Reconstructed attractor of FAG mill power ( $x_1$ ) based on SSA embedding parameters.....	97
Figure 5.5: One-step prediction of FAG mill power ( $x_1$ ) using <i>MLP<sub>MOD_0</sub></i> ( $R^2 = 43.45\%$ ).....	99
Figure 5.6: One-step prediction of FAG mill power ( $x_1$ ) using <i>MLP<sub>SSA_0</sub></i> ( $R^2 = 82.21\%$ ) .....	99
Figure 5.7: Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using <i>MLP<sub>MOD_0</sub></i> .....	101
Figure 5.8: Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using <i>MLP<sub>SSA_0</sub></i> .....	101
Figure 5.9: Correlation dimension curve for FAG mill power ( $x_1$ ; solid line) and of the non-linear surrogates of <i>MLP<sub>MOD_0</sub></i> (broken lines).....	103
Figure 5.10: Correlation dimension curve for FAG mill power ( $x_1$ ; solid line) and of the non-linear surrogates of <i>MLP<sub>SSA_0</sub></i> (broken lines).....	103
Figure 5.11: Free-run prediction of FAG mill power ( $x_1$ ) using <i>MLP<sub>MOD_0</sub></i> .....	105
Figure 5.12: Free-run prediction of FAG mill power ( $x_1$ ) using <i>MLP<sub>SSA_0</sub></i> .....	105
Figure 5.13: Reconstructed attractor based on embedding strategy 2 .....	108
Figure 5.14: Reconstructed attractor based on embedding strategy 3 .....	109
Figure 5.15: Reconstructed attractor based on embedding strategy 4 .....	109
Figure 5.16: Reconstructed attractor based on embedding strategy 5 .....	110
Figure 5.17: One-step prediction of FAG mill power ( $x_1$ ) using <i>MLP<sub>none</sub></i> ( $R^2 = 81.94\%$ ) .....	111
Figure 5.18: One-step prediction of FAG mill power ( $x_1$ ) using <i>MLP<sub>MOD_1</sub></i> ( $R^2 = 37.72\%$ ).....	112
Figure 5.19: One-step prediction of FAG mill power ( $x_1$ ) using <i>MLP<sub>SSA_1</sub></i> ( $R^2 = 80.03\%$ ) .....	112
Figure 5.20: One-step prediction of FAG mill power ( $x_1$ ) using <i>MLP<sub>SSA_2</sub></i> ( $R^2 = 71.26\%$ ) .....	113
Figure 5.21: One-step prediction of FAG mill power ( $x_1$ ) using <i>MLP<sub>SSA_3</sub></i> ( $R^2 = 82.34\%$ ) .....	113
Figure 5.22: Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using <i>MLP<sub>none</sub></i> .....	114
Figure 5.23: Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using <i>MLP<sub>MOD_1</sub></i> .....	115
Figure 5.24: Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using <i>MLP<sub>SSA_1</sub></i> .....	115
Figure 5.25: Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using <i>MLP<sub>SSA_2</sub></i> .....	116
Figure 5.26: Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using <i>MLP<sub>SSA_3</sub></i> .....	116
Figure 5.27: Correlation dimension curve for FAG mill power ( $x_1$ ; solid line) and of the non-linear surrogates of <i>MLP<sub>MOD_1</sub></i> (broken lines).....	118

Figure 5.28: Correlation dimension curve for FAG mill power ( $x_1$ ; solid line) and of the non-linear surrogates of <i>MLPSSA_1</i> (broken lines).....	119
Figure 5.29: Correlation dimension curve for FAG mill power ( $x_1$ ; solid line) and of the non-linear surrogates of <i>MLPSSA_2</i> (broken lines).....	119
Figure 5.30: Correlation dimension curve for FAG mill power ( $x_1$ ; solid line) and of the non-linear surrogates of <i>MLPSSA_3</i> (broken lines).....	120
Figure 5.31: One-step prediction of FAG mill power ( $x_1$ ) using <i>SVM_3</i> ( $R^2 = 82.23\%$ ).....	124
Figure 5.32: Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using <i>SVM_3</i> .....	125
Figure 5.33: Free-run prediction of FAG mill power ( $x_1$ ) using <i>SVM_0</i> .....	126
Figure 5.34: One-step prediction of FAG mill power ( $x_1$ ) using <i>RFSSA_3</i> ( $R^2 = 80.60\%$ ).....	130
Figure 5.35: Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using <i>RFSSA_3</i> .....	130
Figure 5.36: Correlation dimension curve for FAG mill power ( $x_1$ ; solid line) and of the non-linear surrogates of <i>RFMOD_0</i> (broken lines).....	132
Figure 5.37: Correlation dimension curve for FAG mill power ( $x_1$ ; solid line) and of the non-linear surrogates of <i>RFSSA_0</i> (broken lines).....	132
Figure 5.38: Free-run prediction of FAG mill power ( $x_1$ ) using <i>RFMOD_0</i> .....	133
Figure 5.39: Free-run prediction of FAG mill power ( $x_1$ ) using <i>RFSSA_0</i> .....	134
Figure 5.40: Free-run prediction using <i>RFSSA_0</i> ; correctly predicted an increase in FAG mill power ( $x_1$ ) over the prediction interval.....	135
Figure 5.41: Free-run prediction using <i>RFSSA_0</i> ; correctly predicted a decrease in FAG mill power ( $x_1$ ) over the prediction interval.....	135
Figure 5.42: Free-run prediction using <i>RFSSA_0</i> ; incorrectly predicted a decrease in FAG mill power ( $x_1$ ) over the prediction interval.....	136
Figure 5.43: Free-run prediction using <i>RFSSA_0</i> ; incorrectly predicted a decrease in FAG mill power ( $x_1$ ) over the prediction interval.....	136
Figure 5.44: Free-run prediction using <i>linearARX</i> .....	138
Figure 5.45: Free-run prediction using <i>MLPnone</i> .....	139
Figure 5.46: Free-run prediction using <i>MLPSSA_3</i> .....	139
Figure 5.47: Free-run prediction using <i>SVMnone</i> .....	140
Figure 5.48: Free-run prediction using <i>SVM_3</i> .....	140
Figure 5.49: Free-run prediction using <i>RFnone</i> .....	141
Figure 5.50: Free-run prediction using <i>RFSSA_3</i> .....	141
Figure 5.51: Reconstructed attractor of data using an embedding dimension of 20 and an embedding delay of unity.....	143
Figure 5.52: Plot of FAG mill discharge 2 flow rate ( $x_{10}$ ) and rougher feed 2 flow rate ( $x_{12}$ ).....	144
Figure 5.53: One-step prediction of FAG mill rougher feed 2 flow rate ( $x_{12}$ ) using <i>SVM_5</i> ( $R^2 = 77.89\%$ ).....	145
Figure 5.54: Normal P-P plot of the residual values for one-step prediction of the FAG mill rougher feed 2 flow rate ( $x_{12}$ ) using <i>SVM_5</i> .....	146
Figure 5.55: Reconstructed attractor of FAG mill feed size distribution (% -31.5mm; $x_{13}$ ).....	148
Figure 5.56: Reconstructed attractor of FAG mill product size distribution (% -75 $\mu\text{m}$ ; $x_{14}$ ).....	148
Figure 5.57: One-step prediction of FAG mill feed size distribution (% -31.5mm; $x_{13}$ ) ( $R^2 = 72.15\%$ ).....	150
Figure 5.58: One-step prediction of FAG mill product size distribution (% -75 $\mu\text{m}$ ; $x_{14}$ ) ( $R^2 = -21.92\%$ ).....	150
Figure 5.59: Normal P-P plot of the residual values for one-step prediction of the FAG mill feed size distribution (% -31.5mm; $x_{13}$ ).....	151
Figure 5.60: Normal P-P plot of the residual values for one-step prediction of the FAG mill product size distribution (% -75 $\mu\text{m}$ ; $x_{14}$ ).....	152
Figure 5.61: Correlation dimension curve for FAG mill feed size distribution (% -31.5mm; $x_{13}$ ; solid line) and of the non-linear surrogates (broken lines).....	153
Figure 5.62: Correlation dimension curve for FAG mill product size distribution (% -75 $\mu\text{m}$ ; $x_{14}$ ; solid line) and of the non-linear surrogates (broken lines).....	154
Figure 6.1: On-line model validation methodology.....	159
Figure 6.2: $T^2$ -chart indicating abnormal process behaviour (Kharva, 2001).....	163
Figure 6.3: SPE-chart indicating abnormal process behaviour (Kharva, 2001).....	165
Figure 6.4: Score-control chart indicating abnormal process behaviour.....	166
Figure 6.5: $T^2$ - and SPE-charts for model training data.....	169

Figure 6.6: Combined score- and SPE-chart for model training data.....	170
Figure 6.7: T <sup>2</sup> - and SPE-charts for model validation data.....	171
Figure 6.8: Combined score- and SPE-chart for model validation data (red crosses) in comparison to the model training data (blue circles).....	171
Figure 6.9: T <sup>2</sup> - and SPE-charts for model test dataset 1.....	172
Figure 6.10: Combined score- and SPE-chart for model test dataset 1 (red crosses) in comparison to the model training data (blue circles).....	172
Figure 6.11: T <sup>2</sup> - and SPE-charts for model test dataset 2.....	173
Figure 6.12: Combined score- and SPE-chart for model test dataset 2 (red crosses) in comparison to the model training data (blue circles).....	173
Figure 6.13: T <sup>2</sup> - and SPE-charts for model test dataset 3.....	174
Figure 6.14: Combined score- and SPE-chart for model test dataset 3 (red crosses) in comparison to the model training data (blue circles).....	174
Figure 6.15: T <sup>2</sup> - and SPE-charts for the reference dataset.....	177
Figure 6.16: Combined score- and SPE-chart for the reference dataset.....	177
Figure 6.17: T <sup>2</sup> - and SPE-charts for test dataset 1.....	178
Figure 6.18: Individual variable SPE-charts for test dataset 1; FAG mill power ( $x_1$ ), FAG mill load ( $x_2$ ), FAG mill fine feed rate ( $x_4$ ) and FAG mill inlet water flow rate ( $x_7$ ).....	178
Figure 6.19: Combined score- and SPE-chart for test dataset 1 (red crosses) in comparison to the reference dataset (blue circles).....	179
Figure 6.20: T <sup>2</sup> - and SPE-charts for test dataset 2.....	180
Figure 6.21: Individual variable SPE-charts for test dataset 2; FAG mill power ( $x_1$ ), FAG mill load ( $x_2$ ), FAG mill fine feed rate ( $x_4$ ) and FAG mill inlet water flow rate ( $x_7$ ).....	181
Figure 6.22: Combined score- and SPE-chart for test dataset 2 (red crosses) in comparison to the reference dataset (blue circles).....	181
Figure 6.23: T <sup>2</sup> - and SPE-charts for test dataset 3.....	183
Figure 6.24: Individual variable SPE-charts for test dataset 3; FAG mill power ( $x_1$ ), FAG mill load ( $x_2$ ), FAG mill fine feed rate ( $x_4$ ) and FAG mill inlet water flow rate ( $x_7$ ).....	183
Figure 6.25: Combined score- and SPE-chart for test dataset 3 (red crosses) in comparison to the reference dataset (blue circles).....	184
Figure 7.1: Process responses for FAG mill power ( $x_1$ ) initially above 3000 [kW].....	195
Figure 7.2: Process responses for FAG mill power ( $x_1$ ) initially below 3000 [kW].....	196
Figure 11.1: Fourier power spectrum for FAG mill power ( $x_1$ ) and load ( $x_2$ ).....	234
Figure 11.2: Fourier power spectrum for FAG mill total screen feed rate ( $x_5$ ) and feed rate ( $x_6$ ).....	235
Figure 11.3: Fourier power spectrum for FAG mill inlet water flow rate ( $x_7$ ) and motor reference speed ( $x_8$ ).....	235
Figure 11.4: Fourier power spectrum for FAG mill discharge 1 flow rate ( $x_9$ ) and discharge 2 flow rate ( $x_{10}$ ).....	236
Figure 11.5: Fourier power spectrum for FAG mill rougher feed 1 flow rate ( $x_{11}$ ) and rougher feed 2 flow rate ( $x_{12}$ ).....	236
Figure 11.6: Fourier power spectrum for FAG mill feed size distribution % -31.5mm ( $x_{13}$ ) and product size distribution % -75 $\mu$ m ( $x_{14}$ ).....	237
Figure 11.7: Autocorrelation for FAG mill power ( $x_1$ ).....	237
Figure 11.8: Autocorrelation for FAG mill load ( $x_2$ ).....	238
Figure 11.9: Autocorrelation for FAG mill inlet water flow rate ( $x_7$ ).....	238
Figure 11.10: Autocorrelation for FAG mill discharge 1 flow rate ( $x_9$ ).....	239
Figure 11.11: Autocorrelation for FAG mill discharge 2 flow rate ( $x_{10}$ ).....	239
Figure 11.12: Autocorrelation for FAG mill rougher feed 1 flow rate ( $x_{11}$ ).....	240
Figure 11.13: Autocorrelation for FAG mill feed size distribution % -31.5mm ( $x_{13}$ ).....	240
Figure 11.14: AMI plot of FAG mill coarse feed rate ( $x_3$ ).....	241
Figure 11.15: AMI plot of FAG mill fine feed rate ( $x_4$ ).....	241
Figure 11.16: AMI plot of FAG mill inlet water flow rate ( $x_7$ ).....	242
Figure 11.17: AMI plot of FAG mill discharge 2 flow rate ( $x_{10}$ ).....	242
Figure 11.18: AMI plot of FAG mill rougher feed 1 flow rate ( $x_{11}$ ).....	243
Figure 11.19: AMI plot of FAG mill rougher feed 2 flow rate ( $x_{12}$ ).....	243
Figure 11.20: AMI plot of FAG mill feed size distribution % -31.5mm ( $x_{13}$ ).....	244
Figure 11.21: AMI plot of FAG mill product size distribution % -75 $\mu$ m ( $x_{14}$ ).....	244
Figure 11.22: FNN plot of FAG mill power ( $x_1$ ) with $k = 100$ .....	245
Figure 11.23: FNN plot of FAG mill load ( $x_2$ ) with $k = 106$ .....	245
Figure 11.24: FNN plot of FAG mill load ( $x_2$ ) with $k = 100$ .....	246

Figure 11.25: FNN plot of FAG mill coarse feed rate ( $x_3$ ) with $k = 240$ .....	246
Figure 11.26: FNN plot of FAG mill coarse feed rate ( $x_3$ ) with $k = 100$ .....	247
Figure 11.27: FNN plot of FAG mill fine feed rate ( $x_4$ ) with $k = 16$ .....	247
Figure 11.28: FNN plot of FAG mill inlet water flow rate ( $x_7$ ) with $k = 42$ .....	248
Figure 11.29: FNN plot of FAG mill inlet water flow rate ( $x_7$ ) with $k = 42$ .....	248
Figure 11.30: FNN plot of FAG mill discharge 1 flow rate ( $x_9$ ) with $k = 1$ .....	249
Figure 11.31: FNN plot of FAG mill discharge 1 flow rate ( $x_9$ ) with $k = 4$ .....	249
Figure 11.32: FNN plot of FAG mill discharge 2 flow rate ( $x_{10}$ ) with $k = 3$ .....	250
Figure 11.33: FNN plot of FAG mill discharge 2 flow rate ( $x_{10}$ ) with $k = 4$ .....	250
Figure 11.34: FNN plot of FAG mill rougher feed 1 flow rate ( $x_{11}$ ) with $k = 3$ .....	251
Figure 11.35: FNN plot of FAG mill rougher feed 1 flow rate ( $x_{11}$ ) with $k = 5$ .....	251
Figure 11.36: FNN plot of FAG mill rougher feed 2 flow rate ( $x_{12}$ ) with $k = 3$ .....	252
Figure 11.37: FNN plot of FAG mill rougher feed 2 flow rate ( $x_{12}$ ) with $k = 6$ .....	252
Figure 11.38: FNN plot of FAG mill feed size distribution % -31.5mm ( $x_{13}$ ) with $k = 22$ .....	253
Figure 11.39: FNN plot of FAG mill feed size distribution % -31.5mm ( $x_{13}$ ) with $k = 32$ .....	253
Figure 11.40: FNN plot of FAG mill product size distribution % -75 $\mu\text{m}$ ( $x_{14}$ ) with $k = 1$ .....	254
Figure 11.41: FNN plot of FAG mill product size distribution % -75 $\mu\text{m}$ ( $x_{14}$ ) with $k = 2$ .....	254
Figure 11.42: Correlation dimension curve for FAG mill power ( $x_1$ ; solid line) and its corresponding surrogate data sets (broken lines) with $k = 100$ and $m = 6$ .....	255
Figure 11.43: Correlation dimension curve for FAG mill load ( $x_2$ ; solid line) and its corresponding surrogate data sets (broken lines) with $k = 106$ and $m = 6$ .....	255
Figure 11.44: Correlation dimension curve for FAG mill load ( $x_2$ ; solid line) and its corresponding surrogate data sets (broken lines) with $k = 100$ and $m = 8$ .....	256
Figure 11.45: Correlation dimension curve for FAG mill coarse feed rate ( $x_3$ ; solid line) and its corresponding surrogate data sets (broken lines) with $k = 240$ and $m = 2$ .....	256
Figure 11.46: Correlation dimension curve for FAG mill coarse feed rate ( $x_3$ ; solid line) and its corresponding surrogate data sets (broken lines) with $k = 100$ and $m = 5$ .....	257
Figure 11.47: Correlation dimension curve for FAG mill fine feed rate ( $x_4$ ; solid line) and its corresponding surrogate data sets (broken lines) with $k = 16$ and $m = 18$ .....	257
Figure 11.48: Correlation dimension curve for FAG mill inlet water flow rate ( $x_7$ ; solid line) and its corresponding surrogate data sets (broken lines) with $k = 42$ and $m = 8$ .....	258
Figure 11.49: Correlation dimension curve for FAG mill inlet water flow rate ( $x_7$ ; solid line) and its corresponding surrogate data sets (broken lines) with $k = 42$ and $m = 8$ .....	258
Figure 11.50: Correlation dimension curve for FAG mill discharge 2 flow rate ( $x_{10}$ ; solid line) and its corresponding surrogate data sets (broken lines) with $k = 3$ and $m = 15$ .....	259
Figure 11.51: Correlation dimension curve for FAG mill discharge 2 flow rate ( $x_{10}$ ; solid line) and its corresponding surrogate data sets (broken lines) with $k = 4$ and $m = 15$ .....	259
Figure 11.52: Correlation dimension curve for FAG mill rougher feed 2 flow rate ( $x_{12}$ ; solid line) and its corresponding surrogate data sets (broken lines) with $k = 3$ and $m = 11$ .....	260
Figure 11.53: Correlation dimension curve for FAG mill rougher feed 2 flow rate ( $x_{12}$ ; solid line) and its corresponding surrogate data sets (broken lines) with $k = 6$ and $m = 11$ .....	260
Figure 11.54: Correlation dimension curve for FAG mill feed size distribution % -31.5mm ( $x_{13}$ ; solid line) and its corresponding surrogate data sets (broken lines) with $k = 22$ and $m = 7$ .....	261
Figure 11.55: Correlation dimension curve for FAG mill feed size distribution % -31.5mm ( $x_{13}$ ; solid line) and its corresponding surrogate data sets (broken lines) with $k = 32$ and $m = 7$ .....	261
Figure 11.56: Correlation dimension curve for FAG mill product size distribution % -75 $\mu\text{m}$ ( $x_{14}$ ; solid line) and its corresponding surrogate data sets (broken lines) with $k = 1$ and $m = 7$ .....	262
Figure 11.57: Correlation dimension curve for FAG mill product size distribution % -75 $\mu\text{m}$ ( $x_{14}$ ; solid line) and its corresponding surrogate data sets (broken lines) with $k = 2$ and $m = 7$ .....	262
Figure 11.58: One-step prediction of FAG mill power ( $x_1$ ) using <i>SVMnone</i> ( $R^2 = 84.17\%$ ) .....	263
Figure 11.59: Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using <i>SVMnone</i> .....	263
Figure 11.60: One-step prediction of FAG mill power ( $x_1$ ) using <i>SVM_0</i> ( $R^2 = 85.46\%$ ) .....	264
Figure 11.61: Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using <i>SVM_0</i> .....	264
Figure 11.62: One-step prediction of FAG mill power ( $x_1$ ) using <i>SVM_1</i> ( $R^2 = 85.82\%$ ) .....	265
Figure 11.63: Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using <i>SVM_1</i> .....	265
Figure 11.64: One-step prediction of FAG mill power ( $x_1$ ) using <i>SVM_4</i> ( $R^2 = 85.21\%$ ) .....	266

Figure 11.65: Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using <i>SVM_4</i> .....	266
Figure 11.66: One-step prediction of FAG mill power ( $x_1$ ) using <i>RBFFnone</i> ( $R^2 = 77.99\%$ ).....	267
Figure 11.67: Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using <i>RBFFnone</i> .....	267
Figure 11.68: One-step prediction of FAG mill power ( $x_1$ ) using <i>RBFFMOD_0</i> ( $R^2 = 46.37\%$ ).....	268
Figure 11.69: Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using <i>RBFFMOD_0</i> .....	268
Figure 11.70: One-step prediction of FAG mill power ( $x_1$ ) using <i>RBFFSSA_0</i> ( $R^2 = 82.41\%$ ) .....	269
Figure 11.71: Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using <i>RBFFSSA_0</i> .....	269
Figure 11.72: One-step prediction of FAG mill power ( $x_1$ ) using <i>RBFFMOD_1</i> ( $R^2 = 39.91\%$ ).....	270
Figure 11.73: Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using <i>RBFFMOD_1</i> .....	270
Figure 11.74: One-step prediction of FAG mill power ( $x_1$ ) using <i>RBFFSSA_1</i> ( $R^2 = 82.69\%$ ) .....	271
Figure 11.75: Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using <i>RBFFSSA_1</i> .....	271
Figure 11.76: One-step prediction of FAG mill power ( $x_1$ ) using <i>RBFFSSA_2</i> ( $R^2 = 78.44\%$ ) .....	272
Figure 11.77: Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using <i>RBFFSSA_2</i> .....	272
Figure 11.78: Free-run prediction using <i>MLPMD_1</i> .....	273
Figure 11.79: Free-run prediction using <i>MLPSSA_1</i> .....	273
Figure 11.80: Free-run prediction using <i>MLPSSA_2</i> .....	274
Figure 11.81: Free-run prediction using <i>SVM_1</i> .....	274
Figure 11.82: Free-run prediction using <i>SVM_4</i> .....	275
Figure 11.83: Free-run prediction using <i>RBFFMOD_1</i> .....	275
Figure 11.84: Free-run prediction using <i>RBFFSSA_1</i> .....	276
Figure 11.85: Free-run prediction using <i>RBFFSSA_2</i> .....	276
Figure 11.86: Free-run prediction of FAG mill product size distribution (% -75 $\mu\text{m}$ ; $x_{14}$ ) .....	277
Figure 11.87: Free-run prediction of FAG mill feed size distribution (% -31.5mm; $x_{13}$ ).....	277

## LIST OF KEY SYMBOLS USED

### Roman Alphabet

<i>Symbol</i>	<i>Description</i>	<i>Units</i>
$a$	Number of principal components retained	-
$b$	Number of model inputs	-
$c$	Radial basis transfer function centre	-
$C_N$	Correlation function	-
$d$	Fractal dimension	-
$d_c$	Correlation dimension	-
$d_M$	Model order	-
$e$	Error	-
$E$	Residual matrix	-
$f$	Number of model outputs	-
$F_{\alpha}(a, n-a)$	Upper 100 $\alpha$ % critical point of the $F$ -distribution	-
$g$	Gradient	-
$h$	Degrees of freedom	-
$H$	Hessian matrix	-
$I(\cdot)$	Heaviside function	-
$J$	Jacobian matrix	-
$k$	Delay time / embedding lag	-
$K(x, x_k)$	Kernel function	-
$l_i$	Loadings (eigenvectors)	-
$m$	Embedding dimension	-
$N$	Number of observations in time series	-
$p(x)$	Probability	-
$P(x)$	Probability density	-
$q(\cdot)$	Polynomial function	-
$Q$	Lack of model fit statistic	-

$r$	Radial basis transfer function radius	-
$\mathbb{R}^m$	Euclidean space	-
$s_{t_i}^2$	Estimated variance of $t_i$	-
$S$	Sum-squared-error	-
$t_i$	Scores	-
$w$	Network weight	-
$[w]$	Weight vector	-
$[x]$	State vector / input vector	-
$\bar{x}$	Mean of the entire time series	-
$y$	Observed output	-
$\hat{y}$	Network output	-
$z$	Node output	-

### Greek Alphabet

<i>Symbol</i>	<i>Description</i>	<i>Units</i>
$\alpha$	Significance level	-
$\alpha_k$	Lagrangian multipliers (support vectors)	-
$\varepsilon$	Error	-
$\gamma$	Effective number of model parameters	-
$\lambda$	Regularisation parameter	-
$\lambda_1$	First Lyapunov exponent	-
$\lambda_i$	Eigenvalue	-
$\varphi(\cdot)$	Non-linear mapping	-
$\tau_s$	Sampling time	-
$\tau_w$	Time window length	-
$\sigma$	Standard deviation	-

**LIST OF KEY ABBREVIATIONS USED**

<i>Abbreviation</i>	<i>Description</i>
AMI	Average mutual information
BIC	Bayesian information criterion
EA	Evolutionary algorithm
ERM	Empirical risk minimisation
FAG	Fully autogenous
FFT	Fast Fourier transform
FNN	False nearest neighbours
LA	Linear autocorrelation
LS-SVM	Least squares support vector machine
MARS	Multiple adaptive regression splines
MLP	Multi-layer perceptron
MOD	Method of delays
MPC	Model predictive control
MSE	Mean square error
PCA	Principal component analysis
PGM	Platinum group metal
RBF	Radial basis function
RMS	Root mean square
RPM	Rustenburg Platinum Mines
SANE	Symbiotic, adaptive neuro-evolution
SIC	Schwartz information criterion
SPE	Squared prediction error
SPM	Statistical process monitoring
SRM	Structural risk minimisation
SSA	Singular spectrum analysis
SSE	Modified sum squared error
SVD	Singular value decomposition
UCL	Upper confidence limit

# 1 Modelling and control of comminution circuits

---

All mineral processing plants are operated with the main objective of separating the feed material, usually an ore, into a product containing valuable material and a residue containing less valuable material or gangue. The plant is usually composed of several process units selected from a wide range depending on the characteristics of the feed to be processed and the design capacity of the plant. These metallurgical processes are often high-dimensional and non-linear, making them difficult to understand, model and control. Owing to these and other unusual difficulties, automatic process control of mineral processing plants has always been less advanced when compared to most chemical processing plants.

During the month of June 2000, Anglo Platinum commissioned a new UG2 circuit at its Rustenburg Platinum Mines (RPM) Amandelbult concentrator, adding 60 000 tons per month to the UG2 circuit's production capacity. This circuit, unlike previous circuits, was far more complex in its design owing to its separate treatment of the naturally fine UG2 ore. This meant that at the time no effective control strategy existed for the circuit, necessitating the investigation of the non-linear relationships among the mill variables to assist in the development of an optimal advanced control philosophy for the plant.

In order to better understand the importance of and economic benefit that may be obtained from modelling the mill and the subsequent application of an advanced process control system to the mill, some insight into mineral beneficiation, comminution and, more specifically, milling is required.

## 1.1 Mineral beneficiation

Because most metals are found in nature as low-grade, complex ore mixtures, pre-treatment of the ore, in the form of mineral beneficiation, is required to remove and discard a large part of the associated valueless waste rock prior to the more expensive recovery and separation of metal values by pyrometallurgical or hydrometallurgical methods.

Ore size reduction is the first step in mineral beneficiation and is usually done progressively. It starts off with the mining operation itself whereby the ore body is reduced from its natural size, which may be a kilometre or more in extent, down to run-of-mine ore mainly by use of explosives. This ore consists of particles having an extremely wide range of sizes, from lumps that may be as large as a metre or more in diameter down to less than a micron in size. Mining is followed by even more size reduction which is done primarily to reduce the lumpy condition of run-of-mine ore so that it is easier to handle, to ensure that the sizes of the particles are in the range required by the downstream processing operations, and to liberate or expose minerals. The control of the size of particles is exercised through comminution that entails the breaking of particles by mechanical means into smaller fragments. During comminution size reduction of large particles is normally achieved by crushing, whereas for small particles milling is used.

Size reduction is followed by the separation of the values (concentrate) from the waste material (tailings), often making use of physical properties such as specific gravity differences or differences in magnetic permeability between the values and the waste. Cleaning and recovery of the various products, by means of dewatering, screening, etc., is the last stage in mineral beneficiation.

Mineral beneficiation is economically easily justified. The physical-mechanical processes used are much simpler and less expensive than the pyrometallurgical or hydrometallurgical processes, making the valuable content of the ore available at less cost for further treatment. Typical economic benefits of beneficiation preceding pyrometallurgical treatment include (Gill, 1991):

- Lower transportation and handling costs.
- A reduction in smelting costs resulting from a reduced amount of tonnages to treat, less fuel required and less fluxes needed for slag making.
- Increased furnace capacity leading to more space being available for metal production.
- Less slag produced which could carry off metal values.

It must, however, be remembered that the total cost of beneficiation is the cost of making the concentrate plus the cost of the values lost in the discarded tailings.

## **1.2 Economics of comminution**

Comminution usually consists of both crushing and milling. Crushing, normally a dry process, is accomplished by compression of the ore against rigid surfaces or by impact against surfaces in a rigidly constrained motion path whereas milling, normally a wet process, is accomplished by abrasion and impact of the ore by the free motion of unconnected media.

One of the major objectives of comminution is the liberation of the valuable minerals from the associated gangue minerals. In practice complete liberation of the minerals is seldom achieved even if the ore is ground down to the grain size of the desired mineral particles. Ores are therefore only ground to a predetermined optimum mesh of grind to produce an economic degree of liberation. In some instances it may even be economic to grind to a size much coarser than the optimum, producing a large middlings (particles of locked mineral and gangue) fraction in subsequent concentration processes that can then be reground before being subjected to final concentration processes.

Comminution forms a large part of any mineral processing plant's capital and operating cost. From table 1.1 it can be seen that the energy consumed per tonne of feed varies typically with the mean particle size of the product (Hayes, 1993). Cohen (1983) estimated that comminution consumed 30-50% of the total power drawn by a plant, and up to 70% for the treatment of hard ores. The operating cost of a comminution circuit for a typical metalliferous concentrator, as quoted by Wills (1992), was found to be approximately 50% of the total operating cost, with similar figures being inferred from the operating data for a range of metalliferous concentrators given by Weiss (1985). Capital cost figures vary, but lie in the range of 20-50% for most mixed-process plants (Napier-Munn et al., 1999). However, when analysing the balance between production costs and returns from smelting a typical metallic ore, Taggart (1951) found that much economic benefit is derived from milling the ore prior to smelting.

**Table 1.1: Energy required for size reduction (Hayes, 1993)**

Mean particle size of product	[kWh]/(tonne of feed)
5 – 10 [mm]	1 – 2
0.5 – 1 [mm]	3 – 4
80 – 100 [ $\mu\text{m}$ ]	10
60 – 80 [ $\mu\text{m}$ ]	20
40 – 50 [ $\mu\text{m}$ ]	30
20 – 30 [ $\mu\text{m}$ ]	45
- 10 [ $\mu\text{m}$ ]	70 – 80

Based on these statistics it is evident that much is to be gained from improving the practice of comminution. Improvements can be of two kinds (Napier-Munn et al., 1999):

- Fundamental changes in the technology, or the introduction of novel technology.
- Incremental improvements in the technology, its application and operating practice.

Typical improvements to comminution circuits include: optimisation of current process equipment; application of new process equipment; application of cutting edge instrumentation; and the application of advance process control. Possible benefits incurred by these improvements are reduced unit operating costs, increased throughput, and improved downstream process performance as a result of improved feed size specification.

### **1.3 Importance of milling**

Milling is the most common method used to reduce the size of particles below about 1 [mm]. The particle breakage required during milling is achieved through a tumbling action, brought about by the rotation of the mill about its axis, of the feed material with a loose grinding medium inside the mill. The grinding medium could be either foreign media added to the mill or the feed material itself. The mill is usually cylindrical in shape, with the mill load occupying 40-50% of the mill volume (Napier-Munn et al., 1999). Milling is generally done in a wet environment and water is

added with the feed material to make a slurry with a solids concentration of about 50% by volume (Woollacott and Eric, 1994).

The milling types most frequently used are rod, ball (invented by Bruckner in Germany, 1876), autogenous, pebble, and semi-autogenous milling. Rod and ball mills use grinding media consisting of steel rods and balls respectively. In contrast with this, autogenous mills use the material particles themselves as grinding media. Pebble mills are also a type of autogenous mill but with the grinding media being selected more carefully and a significant portion of the size reduction of the ore being done in a previous size reduction operation. Semi-autogenous mills on the other hand are a combination of ball and autogenous mills using only a small portion of steel balls, usually in the range of 5-10% of the mill volume. Selection of mill type to be used depends on the characteristics of the ore to be milled as well as the required size reduction to be obtained.

The main drawback of milling lies with the fact that most of the energy input to the mill is absorbed by the mill itself and only a small portion of the total energy is available for the breaking of the material. Lowrison (1974) showed that less than 1% of the total energy input to a ball mill is available for size reduction of the feed material, with the bulk of the energy being utilised in the production of heat. Other problems commonly experienced during milling are under- and overgrinding of the ore. Undergrinding results in a too coarse product, with a degree of liberation too low for economic separation. Overgrinding reduces the particle size of both the gangue and the mineral value below the size required for efficient separation while simultaneously wasting expensive, under utilized energy.

Efficient, manual control of milling circuits on a minute-to-minute basis so that no matter how the ore entering the system varies, or the physical conditions of the processing machines change, the product characteristics remain at or close to the required characteristics, is extremely difficult. Some means of automatic control is therefore necessary if continually good circuit performance is to be ensured.

## 1.4 Control of a milling circuit

Milling circuits are generally designed and controlled based on three important criteria (Woollacott and Eric, 1994):

1. Maintaining or maximising a desired production rate.
2. Producing a product of desired size.
3. Minimising costs.

In order to maximise the production rate of fine material from the mill, the power drawn by the mill for breakage processes must be maximised. This is normally done through mill load optimisation, with particular emphasis on the mass of grinding medium in the mill. Controlling the size distribution of the mill product is often more difficult when compared to maximising the production rate. The difficulty herein lies with the fact that the breakage process in a mill is stochastic. Consequently, there is no absolute control on the maximum size of broken particles in the mill and at the same time as large particles are being broken down into fragments of the required size, particles that are already fine enough are being broken into fragments that are unnecessarily small.

The control of a mill can be affected by the following principal variables: changes in the feed rate; size distribution and hardness of the feed; rate of water addition to the mill; and interruptions in circuit operation (Wills, 1979). Of these, fluctuations in the feed size and hardness of the ore are probably the most significant factors in disturbing the balance of the milling circuit. These fluctuations could arise from differences in composition, mineralisation, grain size and crystallisation of the ore from different parts of the mine, from changes in the crusher settings, or from damaged screens in the crushing circuit. An increase in feed size or ore hardness produces a coarser mill product, whereas an increase in mill throughput is possible for a decrease in feed size and ore hardness. Both these conditions have a significant effect on the mill-circulating load, making its control important in controlling the mill product size.

Control of these and other process variables are generally achieved using either a simple linear PID controller or a variation thereof. Owing to the fact that PID controllers are structurally simple, reliable and have a favourable ratio between performance and cost (Aström and Hägglund, 1995), they are employed in as many as 95% of all control loops in the process industry (Brambilla et al., 1989). PID controllers also offer the opportunity for simplified modelling requirements, decreased development effort, and requires a lower user skill level, all of which are issues of substantial importance to engineering practice (Hanley, 1990; Lee and Weekman, 1976). For grinding control, SISO PID control loops, are particularly useful, if the loops are decoupled and the PIDs are well tuned (Maciejowski, 1990).

Although there is widespread application of PID controllers in the process industry, PID controllers have some drawbacks. These drawbacks include the fact that their performance is limited by the properties of the fixed structure, the input-output pairing problem has to be solved for multivariable systems, and an efficient tuning method has to be used to obtain the desired dynamics of the controlled loops (Pomerleau et al., 2000). Adusumilli and Rivera (1997) showed clearly that in the presence of significant noise or drifting disturbances, most established techniques for PID controller tuning from plant data would fail.

Based on these drawbacks and owing to the complexities inherent to the control of milling circuits, the current trend is to use control strategies with some degree of intelligence, such as expert systems, fuzzy logic controllers and controllers based on neural networks (Duarte et al., 2001).

## **1.5 Intelligent control**

Since their introduction, rule-based systems for diagnosis and control (Muratet and Bourseau, 1993) have been the most important branch of intelligent control. However, in recent years their popularity has decreased due to their limitations and applicability to ill-defined systems. These knowledge-based expert systems consist essentially of methods for the maintenance of a database of facts and relationships, as well as structured routines for searching the database as efficiently as possible. These systems have made important contributions to the efficient development of complex

systems and have reduced the technical level of expertise required for the development of such systems (Klimasauskas, 1992).

Jamsa (1988) found that an expert control system designed for and installed at the Siilinjarvi grinding circuit could successfully control important variables such as the hydrocyclone feed density and the final particle size or the hydrocyclone feed mass flow. However, the system was only implemented as a tool for the operator to analyse the significant grinding process alarms and the off-line identification of the optimum process conditions. Ynchausti and Hales (1992) successfully implemented an object oriented expert control system on a closed-circuit, post-classification ball mill grinding circuit to either maximise the production rate or minimise the grind within given processing limits and bounds. They found that the benefits of such a supervisory control system could amount to a considerable increase in revenue per year.

It must be noted that these systems are not concerned with the creation of new knowledge, but only with making the available knowledge usable. The main problem with this approach is the bottleneck caused by knowledge engineering. Much time is usually spent talking to operators and understanding what they do and how they deal with emergency situations. Knowledge-based expert systems are also generally inadequate for complex tasks (Mylopoulos et al., 1992) and the successful application thereof depends heavily on both the accuracy and precision of the knowledge base (Pouskouleli and Wheat, 1992).

Similar to knowledge-based expert systems, fuzzy logic systems also use IF-THEN rules to define a particular set of input states, and provide descriptions of the consequences if those particular states were to prevail. The difference between these two systems arise from the fact that fuzzy logic systems attach numerical values to the antecedent of rules to denote the extent to which the premises are valid (Berardinis, 1992). Fuzzy logic systems are especially attractive in closed-loop systems where they eliminate the need for complex mathematical models typically required in more conventional computing and have proved to be the quickest and most inexpensive approach to automated control systems (Aldrich et al., 1993). Since these systems are

based on inferencing, they are also generally much less complicated than other intelligent control systems.

Harris and Meech (1987) successfully tested the use of fuzzy logic for the automatic control of industrial processes on a dynamic simulator of a secondary crushing plant. The nature of the controller made it easy to employ and understand, therefore not requiring management by personnel with special technical skills. It also showed the potential to improve plant throughput above that achieved by manual control for an equivalent level of power utilisation. By allowing the fuzzy set definitions to be adjusted on-line, Meech and Jordan (1993) later showed that a significant performance improvement in the control and operation of the simulated secondary crusher could be obtained by using the current draw and chamber level probes as the regulating variables. Moshgbar et al. (1995) have also demonstrated the use of fuzzy logic for the modelling of cone crusher wear.

Nonetheless, it must be noted that although fuzzy control in principle shows that it may be used to compensate for non-linear plant behaviour, non-linear plants are often linearised in order to provide a uniform dynamic behaviour (Langari, 1993). Kravaris and Arkun (1991) found that linear approximations of the process dynamics around the operating steady state could be unacceptably poor in the presence of strong non-linearities and even lead to the fundamental breakdown of linear control methods.

Artificial neural networks, in contrast with knowledge-based and fuzzy logic systems, are non-linear and generate their own rules by learning from examples. Items of knowledge are distributed across the network, and reasonable responses are obtained when the net is presented with incomplete, noisy or previously unseen inputs. The pattern recognition and generalisation capabilities of artificial neural networks make them much more attractive than the symbol-manipulation methodology of expert systems, especially as far as complex, ill-defined systems are concerned (Aldrich et al., 1993).

Although the application of artificial neural networks for the control of grinding processes is still in its infancy, with no commercially viable artificial neural network currently being used for control in the mineral processing industry, much work has

been done on its development and testing on dynamic simulators. Flament et al. (1992) have developed both a direct and indirect neural controller trained on process input and output differential data. Both these neural controllers were tested on the DYNAFRAG grinding process dynamic simulator (Dubé et al., 1987). It was found that although the individual controllers did not perform better than a regular PI controller for the control strategy illustrated, by combining them with an adaptation module, their response times were significantly improved following set-point changes or process disturbances. Stange (1993) expanded this control strategy developed by Flament et al. (1992) through the integration of a neural network model of the hydrocyclone classifier with the control strategy. The purpose of this additional model was to compensate for feed disturbances when tested on a simulated autogenous circuit (Stange, 1993). However, it was found that the inclusion of the additional hydrocyclone model into the control strategy did not allow for the successful control of the autogenous circuit product size for perturbed feed conditions. This result was attributed to the multivariable nature of the problem.

More recently, Duarte et al. (2001) have developed a multivariable neural network controller with predictive effects that was successfully applied to a model of CODELCO-ANDINA grinding plant using a non-linear dynamic simulator. Although the results obtained from the neural controller were substantially better than those obtained by applying classical control algorithms (fixed parameter controller), they were no better than those previously obtained by applying adaptive control techniques. The neural network scheme has the advantage of being much more independent of plant operating point than adaptive algorithms.

From the above investigation it is evident that the application of artificial neural networks for grinding circuit control shows great promise, although the cost of maintaining such systems are currently unclear. It must nevertheless be noted that the quality of any control strategy, based on a model of the process, depends on the quality of the model itself. The development of such a model to be used in an advanced control system also normally costs in excess of 75% of the total cost associated with the design of the control system (Hussain, 1999). It is therefore here, with the modelling of these non-linear processes, where proper development must start.

## 1.6 A literature survey on autogenous mill modelling

Comminution models are divided into two main classes (Napier-Munn et al., 1999):

- Black box models
- Fundamental models

Whereas fundamental models consider directly the interactions of ore particles and elements within the machine, black box models aim to predict the product size distribution from an ore feed size distribution and hardness, breakage characterisation and mill operating conditions (Napier-Munn et al., 1999):

$$p = F(f, b, m, s, C) \quad (1.1)$$

Here  $f$  is the feed size distribution,  $p$  is the product size distribution,  $b$  is a normalised breakage matrix,  $m$  are key mill factors,  $s$  are mill operating conditions, and  $C$  is a set of experimentally derived parameters from similar devices.

For fundamental modelling, no reference could be found to its use for the modelling of autogenous mills, mainly owing to their requirement for huge computational resources. Since black box modelling is generally more developed than fundamental modelling, references, though limited, have been found for autogenous mill modelling.

### 1.6.1 Black box modelling

One of the first references to autogenous mill modelling is that by Loveday (1978), where he used laws based on those applied for ball mill power prediction to predict the power requirements for an Aerofall type autogenous mill. Data were collected from pilot plant tests done on gold ore and carbonatite (copper ore) from Phalaborwa for the investigation of the scale-up phenomena in autogenous milling. The relationship he used for predicting mill power was:

$$\text{Mill power} \propto \rho L D^{2.5} \quad (1.2)$$

where  $L$  [m] is the length inside half-worn liners,  $D$  [m] is the diameter inside half-worn liners and  $\rho$  [g/ml] is the bulk density of the charge. This relationship assumed the same proportion of critical speed, the same dynamic angle of repose of the charge and the same proportion of volume filling between the pilot-scale and production-size mills. The correspondence of pilot-scale data and production-plant data estimates were found to be excellent, allowing the power prediction plot generated to be used in the estimation of the power draw of autogenous and semi-autogenous mills operating at about 73% of critical speed.

Using the energy of breakage applied to each particle size within the mill load as a basis, Leung et al. (1987) later developed a mill load model. This model indirectly depended on both the mill diameter and the coarse fraction of the mill load and consisted of breakage rates, discharge rates and appearance functions developed by him and incorporated into the perfect mixing matrix model of comminution (Whiten, 1974):

$$f_i - p_i + \sum_{j=i}^{i-1} r_j s_j a_{ij} - (1 - a_{ij}) r_{ij} = 0 \quad (1.3)$$

$$p_i = d_i s_i \quad (1.4)$$

where  $f_i$  is the feed rate vector,  $s_i$  is the mill contents vector,  $r_i$  is the breakage rate vector,  $d_i$  is the discharge rate vector,  $p_i$  is the product rate vector and  $a_{ij}$  is the combined appearance function. Although it was found that the model gave accurate predictions of the mill product for open circuit operation, it was highly dependant on the database used for its development. The major drawback of this type of model development is the fact that the database is constructed using data generated by expensive and time-consuming pilot scale test work.

Heiskanen (1989) took the fundamental differences between ball mills and autogenous mills, which make often used ball mill models unsuitable for autogenous milling, into account, outlining a two model approach based on the division between lumps and fines both having a model of their own. For the lumps he used:

$$\frac{dM_{Li}(t)}{dt} = -s_i M_{Li}(t) - s_i^* M_{Li}(t) + s_{i+1}^* M_{Li+1}(t) + \sum_{j=1}^{i-1} b_{ij} s_j M_{Lj}(t) \quad (1.5)$$

whereas for the fines he used:

$$\begin{aligned} \frac{dM_{Mi}(t)}{dt} = & -s_{Mi} H_{Mi} M_{Mi}(t) + \sum_{j=1}^{i-1} b_{Mij} s_{Mj} H_{Mj} M_{Mj}(t) - \frac{1}{x_{M1}} s_{Mabr} M_{Mi}(t) + \\ & \sum_{j=1}^{i-1} b_{Mij}^* s_{Mabr} \frac{1}{x_{Mj}} M_{Mj}(t) + \frac{1}{x_{Li}} s_{abr} M_{Li}(t) + s_i M_{Ln}(t) \end{aligned} \quad (1.6)$$

Here  $b_{ij}$  is the breakage function for the lumps,  $b_{Mij}$  is the breakage function for the fines,  $H_i$  is the hit probability,  $M_{Li}$  is the lump mass in fraction  $i$ ,  $M_{Mi}$  is the fines mass in fraction  $i$ ,  $s_i$  is the selection function for the lumps,  $s_{Mi}$  is the selection function for the fines,  $s_i^*$  is the selection function,  $s_{abr}$  is the abrasion rate constant for the lumps and  $s_{Mabr}$  is the abrasion rate constant for the fines. These models are coupled together so that the fines produced in the lump model are incorporated in the fines model. Although much work still needed to be done on the developed model, it was successfully used to show that the formation of critical material could be explained by taking into account the critical energies for grinding and comparing them to the energy available.

Hulbert et al. (1990) derived a linear dynamic model from results obtained from perturbation tests done on a gold ore run-of-mine milling circuit. The model characterised the relationship between the plant inputs and product size ( $PSM$  % < 75 [ $\mu\text{m}$ ]), mill load ( $LOAD$  %), slurry level in mill product sump ( $SLEV$  %), and density of the mill product cyclone feed ( $CFD$  [ $\text{t/m}^3$ ]):

$$\begin{bmatrix} g_{11}(s) & g_{12}(s) & g_{13}(s) & g_{14}(s) \\ g_{21}(s) & g_{22}(s) & g_{23}(s) & g_{24}(s) \\ g_{31}(s) & g_{32}(s) & g_{33}(s) & g_{34}(s) \\ g_{41}(s) & g_{42}(s) & g_{43}(s) & g_{44}(s) \end{bmatrix} \begin{bmatrix} \overline{\Delta SFW} \\ \overline{\Delta MFS} \\ \overline{\Delta CFF} \\ \overline{\Delta MFW} \end{bmatrix} = \begin{bmatrix} \overline{\Delta PSM} \\ \overline{\Delta LOAD} \\ \overline{\Delta SLEV} \\ \overline{\Delta CDF} \end{bmatrix} \quad (1.7)$$

where  $SFW$  [ $\text{m}^3/\text{h}$ ] is the feed rate of water to the sump,  $MFS$  [ $\text{t}/\text{h}$ ] is the feed rate of solids,  $CFF$  [ $\text{m}^3/\text{h}$ ] is the flow rate of slurry to the cyclone,  $MFW$  [ $\text{m}^3/\text{h}$ ] is the feed rate of water to the mill inlet, and each element in the matrix,  $G(s)$ , is a transfer function between an input-output pair. The model, a matrix of Laplace transforms, was used in a multivariable control scheme. However, the plant in general and, more specifically, the response of the power drawn by the mill was found to be non-linear and time-varying with slow-changing optimum operating conditions due to changes in the characteristics of the feed ore and liner and spigot wear. The dynamic modelling of the power drawn by the mill was thus not amendable to modelling by linear transfer functions.

Owing to the inherent non-linearity of autogenous mills, and the fact that the measurement of important mill variables such as feed size, ore hardness and mill load are not always possible, a conceptual dynamic model, based on Whiten's contents based model (Whiten, 1974), for autogenous and semi-autogenous grinding was developed by Valery and Morrell (1995). The parameters forming the basis of the model are the breakage rates, the appearance function, and the discharge rate. The breakage rates can be regarded as the breakage frequency and varies with factors such as mill speed and mill charge size distribution. The appearance function describes the size distribution of the products from breakage events and is a function of the ore type and specific breakage energy. The discharge rate relates the mill load to the mill product flow rate and contains both the effect of classification of solids by the grate and fluid flow effects which are related to the hold-up of slurry in the mill. When applied, the model should provide accurate dynamic response in terms of power draw, grinding charge level, slurry level and product size distribution for changes in feed rate, feed size, feed hardness and water addition.

### 1.6.2 Summary

For autogenous mills, it can be seen that only a small amount of research has been done with regard to modelling. Although black box models has matured as a modelling approach, it can also be seen that no one dynamic autogenous mill model currently exists than can be used to effectively predict the conditions surrounding an autogenous mill and successfully be incorporated into an advanced control system.

Furthermore, no reference could be found to the application of artificial neural networks for the modelling of autogenous mills or autogenous grinding circuits. This leaves the use of non-linear time series analysis and the application of artificial neural networks for the dynamic modelling of an autogenous mill unexplored.

### **1.7 Dynamic non-linear modelling**

When dealing with non-linear process systems they are often linearized, allowing the use of well-established and powerful mathematical techniques for simulation and optimisation. These strategies are often severely limited by their inability to capture the essential characteristic of the system (Kim et al., 1990). However, neural networks have been demonstrated by simulation studies on low order non-linear dynamic systems (Narendra and Parthasarathy, 1990) to effectively be used for the identification and control of non-linear dynamical systems by being able to match the underlying dynamics of the process. Since most processes in the metallurgical industry are non-linear and ill-defined by nature, neural networks offer a solution in that they are able to emulate features of processes that are poorly understood from a fundamental point of view. Neural networks therefore also offer an appealing alternative to fundamental modelling made even more attractive by their robustness with respect to noisy data and their adaptive nature.

The multi-layer perceptron neural network with sigmoidal hidden nodes and feedforward configuration is the most widely studied and applied connectionist network and has been shown to model metallurgical and chemical systems satisfactorily without any a priori knowledge about the system (Van der Walt et al., 1993). Van der Walt et al. (1993) developed a standard sigmoidal backpropagation neural network (SBNN) model to estimate the classification efficiency of a hydrocyclone classifier and showed that the network model exhibited superiority over both a SBNN with a single linear node in the hidden layer and multiple adaptive regression splines (MARS) for the relevant process. It was also concluded that given sufficient data, SBNN's could be used effectively as a modelling tool for ill-defined metallurgical systems. However, for many metallurgical processes it is both difficult and expensive to obtain sufficient modelling data of a high quality. It should also be noted that the minimum amount of information (process data points) required to

model a process adequately increases exponentially with a rise in the number of process variables. This “curse of dimensionality” (Bellman, 1961) phenomenon adds to the importance of obtaining sufficient, quality data.

More recently Du et al. (1997) used a neural network model to develop a soft sensor capable of estimating the dynamic particle size distribution of a grinding circuit product (hydrocyclone overflow). It was found that the neural net-based soft sensor performed well in the particle size estimation with the advantage of being simple to formulate and robust to the relatively long sampling horizon and high levels of noise. Neural networks therefore not only possess robust associative and computational properties but also offer both ease of adaptability and plasticity in their structure making them ideally suited for modelling the dynamic behaviour of complex processes. They would admittedly have to be retrained to accommodate periodic changes in the behaviour of the plant, but this would not be nearly as expensive as the development of explicit non-linear regression models.

## **1.8 Thesis objectives**

Based on the preceding introduction it is evident that much scope still exists for the effective modelling and control of mineral processing operations. It is hoped that some of these issues will be addressed in this thesis, its objective being the development of a non-linear, dynamic model for the UG2 fully autogenous (FAG) mill at RPM Amandelbult with the aim of using this model in the application of an advanced automatic control system. This work will also focus on the use of non-linear system identification in the form of state space reconstruction prior to dynamic modelling using actual process input-output data. Furthermore, advanced modelling techniques are used, resulting from recent progress made with regard to the identification of non-linear systems. Specific objectives of this thesis are:

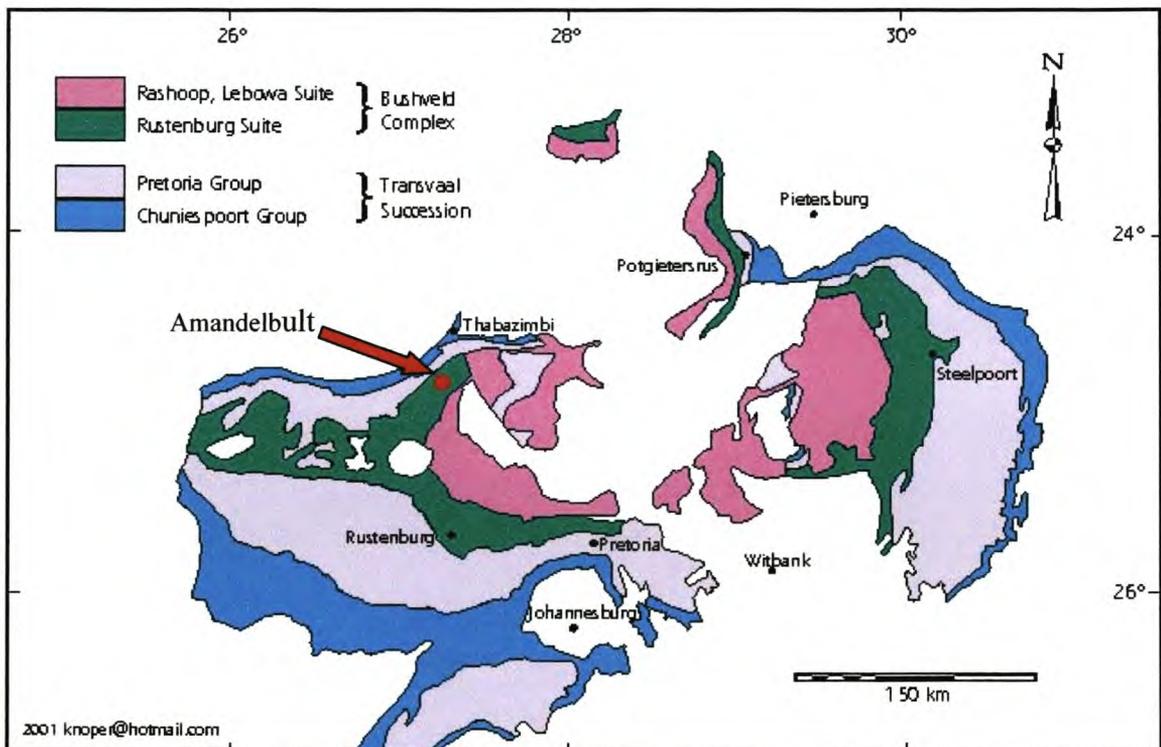
- Investigation into the RPM Amandelbult plant infrastructure, availability of instrumentation, identification of instrumentation deficiencies, variable identification, availability of data and current control operating procedures.
- Analyses of raw process data to determine its suitability for predictive modelling and ultimately automatic control of the plant using non-linear

systems identification by means of state space reconstruction. System identification is accomplished through attractor reconstruction based on Taken's theorem using both the method of delays (MOD), without any noise filtering, and singular spectrum analysis (SSA), with built-in noise filter.

- Modelling and validation of the FAG mill using three different non-linear modelling techniques: multi-layer perceptron (MLP) neural networks, radial basis function (RBF) neural networks, and least squares support vector machines (LS-SVM).
- A methodology will be presented for the development and implementation of a non-linear model in an advanced control system. This will in part be accomplished through the development of a neurocontroller using evolutionary reinforcement learning. The choice of this controller is based on its apparent ability to deliver robust and highly autonomous control to non-linear mineral processing plants.

## 2 Concentration of PGMs at Anglo Platinum's Amandelbult plant

Amandelbult is situated in the North West Province part of the Bushveld Complex, approximately 100 [km] north from Rustenburg at latitude  $24^{\circ} 49' 0S$ , longitude  $27^{\circ} 16' 60E$  and altitude 953 [m] (figure 2.1). The mine is located on a farm containing about 15.4 million tons of ore. The mining operations consist of two vertical shafts and fifteen incline shafts. In addition, four raisebored shafts are also used to assist in rock removal from the deeper levels of the incline shafts.



**Figure 2.1: Geological sketch map of the Bushveld Complex showing the location of RPM Amandelbult (Knoper, 2001)**

The platinum group metals (PGMs) and base metals mined at Amandelbult are found in the Merensky and UG2 reefs. The Merensky Reef is a thin sheet of crystalline pyroxenite that ranges in thickness from 0.3 to 0.6 meters and is bounded by two thin chromite layers. The Merensky Reef is characterised by its high PGM grades and the

high ratio of platinum to other PGMs, especially those of major importance like palladium and rhodium.

The UG2 Reef lies 15 to 370 meters below the Merensky reef, varying in thickness between 0.15 and 2.55 metres, and dips toward the centre of the Bushveld Complex at angles between 5° and 70°. It is rich in chromite but lacks the Merensky's gold, copper and nickel by-products, though its PGM reserves may be almost twice as large as those of the Merensky Reef.

The difficulty in recovering PGMs from any particular ore is determined by the ore's mineralogy; the treatment of UG2 ore having always been difficult owing to its high chromium content.

## 2.1 UG2 Reef mineralogy

The UG2 layer consists primarily of chromite (60-90% by volume), orthopyroxene (5-25%), and plagioclase (5-15%). The chromite is a solid solution of 98% spinel and 2% ulvöspinel. Accessory minerals are clinopyroxene, base metal sulphides (BMS) and other sulphides, platinum group minerals, ilmenite, magnetite, rutile, and other such as rare earth element oxides, biotite, phlogopite, secondary quartz, chlorite, serpentine, zircon, baddeleyite, barite, and calcite. McLaren and De Villiers (1982) calculated the average PGMs plus gold concentration of the UG2 layer in the northwestern parts of the Bushveld Complex to be 9.30 [ppm] (table 2.1).

**Table 2.1: Average chemical analysis of individual PGMs, Au, Cu, and Ni for the UG2 layer in the northwestern parts of the Bushveld Complex**

Element	[ppm]
Ni	1269
Cu	247
Pt	4.34
Pd	2.43
Rh	0.91
Ru	1.15
Ir	0.09
Au	0.02
Total PGM + Au	9.30

Walters (2001) found that for Amandelbult the average most abundant PGM groups present in the UG2 ore are: PGM alloys (34.20%), Ru sulphides (31.81%), Pt/Pd/Rh sulphides (26.24%), Rh phases (5.19%), PGM arsenides and sulpharenides (1.84%), and PGM tellurides (0.72%). The platinum group minerals were observed as discrete grains that are predominantly associated with BMS occurring along grain boundaries (55.98%), in silicates (30.32%), or in chromite (13.71%). From the size distribution results (table 2.2) it can be concluded that the PGM group grains are very small resulting in UG2 ores generally require finer grinding than Merensky ores for optimum PGM recovery. It was also stated that with the ore being ground fine enough to liberate all the PGMs, potential recoveries ranging from 71% to 96% could be expected.

**Table 2.2: Average size distribution of PGM grains**

Size Class [ $\mu\text{m}$ ]	%
62.5 – 44.2	3.20
44.2 – 31.2	0.61
31.2 – 22.1	6.54
22.1 – 15.6	9.45
15.6 – 11.0	11.51
11.0 – 7.8	21.78
7.8 – 5.5	13.41
5.5 – 3.9	19.64
3.9 – 2.8	7.23
2.8 – 1.0	6.64

Pentlandite and chalcopyrite are invariably present as the chief base metal sulphides with pyrrhotite or pyrite as a major or minor constituent (McLaren and De Villiers, 1982). The grains of the associated base metal sulphides are relatively small, the average size being 37.6 [ $\mu\text{m}$ ], with the maximum measuring 205.9 [ $\mu\text{m}$ ]. The modes of occurrence are the same as with the platinum group minerals, namely, along grain boundaries (81-98%), in silicates (2-19%), and in chromite (0-10%), comparing well with Hiemstra's (1982) observations.

The association of Ru sulphides with the chromite-rich layers is important as far as mode of occurrence is concerned; the Ru sulphides being commonly enclosed in the chromite crystals. It should be noted that any platinum group mineral totally enclosed in chromite is virtually non-recoverable except by sophisticated pyrometallurgical

techniques (McLaren and De Villiers, 1982). It can thus be seen that the mineralogy of the ore plays a very important role in the decisions made during the design of mineral processing plants.

## **2.2 UG2 concentrator process overview**

Platinum group metals recovery processes can be grouped into the four stages: concentration, smelting, base metals removal, and precious metals refining. Amandelbult Section treats ore from underground using concentration to produce a concentrate product that is delivered to a smelter for further processing. Prior to concentration the ore is ground to liberate mineral particles. These mineral particles are then recovered in the form of a concentrate by froth flotation. The ore mineralogy dictates both the fineness of grind required for liberation and the ideal flotation conditions.

After mining, the ore is stored in bunkers from where it is reclaimed by a conveyor belt. The ore is transported to a splitter that can either feed a jaw crusher, or bypass it, feeding a stockpile. From the jaw crusher the crushed ore is conveyed, depending on its particle size, to either the coarse ore silo or the fine ore silo. Both the coarse ore silo and the fine ore silo feed the closed circuit primary mill, a FAG mill, via the coarse and fine vibrating screens. These vibrating screens separate the natural fine material from the coarser FAG mill feed. The natural fines are fed through a bank of attritioners to the natural fines rougher bank for flotation. The natural fines rougher concentrate is combined with the primary rougher concentrate, with the natural fines rougher tails being fed to the primary rougher bank for further flotation. After milling, the FAG mill product is classified using the product vibrating screen. The oversize from this screen circulates back into the FAG mill for regrinding while the undersize feeds a sump from where it is pumped to the chrome hydrocyclone.

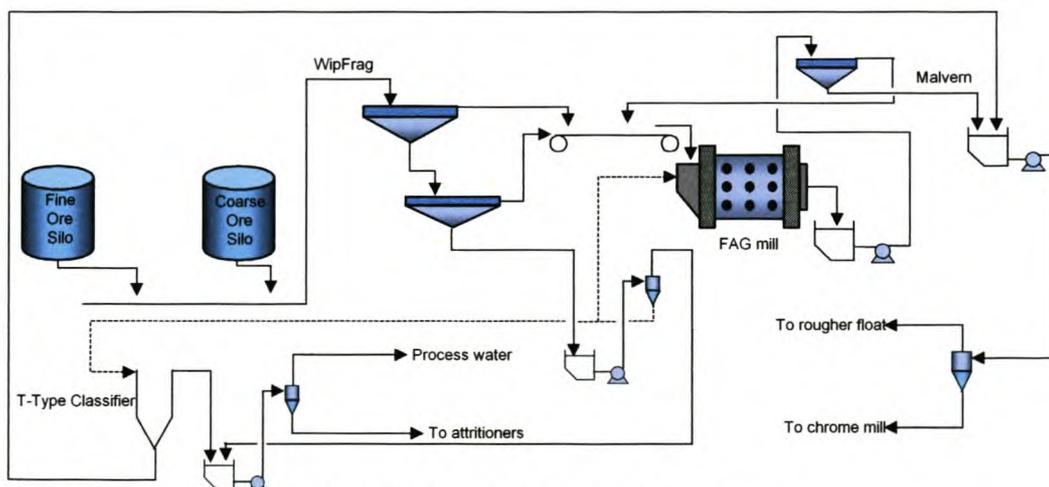
The overflow from the chrome hydrocyclone is fed to the primary rougher bank for flotation and the underflow fed to the chrome mill. The chrome mill is necessitated due to the high chrome ( $\text{Cr}_2\text{O}_3$ ) content of the UG2 ore being processed. The chrome mill is operated on an open circuit, with the mill product feeding the flash float. After flash flotation the flash float concentrate is fed to the rougher re-cleaners and the tails is combined with the final tailings.

The primary rougher bank tails contains high amounts of silica ( $\text{SiO}_2$ ) and is fed to two silica mills, operated in parallel, for further grinding. The feed to the silica mill is first classified using a hydrocyclone, the hydrocyclone overflow feeding the silica mill product sump and the hydrocyclone underflow forming the silica mill feed. The silica mill is operating in a closed circuit. The silica mill product is fed to a hydrocyclone, with the hydrocyclone underflow returning to the silica mill for regrinding and the hydrocyclone overflow feeding a Smart cell.

Owing to the intricacy of the flotation circuit, it will not be discussed in detail. It should, however, be noted that the flotation circuit consists of two primary rougher banks, a rougher cleaner bank, a rougher re-cleaner bank, three scavenger banks, a scavenger cleaner bank, and a scavenger re-cleaner bank.

### 2.3 UG2 FAG mill process variable classification and instrumentation identification

Figure 2.2 shows the Amandelbult UG2 FAG mill circuit. The flow of material through this circuit stayed the same throughout the data collection period, except for the primary hydrocyclone underflow, indicated by a dashed line. Initially the primary hydrocyclone underflow reported to the T-type classifier, but since 12 May 2001 the T-type classifier was bypassed, with the primary hydrocyclone underflow now reporting to the FAG mill feed.



**Figure 2.2: Amandelbult UG2 FAG mill section flow diagram**

For the automatic control of such a circuit, changes occurring within the circuit or in the characteristics of the product leaving the circuit must be detectable, and then compensated for by suitable variation of the controllable variables. The following four questions should be considered for the development of an automatic control system (Lynch, 1977):

1. What is the objective to be achieved?
2. What are the controllable variables?
3. What sensing instruments may be used to detect changes in circuit operation?
4. How must the controllable variables be altered to compensate for these changes?

Before answering these questions, a list of possible disturbances that commonly occurs in the UG2 milling circuit at RPM Amandelbult, and must be compensated for, is presented (some of these disturbances have already been mentioned but are included again for completeness):

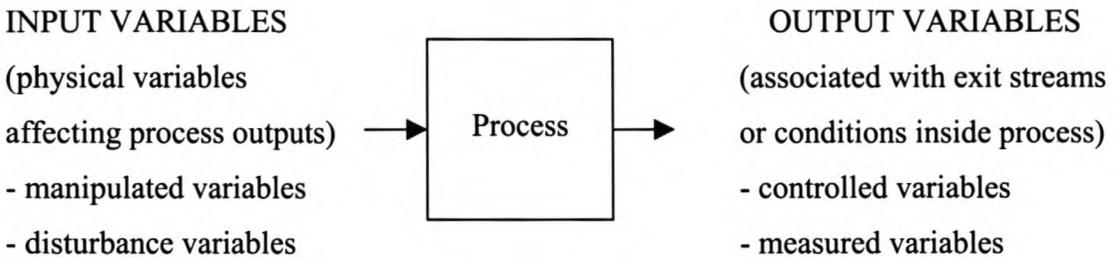
- Variations in the characteristics of the ore entering the circuit, including its hardness, size distribution, flow rate and mineral composition.
- Variation in the flow rate of water entering the circuit.
- Variations in the nature of the hydrocyclone feed pulp. Such a change can be caused by intermittent aeration of the pulp owing to a surging pump.
- Variations in the flow rates of pulp within a circuit. Such changes are usually due to mechanical reasons that include partial or complete blockage of a launder or a spigot in a hydrocyclone.
- Variations in the distribution of pulp from a splitter feeding parallel processing units. This can lead to the proportion of the total pulp entering each unit varying in an intermittent and random manner.
- A gradual change in the flow rates and size distributions of the solid particles in all streams in a closed grinding circuit.

### **2.3.1 What is the objective to be achieved?**

Depending on the data used for modelling, the objective to be reached will vary. All objectives and corresponding models investigated can be found in chapter 5.

### 2.3.2 What are the controllable variables?

The processes variables needed for control of the FAG mill at RPM Amandelbult can be classified as follows:



Output: Controlled variables:

- Product size distribution
- Power
- Discharge density

Measured variables:

- Load
- Discharge flow rate

Input: Manipulated variables:

- Inlet water flow rate
- Discharge water flow rate
- Total mill feed rate (fed to screens)
- Coarse ore feed rate
- Fine ore feed rate
- Speed

Disturbance variables:

- Actual mill feed rate
- Feed size distribution
- Ore hardness
- Varying pulp flow rates and densities
- Uneven distribution of pulp from splitter

### 2.3.3 What sensing instruments may be used to detect changes in circuit operation?

Based on the aforementioned variable classification, Napier-Munn et al. (1999) lists the minimum recommended instrumentation for an autogenous mill as:

- Feed belt weightometer – to ensure that the feed rate can be maintained at a given level with minimum surging.

- Accurate mill power measurement – to ensure that the mill is operated close to maximum power of the motor, maximising throughput while minimising specific energy.
- Charge level indicator such as load cell or bearing pressure meters – to maintain the mill load at a level below the maximum volume of charge that can be accommodated in the mill.
- Discharge slurry density gauge – to maintain a constant discharge slurry rheology.

Apart from these recommended measurements, the below listed instrumentation and significant measurements are also available for the Amandelbult UG2 FAG mill. All the of data is, however, not permanently logged and therefore some data was unavailable for use during modelling.

#### 2.3.3.1 General

The measured feed rates to the FAG mill included the total mill screen feed, the coarse ore feed rate (a calculated value), and the fine ore feed rate. Flow rates measured around the FAG mill are that of the mill discharge, the mill inlet water, and the mill discharge water. Other important measurements include mill speed and mill discharge sump level. The data collected for these aforementioned process variables are contained within the Process dataset.

#### 2.3.3.2 Malvern

The FAG mill product size distribution is measured using a Malvern Particle Size Monitor (PSM). The PSM works backwards from theories used to predict the way particles scatter and absorb light (such as the Fraunhofer model and the Mie theory) by using an optical unit to capture the actual scattering pattern from a field of particles. It then uses these theories to predict the size of particles that created the pattern.

The data collected for the Malvern is available as a size distribution (%  $-75\mu\text{m}$ ) and contained within the Malvern dataset. However, many installation, software and operational problems were experienced with the Malvern resulting in very little accurate particles size distribution data being available for modelling.

### 2.3.3.3 WipFrag

The FAG mill feed size distribution is measured using a WipFrag Photo Analysis System. The system is a PC based fragmentation analysis system that uses photo analysis techniques to measure the size distribution of broken rocks. This is accomplished through identifying individual rocks, creating an outline “2-D net”, which is subsequently measured, and then reconstructing a 3-D distribution using principles of geometric probability. The data collected for the WipFrag is available as a size distribution (% -31.5mm) and contained within the Wipfrag dataset.

It should be noted that all vision based granulometry systems, including the WipFrag Photo Analysis System, could suffer from the following sources of significant error (Maerz et al., 1996):

- Sampling errors.
- Poor delineation of ore fragments, resulting in either overestimation or underestimation of the true size of the measured fragments. This results either from groups of fragments being mistakenly grouped together and identified as a single block or from a single fragment being mistakenly divided into two or more individual blocks.
- Missing fines, resulting in a bias towards over representing the size of the distribution.

It is therefore important to take the above into consideration when analysing the data obtained from the WipFrag Photo Analysis System.

### **2.3.4 How must the controllable variables be altered to compensate for these changes?**

Although the answer to this question falls outside the scope of this thesis, some guidelines are mentioned under autogenous milling control in the appendices and the current FAG mill control and principles of operation are discussed below.

At present the main objective during the FAG mill operation is to keep the mill power drawn between 3000 and 3300 kW. For a too low FAG mill power drawn, either the FAG mill inlet water flow rate is increased or the FAG mill speed is decreased. If

both the FAG mill power and FAG mill load is low, the FAG mill speed is normally reduced. For a too high FAG mill load the FAG mill inlet water flow rate could also be increased, washing the fines from the mill, decreasing the FAG mill load. Ideally the FAG mill load should lie between 145 and 150 tons with the FAG mill speed somewhere between 67 and 70% of the mill critical speed. For conditions contrary to those mentioned, reverse action to that stated is applied.

Whereas fine ore is fed continuously to the FAG mill circuit, coarse ore is fed as needed at a rate varying between 50 and 60 t/h. The coarse ore feed rate is normally increased when the FAG mill load is high and decreased for a too low FAG mill load. The coarse ore feed rate is also varied when ore is to be saved due to the feed bins running empty. When both coarse and fine ore is being fed to the FAG mill circuit, it is done at a coarse to fine ore ratio of 1 to 4. The combined coarse ore fine ore feed rate for the process is maintained at 380 t/h.

Operating the mill at the abovementioned conditions should result in sufficient feed to the rougher banks at 35% -75 $\mu$ m.

### **2.3.5 Miscellaneous milling circuit information**

The following general information regarding mill is also deemed important (Napier-Munn et al., 1999):

- Type – fully autogenous mill
- Number of mills – one
- Diameter – 5.92 [m]
- Cylinder length – 7.5 [m]
- Liner thickness (new) – 75 [mm]
- Lifter height (new) – 150 [mm]
- Discharge mechanism - overflow
- Grate open are – 9.8%
- Grate slot width – 20 [mm]
- Trommel aperture – 0.8 [mm]
- Total charge filling (% mill volume) – approximately 35%
- Speed – 76% of critical speed

- Maximum power the mill motor can safely sustain – 3.4 [MW]
- Product screen aperture – 10 x 0.6 [mm]

When the natural fines section is running, the mill feed consists of ore particles with size +150 [ $\mu\text{m}$ ], whereas when the natural fines section is not running, the mill feed consists of a combination of coarse feed (ore particles with size +100 [mm]) and fine feed (ore particles with size –100 [mm]).

## **2.4 Summary**

As is evident from the above discussion, the performance of a mineral processing plant is subject to strong and uncontrolled variation with time, both randomly and systematically. During modelling, these unmeasured disturbances all add to the decrease in the amount of common variation between variables, resulting in a less than optimal final process model. A comprehensive understanding of the process flow, the ore being processed, the control strategy being implemented and the available instrumentation, as discussed above, will all assist in ensuring the best possible model to be developed.

### 3 System identification

---

The study of a time series can be divided into three distinct areas: *identification*, *modelling and prediction*, and *control* (Kugiumtzis et al., 1995).

*Identification* is accomplished through attractor reconstruction based on Taken's theorem (Takens, 1981). The use of both the method of delays (MOD) and singular spectrum analysis (SSA) is investigated for state space reconstruction. During this stage of analysis, the data is also tested for determinism and non-linearity using the method of surrogate data.

*Modelling* of the system is accomplished using three different non-linear modelling techniques: multi-layer perceptron (MLP) neural networks, radial basis function (RBF) neural networks, and support vector machines (SVM). Each of these techniques has its own advantages and disadvantages setting them apart from each other, as will be discussed later.

All process models should also be validated through its application to *control*. Although the development of a controller falls outside the scope of this thesis, the possible implementation of the developed models for control will be discussed in chapter 6.

#### 3.1 Identification through state space reconstruction

For deterministic dynamic systems, parametric identification methods, such as a state space model, are often regarded superior to non-parametric methods. This empirical identification of the system is done through the observation of change in one or more variables (a time series) of the dynamic system.

The time series,  $x_i = x(i\tau_s)$  for  $i = 1, \dots, N$ , where  $\tau_s$  is the sampling time, in Euclidean space  $\mathbb{R}^m$ , is often a function of all the variables associated with the process and should contain information about the dynamics of all the variables involved in the evolution of the system. By embedding the time series and its successive shifts by a

delay parameter, the dynamics of the system can be captured. Since the time series is a projection of the state variables onto the observation, the delay variables, if sufficiently independent, can be regarded as numerical estimations of a set of representative system state variables. Whereas the evolution of the state variables converge onto the dynamic state attractor of the system, the delay variables can thus form an approximate reconstruction of the state attractor. This reconstructed state allows the deterministic non-linear dynamics of the system that underlie the observed time series to be analysed and modelled. The accuracy of this reconstruction is therefore critically important to the development of a useful prediction algorithm as well as the computation of other invariant measures characterising the system's behaviour. These invariant measures include computation of the dimension, offering insight into the number of state variables needed to capture the systems dynamics, and the Lyapunov spectrum, providing a measure of the systems predictability (Nichols and Nichols, 2001).

The two most dominant methods of state space reconstruction, the method of delays and singular spectrum analysis, will be discussed below.

### 3.1.1 Method of delays

MOD is the simplest method of deriving the  $m$ -dimensional reconstructed state vector (Packard et al., 1980):

$$\mathbf{x}_i^m = [x_i, x_{i+\rho}, \dots, x_{i+(m-1)\rho}]^T \quad (3.1)$$

where  $\rho$  is a multiple integer of  $\tau_s$ , so that the delay time  $k$  is defined as  $k = \rho\tau_s$ . The  $m$  coordinates are samples (separated by a fixed  $k$ ) from a time window length  $\tau_w$ , such that

$$\tau_w = (m-1)k \quad (3.2)$$

The importance of the time window length results from the fact that in a certain sense, it determines the amount of information passed from the times series to the embedding vectors.

It has been found (Kugiumtzis and Christophersen, 1997) that MOD performs equivalent to SSA in practice for noise-free data, provided the parameters of the two methods are properly related. For both methods the quality of the reconstruction was determined by the choice of  $\tau_s$  and is independent from the selected method.

Takens' theorem (Takens, 1981) of reconstruction gives no restriction on  $k$ , while for  $m$  it states the sufficient (but not necessary) condition that  $m \geq 2d + 1$ , where  $d$  is the fractal dimension of the underlying attractor (Lillekjendlie et al., 1994). Sauer et al. (1991) have shown that by adhering to this condition, the resulting reconstructed attractor will be topologically equivalent to the true attractor. Whereas Takens' theorem is valid for the case of infinitely many noise-free data, in practice, with a limited number of possibly noisy observations, the selection of  $k$  and  $m$  is rather important for the quality of the reconstruction. Although many methods have been suggested for estimating these parameters, none of them necessarily provide appropriate estimates (Kugiumtzis, 1996).

It should also be noted that for systems with multiple outputs (components), each individual component should be analysed separately and treated as a one-dimensional time series. Since different components of the same system may need different embedding delays and embedding dimensions to represent the dynamics of the system, each component should be embedded individually. Once embedded, these individual subspaces should be concatenated column wise to form a first approximation of the attractor. Hively et al. (1999) noted that since different variables of a system contain disparate levels of dynamical information, phase space reconstruction based on one variable could be easy, but more difficult (or even impossible) when based on another.

#### 3.1.1.1 Delay time ( $k$ )

The choice of delay time is important in that too small a delay will result in vectors that are very nearly the same, each carrying a great deal of redundant information, while too large a delay may produce  $m$  coordinates that are essentially unrelated. Small errors in the data will also become exponentially magnified in time, resulting in a too large a delay and decorrelating the signal from itself (Abarbanel, 1996). The

correct delay, maximising the independence between the first two state vectors, should therefore be used as reconstruction delay time.

The two widely accepted criteria for the selection of  $k$  are (Kugiumtzis, 1996):

- The reconstructed attractor must be sufficiently expanded from the diagonal without folding back onto itself, and
- The components of the vector  $x_i$  must be uncorrelated.

For both these criteria, increasing  $k$  expands the attractor from the diagonal (components get less correlated) until at some point at which folding may occur (components again get correlated). For criteria one, the goal of stretching the attractor to make it cover the largest region or expand it maximally from the diagonal to get good reconstruction is based on empirical rather than theoretical grounds. In theory, however, a good reconstruction means near topological equivalence of the reconstructed attractor, which can be viewed as an effort to recreate the full  $n$ -dimensional state space using the time series of a single variable, to the original one. The question therefore arose why one should seek the delay time that gives sufficient expansion from the diagonal? Kugiumtzis (1996) concluded that expansion per se does not guarantee a configuration of the reconstructed attractor closer to the original one.

For criteria two, the estimates of  $k$  are based either on linear decorrelation, using the linear autocorrelation (LA) function, or general decorrelation, using average mutual information (AMI). Although other methods are also available for delay time estimation, these methods are most commonly used, with AMI having been suggested to be the most robust in the presence of noisy data (Abarbanel, 1996).

In essence, the autocorrelation function determines to what extent one part of the time series looks like another part of the same time series, measuring the degree of correlation of a variable at one time with itself at another time. The calculation of autocorrelation is based on both the autocovariance and variance of the data.

Autocovariance is used to determine how the time series varies with itself by comparing the time series with itself in time:

$$\text{autocovariance} = \frac{1}{N} \sum_{t=1}^{N-k} (x_t - \bar{x})(x_{t+k} - \bar{x}) \quad (3.3)$$

When calculating the autocovariance it should be done for a time series containing at least 50 observations and the largest delay to be used should not be bigger than  $\frac{1}{4}$  of the number of observations in the time series (Williams, 1999). Since the autocovariance reflects the units of the physical features measured, it can be difficult to interpret. It is therefore standardised, making it dimensionless and easy to directly compare to other standardised autocovariances, using the variance:

$$\text{variance} = \frac{1}{N} \sum_{t=1}^N (x_t - \bar{x})^2 \quad (3.4)$$

By computing the autocorrelation for all time delays from zero:

$$\begin{aligned} \text{autocorrelation} &= \frac{\text{autocovariance}}{\text{variance}} \\ &= \frac{\sum_{t=1}^{N-k} (x_t - \bar{x})(x_{t+k} - \bar{x})}{\sum_{t=1}^N (x_t - \bar{x})^2} \end{aligned} \quad (3.5)$$

the function will reveal if the time series has built-in self-similarity and by how much it should be shifted (time delay) to have maximum autocovariance. For random data the autocorrelations should be near zero for any and all time-lag separations indicating complete uncorrelated data with a high degree of noise, whereas for non-random data one or more of the autocorrelations will be significantly non-zero.

Similar to the autocorrelation function, mutual information tries to measure the extent to which values of  $x_{t+k}$  are related to values of  $x_t$ , at a given lag, but with the

advantage of using probabilities rather than a linear basis to assess the correlation (Nichols and Nichols, 2001):

$$AMI = \sum_{x_i, x_{i+k}} P(x_i, x_{i+k}) \log_2 \left[ \frac{P(x_i, x_{i+k})}{P(x_i)P(x_{i+k})} \right] \quad (3.6)$$

where  $P(x_i)$  and  $P(x_{i+k})$  is the marginal and  $P(x_i, x_{i+k})$  the joint probability density for observations  $x_i$  and  $x_{i+k}$ . Rewriting equation 3.6 using self and joint entropies we find:

$$AMI = \sum_i px_i \log_2(px_i) + \sum_j py_j \log_2(py_j) - \sum_i \sum_j px_i y_j \log_2(px_i y_j) \quad (3.7)$$

By plotting  $x_{i+k}$  against  $x_k$ , the entropies can be graphically determined by placing a mesh over the phase-space, designed such that the number of boxes is approximately equal to the number of observations in the time series,  $N$ , and counting the observations. For self entropies, all the observations in a row or column is taken and divided by  $N$ , whereas for the joint entropies, only the observations in the box referred to by the joint entropy is counted and divided by  $N$ .

When coordinates of a point have the same value, they represent the same information (maximum mutual information) and one can be used to exactly predict the other. For attractor reconstruction  $x_t$  should not provide a lot of information about  $x_{t+k}$  (minimum mutual information). Therefore, for strongly related values at a given lag the mutual information will be relatively high, whereas for weakly related values at a particular lag, the mutual information will be relatively low (Williams, 1999). The first minimum of the Average Mutual Information plot for a time series can thus be used to tell at which point the values become independent from one another. Having a distribution of mutual information which is sharp, and not only a low AMI value, is also important in that a single choice of  $k$  over the whole attractor can accurately capture the underlying dynamical process (Abarbanel et al., 2001).

Since the autocorrelation function expresses a linear relationship between the signal and itself at a given delay, the use of average mutual information for delay time estimation is favoured when analysing non-linear systems. Although the use of the first minimum of the AMI statistic for delay time estimation has not been made mathematically rigorous, it is not tied to linear/non-linear properties of the system (Abarbanel, 1996), and represents a clear improvement over the autocorrelation function (Nichols and Nichols, 2001). The disadvantage of using the AMI statistic for delay time estimation is that it requires very large datasets, rendering many data points unusable (Atmanspacher et al., 1988).

Although both these methods guarantee decorrelation between two successive components  $x_i$  and  $x_{i+k}$  of the reconstructed vector  $\mathbf{x}_i$ , even if  $x_i$  and  $x_{i+k}$  are uncorrelated and  $x_{i+k}$  and  $x_{i+2k}$  are uncorrelated, it does not follow that  $x_i$  and  $x_{i+2k}$  are also uncorrelated (Kugiumtzis, 1996).

These problems, combined with the fact that the autocorrelation function may get approximately zero only after an extremely long time or the AMI may not have a clear minimum, indicate that the analysis of non-linear systems is not algorithmic alone, but also requires some experience and intuition.

### 3.1.1.2 Embedding dimension ( $m$ )

The embedding dimension  $m$  is normally determined by successively embedding the attractor into higher dimensional spaces until some criterion of the attractor geometry is met. As stated earlier, Takens (1981) showed that theoretically the condition  $m \geq 2d + 1$  should be satisfied in order to preserve the dynamical properties of the original attractor. When estimating  $m$  using MOD,  $k$  must be fixed.

False nearest neighbours (FNN), developed by Kennel et al. (1992), is probably the most popular of the different geometrical criteria available for estimating the embedding dimension. FNN is concerned with the fundamental condition of no self-intersections of the reconstructed attractor by attempting to eliminate the false projections that occur when a system is embedded in too few dimensions. False

nearest neighbours are usually near each other for non-dynamical reasons, e.g. owing to projection, and are connected with unpredictability.

The condition of no self-intersections states that if the attractor is to be reconstructed successfully in  $\mathbb{R}^m$ , then all neighbouring points in  $\mathbb{R}^m$  should also be neighbours in  $\mathbb{R}^{m+1}$  (Kugiumtzis, 1996). Whereas the original attractor lies on a smooth manifold of dimension  $d$ , self-intersections of the reconstructed attractors indicate that it does not lie on a smooth manifold resulting in unsuccessful reconstruction. The lowest embedding dimension that gives reconstructions without self-intersections is estimated by checking the neighbours of successively higher embedding dimensions until it finds only a negligible number of false neighbours when increasing the dimension from  $m$  to  $m+1$ ; a neighbour being classified in terms of the Euclidean distance being within a preset limit. This does not necessarily mean that these true neighbours on the reconstructed attractor are true neighbours on the original attractor (Kugiumtzis, 1996). The advantage of using an empirical approach over theoretical criteria is that no a priori knowledge of the system's dimensionality is required while also serving to eliminate the redundant information carried in extra delay vectors (Nichols and Nichols, 2001).

When considering the above it must be noted that it has been shown for certain instances that the proportion of false nearest neighbours does not fall to zero for the same  $m$  as  $k$  increases, but rather the estimated  $m$  increases slowly with  $k$ , making the estimation of  $m$  somewhat arbitrary unless the same  $m$  is found for a sufficiently large range of  $k$  values (Kugiumtzis, 1996). The use of FNN for embedding dimension estimation is also not robust to the presence of noise in the data (Rhodes and Morari, 1997), giving larger values of  $m$  for noisy data (Aleksić, 1991), being a serious drawback to its application to real data were one is led to choose a larger  $m$  than really needed, possibly leading to over-fitting. Too large an embedding dimension could also lead to a higher computational burden than necessary, and probably a less accurate predictor. This problem is somewhat overcome when using SSA with its built-in noise filter.

An alternative to the FNN technique for embedding dimension estimation is the use of the false nearest strands (FNS) technique (Kennel and Abarbanel, 1995). In particular, the goal of FNS is to give a sharp indication of the minimum embedding dimension by providing corrections to account for:

- Oversampled continuous data.
- Autocorrelation owing to small time delays.
- Sparsely populated regions of the attractor.

Whereas the FNN algorithm counts individual observations to determine nearest neighbours, the FNS algorithm counts nearest strands. Given pairs of real nearest neighbour points, nearest strand pairs are formed by collecting a whole strand of trajectory nearby to the trajectory that is being examined. A strand pair is characterised by its points being neighbours to  $x_i$  but not being on the same trajectory segment within a decorrelation window around the time point  $i$ . The number of false nearest strands is then compared to the total number of strands.

As before, the need for experience and intuition for the analysis of non-linear systems is apparent.

### 3.1.2 Singular spectrum analysis

What sets SSA apart from MOD is the fact that in MOD the  $m$  coordinates are samples separated by a fixed  $\tau$  and cover a time window length  $\tau_w$ , while in SSA all the available samples in  $\tau_w$  are initially used before being further processed with singular value decomposition (SVD) so that the final  $m$  coordinates are linear combinations of these measurements (Kugiumtzis and Christophersen, 1997). SSA can therefore be viewed as a linear root mean square (RMS) fitting method.

In general, SSA is the application of principal component analysis to the trajectory matrix obtained from the original time series with a subsequent reconstruction of the time series. A large  $p$ -dimensional state vector is first derived from successive

samples, by embedding with a delay of unity and an embedding dimension  $p \gg m$ , forming a trajectory matrix:

$$\mathbf{x}_i^p = [x_i, x_{i+1}, \dots, x_{i+p-1}]^T \quad (3.8)$$

The data is decomposed using SVD to yield a set of singular vectors and associated singular values (see chapter 6). The singular vectors are chosen to accommodate the most populated directions while the singular values relate the degree to which the data fill the new directions. The final  $m$ -dimensional state vector  $\mathbf{x}_i^m$  is therefore a projection onto the first  $m$  principal components (the most populated directions) defined by the data in  $\mathbb{R}^p$ , i.e.  $\mathbf{x}_i^m = P\mathbf{x}_i^p$ , where  $P$  is a  $m \times p$  matrix (Broomhead and King, 1986). Furthermore, the  $p - m$  remaining vectors are discarded, thus removing system noise. For SSA we have:

$$\tau_w = (p - 1)\tau_s \quad (3.9)$$

With SSA the choice of embedding dimension is not critical and any choice over a lower limit will give essentially the same reconstruction because the additional coordinates correspond to less significant singular values and give negligible variance assuming  $\tau_w$  is sufficiently large (Kugiumtzis and Christophersen, 1997).

Although one has to decide subjectively the number of eigenvalues to retain, using SSA for attractor reconstruction removes some of the guesswork associated with MOD by fixing the delay time and removing additive noise from the data. Sometimes SSA fails to distinguish a chaotic signal from noise with nearly the same Fourier spectrum (Fraser, 1989). However, for noisy data it was found that the in-built filter in the SSA reconstruction made SSA superior to MOD (Kugiumtzis and Christophersen, 1997), establishing the applicability of SSA to “real” data.

### 3.1.3 Quality of state space reconstruction

From the aforementioned discussion it is evident that the estimation of  $k$  and  $m$  does not always guarantee good reconstructions and that the performance of the estimation

methods depends on the problem at hand. Properties of the attractor that could be used to determine the quality of the reconstruction or more thoroughly study the dynamics of the system include the correlation dimension and Lyapunov exponents. These attractor properties are independent of any particular trajectory and are called invariants of the system. In reality, with a finite number of noisy data, they were found to be dependent on both  $m$  and  $k$  (Kugiumtzis et al., 1995).

Whereas the correlation dimension gives an estimate of the systems complexity, the Lyapunov exponent estimates the amount of chaos in a system.

### 3.1.3.1 Correlation dimension

The correlation dimension,  $d_c$ , was first defined by Grassberger and Procaccia (1983). This is the most popular measure of an attractor's dimension, being defined as:

$$d_c = \lim_{\varepsilon \rightarrow 0} \lim_{N \rightarrow \infty} \frac{\log C_N}{\log \varepsilon} \quad (3.10)$$

where  $C_N$  is the correlation function, defined by:

$$C_N(\varepsilon) = \binom{N}{2}^{-1} \sum_{0 \leq i < j \leq N} I(\|v_i - v_j\| < \varepsilon) \quad (3.11)$$

$N$  is the number of observations in the time series and  $I(\cdot)$  a Heaviside function that returns one if the distance between point  $i$  and  $j$  is within  $\varepsilon$ , otherwise, zero. Here  $\varepsilon$  defines the correlation length. Kugiumtzis (1996) observed that the estimate of  $d_c$  for a chaotic time series (clean or noisy) is approximately the same under variations of the parameters  $k$  and  $m$  while keeping  $\tau_w = (m-1)k$  fixed (assuming that  $m$  is always larger than the dimension of the attractor). To assure a good estimation the same  $d_c$  value should therefore be found for different reconstructions with systematically varying parameters; the curves of  $C_N(\varepsilon)$  vs.  $\varepsilon$  on a double logarithmic scale should be approximately straight and parallel for consecutive values of  $m$

indicating a  $\varepsilon$ -interval where the scaling law holds. Nichols and Nichols (2001) found that due to the fact that correlation dimensions focus strictly on the distribution of points with length scale and not temporal evolution, inaccuracies in the reconstruction have little, if any, effect for a reasonably chosen delay.

The calculation of the correlation dimension using the Grassberger-Procaccia method is robust to low-amplitude noise, making it a powerful tool for distinguishing chaotic time series from white noise (Kugiumtzis, 1995). However, since the separation of correlated noise from chaos is more complicated, dimension estimates from real data must be considered with caution.

Judd (1992) indicated that the Grassberger-Procaccia method of dimension estimation had certain deficiencies. Probably the most important deficiency with regard to its application to high dimensional data is the fact that linear correlation in a time series can mislead the Grassberger-Procaccia method into wrongly showing convergence to some low dimension, which in turn could be misinterpreted for inherently low-dimensional dynamics. This led Judd (1992) to propose a different algorithm for estimating correlation dimension. Whereas the Grassberger-Procaccia method required a linear scaling region, the algorithm proposed by Judd replaces this requirement by fitting a polynomial of the order of the topological dimension in that region. The correlation dimension could therefore be estimated as a function of scale  $\varepsilon_0$  using:

$$C_N(\varepsilon) \approx \varepsilon^{d_c} q(\varepsilon) \quad (3.12)$$

valid for  $\varepsilon < \varepsilon_0$ , where  $q(\cdot)$  is a polynomial of the order of the topological dimension (Small and Judd, 1998a). The Judd algorithm does have a failing in that it can only express the correlation dimension for inter-point distances below a specific scale  $\varepsilon_0$ . However, the Judd algorithm is less sensitive to the number of observations by and order of magnitude when compared to the Grassberger-Procaccia method. A time series of approximately 1000 observations is usually sufficient for the Judd algorithm, generally resulting in better correlation dimension estimates. Also, instead of comparing correlation dimension estimates, one rather compares the clustering of the

correlation dimension curves, allowing the use of the correlation dimension for the examination of the macro- and microscale of the reconstructed dynamic attractor.

### 3.1.3.2 Lyapunov exponents

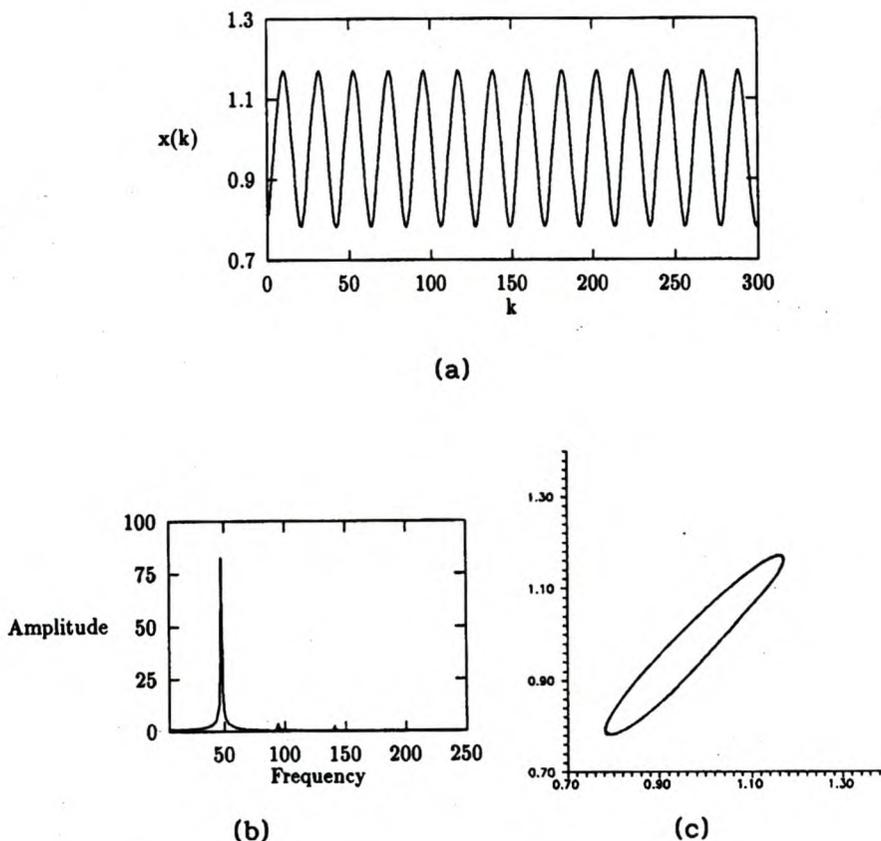
Attractor dynamics can be characterised by stretching, the divergence of nearby trajectories, and folding, constraining the dynamics to a finite region in a subspace. Lyapunov exponents, defined as the logarithms of the absolute value of the eigenvalues of the linearised dynamics averaged over the attractor (Kugiumtzis et al., 1995), can be used to measure the average local rates of linear expansion and contraction of infinitesimal perturbations to a trajectory in each of the system's principal directions. A negative Lyapunov exponent indicates a local average rate of attractor contraction (stable behaviour), whereas a positive Lyapunov exponent indicates a local average rate of attractor expansion (unstable, chaotic behaviour). For chaotic systems, both positive and negative Lyapunov exponents can coexist. The exponents then indicate the complexity of the system dynamics with the largest positive value determining the upper prediction limit. Even though a single positive Lyapunov exponent is sufficient for diagnosing chaos, the system can still be globally stable if the sum of the entire Lyapunov spectrum is negative. The sum of only the positive Lyapunov exponents in the spectrum equals the Kolmogorov entropy or mean rate of information gain (Eckmann and Ruelle, 1985).

The method used to calculate the Lyapunov exponents for this research is based on the work of Sano and Sawada (1985), employing local linear fits to estimate the required linearised flow map of tangent space from the observed data. It should be noted that although more data is required to estimate Lyapunov exponents than correlation dimensions, a biased estimate of the largest Lyapunov exponent gives stronger evidence for chaos, unless it is close to zero, than a biased correlation dimension estimate. Because the algorithm cannot internally check for the consistency of the assumption of an underlying dynamic system, it should not be blindly applied to data from a random process resulting in a completely meaningless Lyapunov spectrum.

### 3.1.3.3 Attractor shape

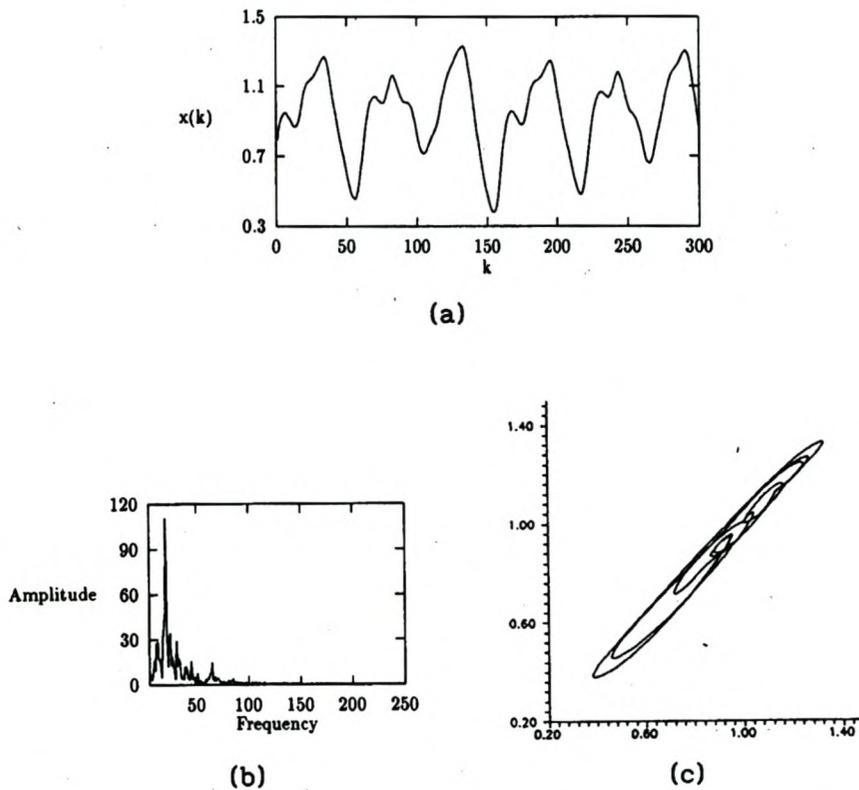
According to Kanjilal (1995), data series or signals with certain degrees of periodicity can broadly be classified as periodic, quasiperiodic, or chaotic. A periodic series shows a repeating pattern, the repetition frequency being the same as the periodicity, while a quasiperiodic series is only a linear combination of two or more periodic series each of whose frequencies is a linear combination of a finite set of frequencies, at least two of which are unequal. In contrast to this, distinguishing features of a chaotic series is a continuous frequency spectrum and sensitivity to initial conditions.

For a periodic process (figure 3.1) the state space diagram will show at least one closed contour per period. If there is any further explicit periodicity within the periods, the state space diagram will show a smaller closed contour corresponding to each of such periods within the main period.



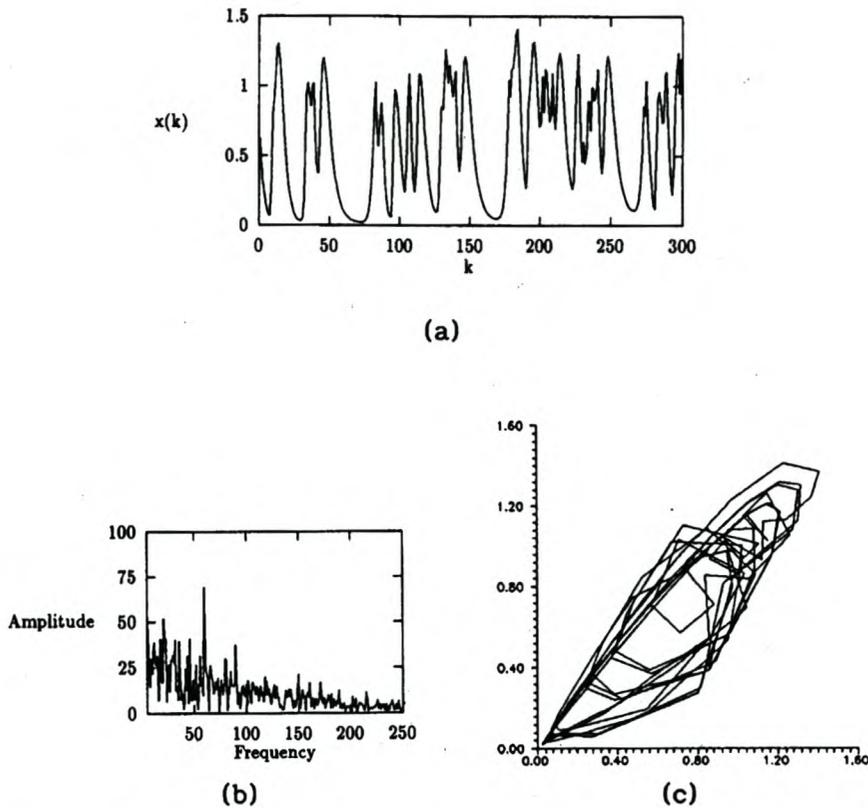
**Figure 3.1: (a) The nearly periodic Mackey-Glass series, (b) the frequency spectrum showing primarily one frequency component, (c) the attractor (Kanjilal, 1995)**

For a quasiperiodic process (figure 3.2) generated from a linear combination of  $n$  periodic processes, there will be at least  $q$  number of closed contours, where  $1 \leq q \leq n$ , if harmonic relations exist between the component processes. Should non-harmonic components exist, the composite process will show at least  $n$  closed contours.



**Figure 3.2: (a) The quasiperiodic Mackey-Glass series, (b) the frequency spectrum showing primarily two bands of frequencies, (c) the attractor (Kanjilal, 1995)**

For a chaotic process (figure 3.3) the state space diagram will show a large number of closed contours with no apparent order of arrangement. These contours will be nonintersecting in three dimensional or larger dimensional spaces.



**Figure 3.3: (a) The chaotic Mackey-Glass series, (b) the frequency spectrum showing relatively widely spread frequency components, (c) the attractor (Kanjilal, 1995)**

### 3.2 Surrogate data

Surrogate data are non-deterministic, artificially generated data that mimic certain features of a measured time series but are otherwise stochastic. Surrogate data is used to test for non-linearity and determinism in a time series by determining whether the observed time series, similar in many respects to the specially designed non-deterministic data, is likely to have essentially the same value of some chosen non-linear measure.

The general steps followed when applying the method of surrogate data is (Williams, 1999):

1. Generate many surrogate times series, similar to the original time series, using an assumed process to which the original time series was assumed to belong.

The surrogate time series should be of equal length to that of the original time series and statistically indistinguishable with regard to certain specified characteristics.

2. Compute a discriminating statistic for the original and each of the surrogate time series. For non-linear dynamics the discriminating statistic used is usually the correlation dimension or the largest Lyapunov exponent.
3. Hypothesize that there is no difference between the discriminating statistic of the original time series and that of each of the surrogate time series (the null hypothesis).
4. Determine whether or not the discriminating statistic computed for the original time series differs significantly from that calculated for each of the surrogate time series. If it does, reject the null hypothesis that the process that has generated the original data is of the same class as the system that has generated the surrogate data, concluding that there is a discernible difference between the original and each of the surrogate time series.

Mathematically, let  $\psi$  be a specific hypothesis,  $\mathfrak{S}_\psi$  the set of all processes (or systems) consistent with that hypothesis,  $x \in \mathbb{R}^N$  be a time series consisting of  $N$  scalar measurements, and  $T: \mathbb{R}^N \rightarrow U$  be a statistic which will be used to test the hypothesis  $\psi$  that  $x$  was generated by some process  $\mathfrak{S} \in \mathfrak{S}_\psi$ . Generally  $U$  will be  $\mathbb{R}$  and it will be possible to discriminate between the original data  $x$  and the surrogate data  $x_s$  consistent with the hypothesis given the probability density of  $T$ , given  $\mathfrak{S}$ , i.e.  $p_{T,\mathfrak{S}}(t)$  (Small and Judd, 1998a).

Three main algorithms exist to generate surrogate data (Popivanov and Mineva, 1999):

- Type 0 – shuffling the data, preserving the probability distribution but destroying any linear correlations
- Type 1 – randomising the phases of the data using Fourier transforms, preserving the linear correlation (i.e. power spectrum) but destroying the non-linear structures

- Type 2 – applying a phase randomising procedure to the amplitude adjusted Gaussian noise, preserving the linear correlation and possible static non-linear transformation, but destroying the non-linear dynamics.

The surrogate data generated for this research was done using the algorithm for type 2 surrogate data. The important properties of the surrogate data generated using this algorithm include: dynamic non-linearities, a Gaussian distribution, and generation using a stationary process (Theiler et al., 1992). Although Barnard (1999) has shown that type 2 surrogate data is beneficial for screening data prior to model building, it can still not be proven that there is no dynamics in the process beyond what is covered by the null hypothesis (Schreiber and Schmitz, 2000).

### **3.3 Modelling**

Once the state space is successfully reconstructed it can be used to provide effective models of the dynamics of the system, allowing one to predict the evolution of any new point in the state space within the basin of observed attraction. Since the data used for modelling are observed on the attractor alone, models that would have general validity throughout the system state space cannot be built. For instance, if a set of initial conditions exist in the original phase space that leads to other behaviour than what was observed and analysed, it is possible that the developed model for the motion in that other basin of attraction could be different. The model would therefore be useful over the whole attractor but is unlikely to be of any value off the attractor.

Neural networks offer some of the most versatile ways of modelling non-linear processes of a diverse nature. Owing to the fact that the optimal approximation errors for MLP neural networks and RBF neural networks grow more slowly with dimension than for weak functional approximations, Judd and Mees (1995) classified these neural networks as strong functional approximations. Strong functional approximations not only comply with the requirement for higher dimensionality in models, but their fitting processes are also inherently non-linear, making them ideal for application to the dynamic system under investigation. For these reasons, least squares support vector machines (LS-SVM), a new class of neural networks, can also be classified as strong functional approximations.

### 3.3.1 Multi-layer perceptron neural networks

MLP neural networks are probably the most commonly used neural networks for non-linear function approximation. Of this class of artificial neural networks, the feed-forward backpropagation neural network, as implemented in the Matlab Neural Network Toolbox, was employed extensively for MLP neural network model development.

The development of a MLP neural network consists of first determining the neural network model structure (number of layers, number of nodes per layer, types of nodes, etc.) followed by the training of the neural network model to determine its parameters (weights).

#### 3.3.1.1 Model structure

A neural network consists of various interconnected neurons or nodes, each having a processing element with a set of input connections as well as a single output connection. These connections are all characterised by a numerical value or weight that is an indication of the strength of the connection. The output of a node can be expressed as

$$z = f(\mathbf{w}^T \mathbf{x}) \quad (3.13)$$

The function  $f(\mathbf{w}^T \mathbf{x})$  is the activation or transfer function of the node, where  $\mathbf{w}$  is the weight vector of the neural node, defined as

$$\mathbf{w} = [w_1, w_2, w_3, \dots, w_N]^T \quad (3.14)$$

and  $\mathbf{x}$  the input vector, defined as

$$\mathbf{x} = [x_1, x_2, x_3, \dots, x_N]^T \quad (3.15)$$

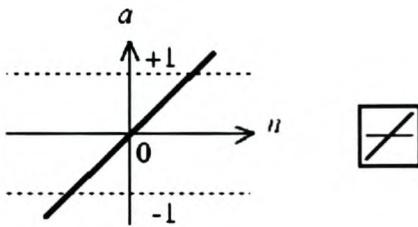
Although many transfer functions exist, the most commonly used are the linear transfer function (used in the output layer):

$$z = x \quad (3.16)$$

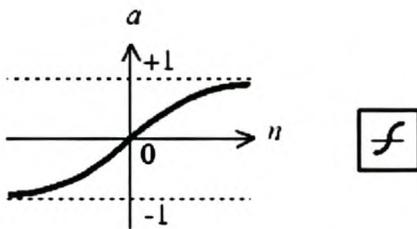
and the tan-sigmoid transfer function (used in the hidden layer):

$$z = \frac{2}{(1 + \exp(-2x))} - 1 \quad (3.17)$$

Whereas the linear transfer function generates an output between  $-\infty$  and  $+\infty$  (figure 3.4), the tan-sigmoid transfer function generates an output between  $-1$  and  $+1$  (figure 3.5).



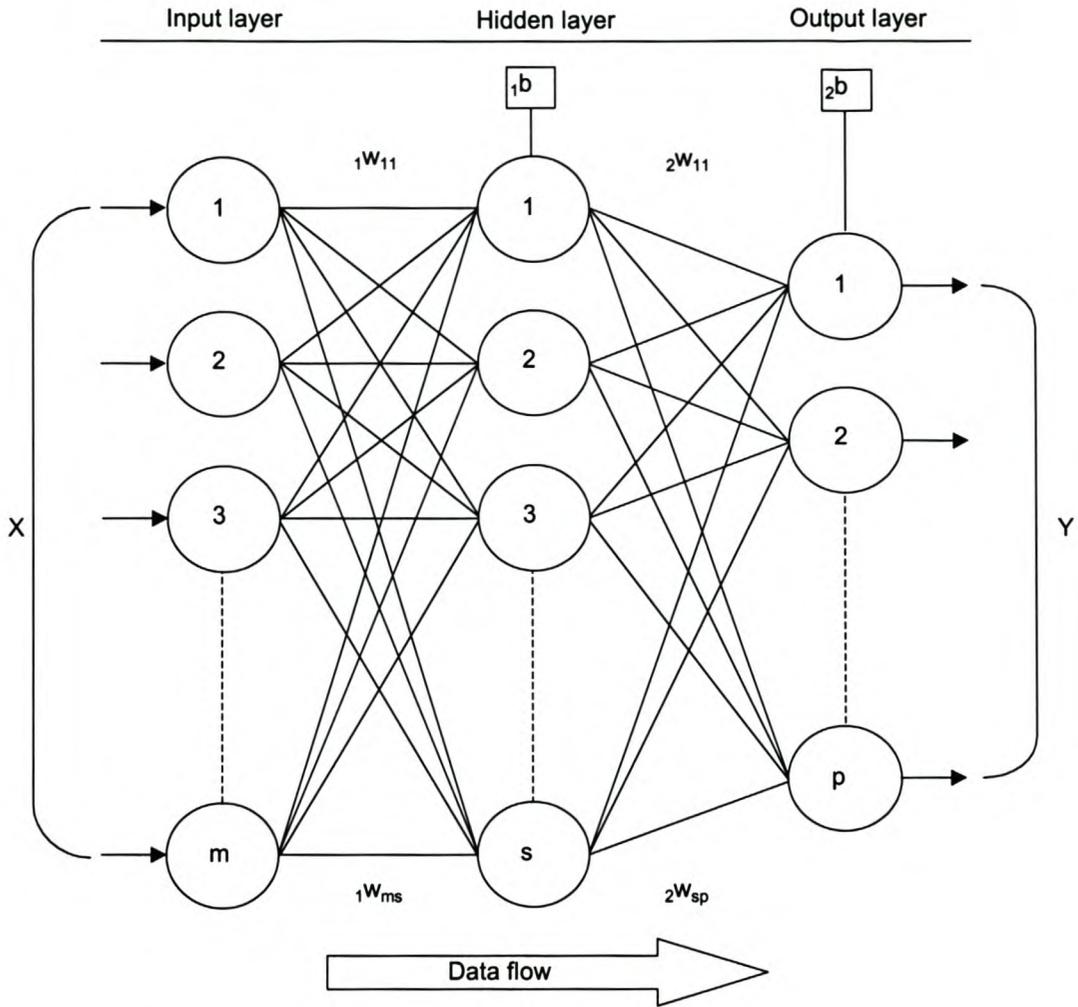
**Figure 3.4: Typical linear transfer function (Demuth and Beale, 2001)**



**Figure 3.5: Typical tan-sigmoid transfer function (Demuth and Beale, 2001)**

For some neurons an additional input, the bias, can also be defined. By offsetting the output of the neural network, the bias input value sometimes enables the network to form a more accurate representation of process trends.

All these nodes are divided into layers, each layer comprising of nodes having similar computational characteristics. The layers are divided into input, hidden and output layers with the nodes in one layer being connected to other nodes in successive layers by means of the aforementioned weighted connections. Figure 3.6 illustrates the structure of a typical MLP neural network.



**Figure 3.6: Structure of a typical MLP neural network**

For any neural network the number of nodes in the input and output layers of the network is fixed by the number of network inputs and outputs, with the number of nodes in the hidden layer of the network being determined by some predefined criteria. One such criterion used to determine the number of nodes required in the hidden layer of the network is the Schwartz Information Criterion (Schwartz, 1978):

$$SIC = N \sum_{i=1}^p \log(MSE_i) + d_M \log(N) \quad (3.18)$$

Here the summation spans the set of mean square error of each component of the multi-dimensional time series and  $d_M$  is the model order, defined as:

$$d_M = S(b + f + 1) + f \quad (3.19)$$

with  $S$  being the number of nodes specified for the hidden layer of the network,  $f$  the number of model outputs and  $b$  the number of model inputs.  $MSE$  is the mean square error of prediction of each of the model outputs, defined as:

$$MSE = \frac{1}{N} \sum_{i=1}^N y_{res,i}^2 \quad (3.20)$$

The SIC method presents a formalised structure to determine the model order, compared to other rather ad hoc approaches available.

### 3.3.1.2 Model training

Multi-layer perceptron neural networks are trained by repeatedly attempting to match sets of input data to corresponding sets of output data or target values (supervised learning). The network weights and biases constitute the model parameters and are updated by propagating the training information back through the net during the learning process. This backpropagation is used to determine the gradient of the performance function, the MSE, which in turn is then used to determine how to adjust the network weights to minimise the performance. This training algorithm is classified as a gradient descent algorithm and, more specifically, the batch gradient descent algorithm with momentum function was used as the weight/bias learning function during the training of all MLP neural networks. The addition of the momentum function (Plaut et al., 1986) to the gradient descent algorithm not only results in way of dealing with the problem of widely differing eigenvalues but also effectively adds inertia to the motion through weight space and smoothes out possible oscillations (Bishop, 1999).

The algorithm works by first guessing some initial network weights and biases and then moving these network weights and biases in the direction of the negative gradient of the performance function, while momentum allows the network to respond not only to local gradient, but also to recent trends in the error surface. Because momentum acts like a low pass filter, small features in the error surface are ignored, allowing the

network to slide through any shallow local minimums in which it would normally get stuck. Momentum is added to the backpropagation learning function by making weight changes equal to the sum of a fraction of the last weight change and the suggested new changes and generally leads to a significant improvement in the performance of the gradient descent algorithm (Bishop, 1999).

The training function used during the training of all MLP neural networks was the Levenberg-Marquardt algorithm (Levenberg, 1944; Marquardt, 1963). The Levenberg-Marquardt algorithm, a quasi-Newton method, was designed to approach second-order training speed without having to compute the Hessian matrix (matrix of second-order partial derivatives) for non-linear networks, but approximating it as:

$$\mathbf{H} = \mathbf{J}^T \mathbf{J} \quad (3.21)$$

with the gradient begin computed as:

$$\mathbf{g} = \mathbf{J}^T \mathbf{e} \quad (3.22)$$

where  $\mathbf{J}$  is the Jacobian matrix, containing the first-order derivatives of the network errors with respect to the weights and biases, and  $\mathbf{e}$  the network errors vector. This approximation to the Hessian matrix (its inverse approximating the variance/covariance matrix of parameter estimates) is then used to update the network weights at the end of a training iteration as follows:

$$\mathbf{w}_{k+1} = \mathbf{w}_k - [\mathbf{H} + \mu \mathbf{I}]^{-1} \mathbf{g} \quad (3.23)$$

For  $\mu$  equal to zero, equation 3.18 reduces to Newton's method, whereas for a large  $\mu$  it becomes gradient descent with a small step size. Since Newton's method is faster and more accurate near an error minimum,  $\mu$  is reduced after each successful step, reducing the performance function, and is only increased when a tentative step would increase the performance function. The Levenberg-Marquardt algorithm always seeks to minimise the error function while at the same time trying to keep the

step size as small as possible as to ensure that the linear approximation, on which the method is based, remains valid.

Although the Levenberg-Marquardt algorithm appears to be the fastest method for training of moderate-sized feed-forward backpropagation neural networks (Hagan and Menhaj, 1994), it has a drawback in that it requires the storage of some matrices which can be quite large for certain problems.

### 3.3.2 Radial basis function neural networks

Whereas MLP neural networks are based on units that compute a non-linear function of the scalar product of the input vector and a weight vector, for RBF neural networks the distance between the input vector and a prototype vector determines the activation of a hidden unit. The Matlab functions used for RBF neural network model development were those developed by Orr (1999b) for nonparametric regression. Due to performance considerations (Orr, 1999c) the method employing forward subset selection, including ridge regression with a single regularisation parameter, is focused upon.

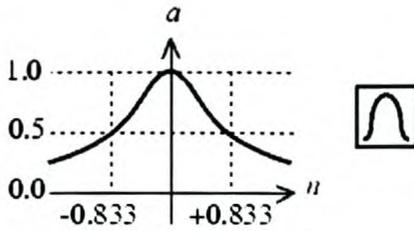
As with the development of a MLP neural network, the development of a RBF neural networks also consists of first determining the neural network model structure followed by the training of the neural network.

#### 3.3.2.1 Model structure

Similar to MLP neural networks, RBF neural networks consists of various interconnected neurons or nodes. The main difference arises from the type of transfer function used. A typical radial basis transfer function is the Gaussian:

$$z = \exp\left(-\frac{(x-c)^2}{r^2}\right) \quad (3.24)$$

where  $c$  is the centre and  $r$  the radius. This RBF is more commonly used than multiquadric-type RBFs (Orr, 1996), and decreases monotonically with distance from the centre while giving a significant response only in a neighbourhood near its centre (figure 3.7).



**Figure 3.7: Typical radial basis transfer function (Demuth and Beale, 2001)**

The multiquadric-type RBF is described by:

$$z = \frac{\sqrt{r^2 + (x - c)^2}}{r} \quad (3.25)$$

and monotonically increases with distance from the centre.

Although radial functions could be used to build both linear and non-linear models, its application (Orr, 1996) to the research done for this thesis deviates somewhat from that used for non-linear model building. Whereas RBF neural networks are non-linear when more than one hidden layer exists in the network or if the basis functions can move or change size, the focus in Orr's (1996) work is on single-hidden-layer networks (Broomhead and Lowe, 1988) containing radial functions of fixed position and size with a single, linear output node. Non-linear optimisation is, however, used for determination of the regularisation parameters used in ridge regression and during selection of the optimal subset of basis functions using forward selection methods. The kind of expensive and more generally applied gradient descent algorithms that are usually employed in explicitly non-linear neural networks is therefore also avoided.

The RBF neural network structure will be similar to that illustrated in figure 3.6, but for a few changes. The input nodes merely distribute the input values to the hidden nodes and are not weighted, performing a non-linear mapping from the input space to the hidden space. From the hidden space the values are linearly mapped to the output space, being a single node. Unlike the MLP neural network, the RBF neural network therefore represents a mapping from the  $b$ -dimensional input space to a one-dimensional output space.

### 3.3.2.2 Model training

Training of the RBF neural network is accomplished through forward subset selection (Miller, 1990) having a single regularisation parameter implemented through ridge regression (Orr, 1995; Orr, 1996), using orthogonal least squares.

A major problem when training RBF neural networks is the choice of RBF centres, which determines the number of free parameters of the model. A network having too few RBF centres will in all probability not be capable of fitting the data to an acceptable degree. In contrast with this, a network having too many RBF centres may fit misleading variations owing to imprecise or noisy data. Fortunately this problem is easily overcome using forward subset selection. Forward subset selection starts with an empty subset (RBF network) to which is added one basis function at a time from a candidate set of hidden units with different positions but the same size. Addition of basis functions continues until some chosen model selection criterion stops decreasing, resulting in the lowest prediction error for validation data. Although the prediction error of the network will never increase with the addition of a basis function, the model selection criterion will start to increase as overfitting sets in at which point the addition of basis functions should cease. The choice of basis function is determined by the sum-squared-error:

$$S = \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (3.26)$$

where  $y$  is the observed output and  $\hat{y}$  is the network output; the basis function reducing the sum-squared-error being added to the network. The obvious choice for the model selection criterion is certainly the minimisation of the mean square error of the model. However, this will focus too much attention to the unique peculiarities (mostly noise) of the model training dataset making the model less than ideal when applied to an unseen dataset. This problem is easily overcome using a model selection criterion where the sum-squared-error is modified to take the effective number of model parameters ( $\gamma$ ) into account. Possible model selection criterion includes: the unbiased estimate of variance (*UEV*), the final prediction error (*FPE*),

the generalised cross-validation criterion (*GCV*) or the Bayesian information criterion (*BIC*).

$$UEV = \frac{1}{(N - \gamma)} SSE \quad (3.27)$$

$$FPE = \frac{N + \gamma}{N(N - \gamma)} SSE \quad (3.28)$$

$$GCV = \frac{N}{(N - \gamma)^2} SSE \quad (3.29)$$

$$BIC = \frac{N + (\ln(N) - 1)\gamma}{N(N - \gamma)} SSE \quad (3.30)$$

Here  $\gamma$  is a function of the regularisation parameter,  $\lambda$ , and *SSE* is a modified sum-squared-error with the inclusion of a penalty function to reduce the model complexity:

$$SSE = \sum_{i=1}^N (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^m w_j^2 \quad (3.31)$$

where  $w$  is the network weights. This inclusion of a penalty function to the sum-squared-error is termed ridge regression. Whereas ridge regression is used to penalise large network weights, the regularisation parameter is used to control the strength of the penalty.

Although forward selection has the following advantages over the more conventional gradient descent algorithms (Orr, 1996):

- The number of hidden units is not fixed in advance.
- The model selection criteria are of a high standard.
- The computational requirements are relatively low.

it is further improved upon by a technique called orthogonal least squares (Chen et al., 1991). This technique ensures that each new column added to the design matrix of the growing subset is orthogonal to all previous columns, simplifying the equation for the change in sum-squared-error, resulting in a faster, more efficient algorithm. Combining this adapted forward selection algorithm with ridge regression leads to a modest improvement in performance (Orr, 1995) by allowing the optimisation of the regularisation parameter, through re-estimation, between the additions of new basis functions. The addition of ridge regression also serves as an additional safeguard against overfitting.

### 3.3.2.3 Comparison with MLP neural networks

Although both RBF neural networks and MLP neural networks can be applied to the same problem for a similar solution, their structures are very different, making it important to take note of the following summary (Bishop, 1999):

- For a MLP neural network many hidden units will typically contribute to the determination of an output value whereas a RBF neural network with localised basis functions, typically requires only a few hidden units to have significant activations for a given network input.
- MLP neural networks often have many layers of weights with a variety of different activation functions being used within the same network. RBF neural networks generally consists of only two layers of weights, the first layer containing the basis functions and the second layer forming linear combinations of the activations of these basis functions.
- Due to the nature of MLP neural networks they require a highly non-linear training process whereas RBF neural networks does not require full non-linear optimisation of the network for the choosing of suitable parameters for the hidden units.

### 3.3.3 **Least squares support vector machines**

Both MLP and RBF neural networks as described above, have been successfully applied to a wide range of problems. However, these networks still contain a number of weak points that include the existence of many local minima solutions and how to choose the number of hidden units. Although many possible solutions to these

problems have been suggested, few ever prove to be successful. SVMs, a new class of kernel based techniques developed by Vapnik (1995), bypass these problems by having its solutions characterised by convex optimisation problems (more specifically a quadratic programming problem), outperforming both MLP and RBF neural networks (Mukherjee et al., 1997). Solving this quadratic programming problem also impart the model complexity (number of hidden units) needed to approximate the underlying function given the training data.

The Matlab functions used for LS-SVM model development were based on the LS-SVM approximator developed by Suykens et al. (2000). These LS-SVMs enables highly non-linear and noisy black-box modelling problems to be solved, even in high dimensional input spaces (Suykens, 2001), by solving a linear system instead of a quadratic programming problem.

### 3.3.3.1 Model structure and training

From Suykens (2000) it follows that the LS-SVM model for function estimation has the following representation in feature space:

$$y(x) = w^T \varphi(x) + b \quad (3.32)$$

where the non-linear mapping,  $\varphi(\cdot)$ , maps the input data into a higher dimensional feature space, with  $w$  allowed to be infinitely dimensional. For the training set  $\{x_k, y_k\}_{k=1}^N$  the following optimisation problems has to be considered:

$$\min_{w,b,e} \mathfrak{J}(w,e) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{k=1}^N e_k^2 \quad (3.33)$$

subject to the equality constraints

$$y_k = w^T \varphi(x_k) + b + e_k, \quad k = 1, \dots, N \quad (3.34)$$

In equation 3.33 the cost function with squared error and regularisation corresponds to a form of ridge regression. This is similar to that found in the RBF neural network training cost function (equation 3.31). The Lagrangian is now given by

$$L(w, b, e, \alpha) = \mathfrak{J}(w, e) - \sum_{k=1}^N \alpha_k \{w^T \varphi(x_k) + b + e_k - y_k\} \quad (3.35)$$

where  $\alpha_k$  are Lagrangian multipliers (support vectors). The conditions for optimality are given as:

$$\begin{cases} \frac{\partial L}{\partial w} = 0 \rightarrow w = \sum_{k=1}^N \alpha_k \varphi(x_k) \\ \frac{\partial L}{\partial b} = 0 \rightarrow \sum_{k=1}^N \alpha_k = 0 \\ \frac{\partial L}{\partial e_k} = 0 \rightarrow \alpha_k = \gamma e_k \\ \frac{\partial L}{\partial \alpha_k} = 0 \rightarrow w^T \varphi(x_k) + b + e_k - y_k = 0, \quad k = 1, \dots, N \end{cases} \quad (3.36)$$

with the solution given by the following set of linear equations

$$\left[ \begin{array}{c|c} 0 & 1_v^T \\ \hline 1_v & \Omega + I/\gamma \end{array} \right] \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} \quad (3.37)$$

where  $y = [y_1; \dots; y_N]$ ,  $1_v = [1; \dots; 1]$ ,  $\alpha = [\alpha_1; \dots; \alpha_N]$  and the Mercer condition applied (in order to relate the non-linear feature space mapping to the chosen kernel function):

$$\begin{aligned} \Omega_{kl} &= \varphi(x_k)^T \varphi(x_l), \quad k, l = 1, \dots, N \\ &= K(x_k, x_l) \end{aligned} \quad (3.38)$$

The resulting LS-SVM model for function estimation becomes

$$y(x) = \sum_{k=1}^N \alpha_k K(x, x_k) + b \quad (3.39)$$

where  $\alpha_k, b$  are the solution to the linear system. Application of the Mercer conditions allows the solution of support values to be computed in the dual space instead of the primal weight space where the constrained optimisation problem is formulated. Several possibilities exist for the choice of kernel function,  $K(x, x_k)$ , including linear, polynomial, splines, radial basis functions, and multi-layer perceptions with one hidden layer. For the research done for this thesis the focus was on using the Gaussian RBF kernel. Using RBF kernels only adds two additional tuning parameters,  $\gamma, \sigma$ , which is less than for standard SVMs. These parameters are optimised by testing a set of LS-SVM corresponding to any one combination of  $(\gamma, \sigma)_i$  in a proposed grid of  $(\gamma, \sigma)$  after estimating the particular LS-SVM. The optimal  $(\gamma, \sigma)_i$  combination will result from a minimal error on the validation data set, leading to a model with a good ability to generalise well.

Owing to the use of a least squares cost function by LS-SVMs, a potential drawback exists in the possible lack of sparseness in the solution vector as is evident from the lack of  $\alpha_k$  values in the solution being exactly equal to zero when compared to standard SVMs. This problem can fortunately be overcome using pruning procedures such as sparse approximation.

### 3.3.3.2 Sparse approximation by LS-SVM

Initially it seems that for LS-SVMs sparseness is lost due to the choice of the 2-norm and the fact that the support values are proportional to the errors at the data points:

$$\alpha_k = \gamma e_k \quad (3.40)$$

However, a spectrum of support values exists and by plotting this spectrum of sorted  $|\alpha_k|$  values it can be evaluated which data are most significant for contribution to the LS-SVM. Sparseness is imposed by gradually omitting the least important data from the training set and re-estimating the LS-SVM. Suykens (2000) proposed the following algorithm for sparse approximation using LS-SVMs:

1. Train the LS-SVM based on  $N$  points.
2. Remove a small amount of points with the smallest values in the sorted  $|\alpha_k|$  spectrum.
3. Using the reduced training set to re-train the LS-SVM.
4. Repeat from 2, unless the user-defined performance index degrades.

Omitting data points implicitly correspond to creating an  $\varepsilon$ -insensitive zone in the underlying cost function which in turn leads to sparseness.

### 3.3.3.3 Advantages over MLP and RBF neural networks

Advantages of LS-SVM over more traditional neural networks include:

- SVMs embody the Structural Risk Minimisation (SRM) principle as opposed to the more traditional Empirical Risk Minimisation (ERM) principle employed by more conventional neural network. SRM minimises an upper bound on the generalisation error, compared to ERM which minimises the error on training data, making it superior to ERM (Gunn et al., 1997) and equips SVMs with a greater ability to generalise.
- The number of hidden units in a SVM with RBF kernels follows from solving the convex optimisation problem, being an additional advantage in comparison with regularisation theory of RBF neural network.
- When RBF kernels are used no need exists to select the kernel width and regularisation parameter, as these are implicitly determined during the model training process.
- In comparison with the pruning of classical neural network techniques, there is no need for the computation of a Hessian matrix during LS-SVM pruning. LS-SVM pruning is based only upon the physical meaning of the solution vector,  $\alpha$ , and all necessary pruning information is obtained from the solution of the linear system itself.
- Unlike MLP and RBF neural networks, SVM models scale very well to high-dimensional input spaces (Suykens, 2001). Whereas the number of weights to be estimated for MLP neural networks increases with an increase in input space dimensionality, it is not the case for LS-SVMs.

### 3.4 Prediction

Once the neural network has learnt to generalise the underlying trends or relationships demonstrated by the data, as determined through proper validation of the model, training is stopped. Model validation is often based on linear statistical tests such as the  $R^2$  statistic, defined as:

$$R^2 = 1 - \frac{\sum (y - \hat{y})^2}{(N - 1)\sigma_y^2} \quad (3.41)$$

where  $y$  is the observed state,  $\hat{y}$  the simulated state,  $\sigma$  the standard deviation of  $y$  and  $N$  the number of data points, or the RMS criterion. These statistics are usually employed during the application of either one-step prediction or free-run prediction. Whereas one-step prediction is not necessarily a good indicator of the ability of the model to generalise the underlying process represented by the data (Zhu and Rohwer, 1996), free-run prediction is a considerably more rigorous test of the validity of the model (Small and Judd, 1998b). However, neither of these statistics evaluates the local dynamic correspondence between the model and observations and is only static, global measures.

Another useful tool for validating models is P-P plots. If one assumes that the model fit to the data is correct, the residuals approximate the random errors that make the relationship between the explanatory variables and the response variable a statistical relationship. If the residuals appear to behave randomly, it suggests that the model fits the data well. However, if non-random structure is evident in the residuals, it is a clear sign that the model fits the data poorly. For a P-P plot, the cumulative proportion for a single numeric variable is plotted against the cumulative proportion expected if the sample were from a normal distribution. If the points tend towards clustering around a straight line it could be said that the sample is from a normal distribution, and consequently it can be concluded that the model has extracted all possible information from the data. However, should the data tend to deviate from a straight line, it can be concluded that the model is not necessarily a good representation of the data.

Once properly validated and training has stopped, the model should be able to sensibly interpolate at points not contained in its training set, ensuring its reliable application on new observations from the same process. This is not always the case. Lillekjendlie et al. (1994) have found that even with perfect reconstruction of the dynamical equations of a noise free chaotic system, only short-term predictions were possible. This was ascribed to the fact that the system investigated, while globally constrained to a finite region of space, is locally unstable everywhere, causing extreme sensitivity in the chaotic system to uncertainties in initial conditions. For a chaotic, non-linear deterministic system a finite prediction horizon exists regarding free-run prediction, because irrespective of closely defined initial conditions the time series and the predictor trajectories may increasingly diverge with time and eventually be uncorrelated with each other (Kanjilal, 1995). It is therefore important to establish the prediction horizon.

Using the first Lyapunov exponent, the prediction horizon of a non-linear dynamic system can be estimated. Starting a prediction from index  $i$  and predicting in  $L$  steps until  $i + L$ , an upper bound on  $L$  can be approximated by

$$t_L = \frac{\tau_s}{\lambda_1} \quad (3.42)$$

where  $\tau_s$  is the sampling period of the observer function and  $\lambda_1$  the first Lyapunov exponent. The upper bound on  $L$  is also referred to as the instability horizon (Abarbanel, 1996).

## 4 Analysis of plant data

---

Any time series should be carefully examined, using even the most rudimentary of data analysis methods, as part of system identification. Familiarisation with the data is not only a basic obligation and necessity, but raw data often contain features that render them unsuitable for many tests, including many standard statistical tests. Familiarisation should therefore give insight into the nature of the data, emphasize problem areas in the data, and help with the selection of the modelling technique required for successful prediction.

Preliminary analysis of the time series data was performed as outlined by Williams (1999):

1. Plot each time series and visually inspect the resulting graph. This will help in selecting a suitable continuous time series for modelling, recognising long-term trends in the data, and ensuring that the data selected for modelling is representative of the process.
2. Apply Fourier analysis to each time series and inspect the resulting power spectrum. Fourier analysis is used to detect periodicity in data by mathematically describing a time series in terms of periodic constituents, giving special attention to the relative strengths of the periodic constituents.
3. Test the data for linear autocorrelation. Should autocorrelation be present in the data it can be ignored, or removed, or the system can be declared (whether rightly or wrongly) as not being chaotic. The means of handling autocorrelation will depend on the type of data, the extent of autocorrelation, and the results obtained from handling the autocorrelation differently. Additional statistics, not mentioned by Williams (1999) but included here, is the calculation of the point of general de-correlation using AMI and no self-intersection using FNN.

Finally the time series data were tested for determinism and non-linearity using the method of surrogate data as described in chapter 3.

## 4.1 Data inspection

Listed below (table 4.1) is a summary of all the available measurements for the Amandelbult UG2 FAG mill.

**Table 4.1: Measured variables for the Amandelbult UG2 FAG mill**

Variable	Dataset	Description	Units
$x_1$	Process	FAG mill power	kW
$x_2$	Process	FAG mill load	T
$x_3$	Process	FAG mill Coarse feed rate (calculated)	T/hr
$x_4$	Process	FAG mill Fine feed rate	T/hr
$x_5$	Process	FAG mill total screen feed rate (coarse + fine ore)	T/hr
$x_6$	Process	FAG mill feed rate (measurement unreliable)	T/hr
$x_7$	Process	FAG mill inlet water flow rate	m <sup>3</sup> /hr
$x_8$	Process	FAG mill motor reference speed	%
$x_9$	Process	FAG mill discharge 1 flow rate	m <sup>3</sup> /hr
$x_{10}$	Process	FAG mill discharge 2 flow rate	m <sup>3</sup> /hr
$x_{11}$	Process	FAG mill rougher feed 1 flow rate	m <sup>3</sup> /hr
$x_{12}$	Process	FAG mill rougher feed 2 flow rate	m <sup>3</sup> /hr
$x_{13}$	Wipfrag	FAG mill feed size distribution (% -31.5mm)	%
$x_{14}$	Malvern	FAG mill product size distribution (% -75 $\mu$ m)	%

### 4.1.1 Raw data analysis

Table 4.2 present some basic statistics calculated for each time series based on all available data and not on reduced subsets. Process data were collected for the periods 27/12/2000 – 30/04/2001 and 16/07/2001 – 11/09/2001, Wipfrag data for the period 12/06/2001 – 26/09/2001, and Malvern data for the period 04/05/2001 – 17/09/2001.

For the data to be used as a whole, the Process, Wipfrag and Malvern data had to be concatenated. From the data collection periods it can be seen that only for a relatively small time interval data collection from all three systems were overlapping. A further problem is the difference in sampling interval, all three systems having different sampling intervals. Whereas the Process data is logged consistently at 30 [s] intervals, the Wipfrag logs data relatively consistently at an average interval of 1 [min], and the Malvern samples data very inconsistently at an average interval of 20 [min]. Problems were also experienced with breaks in the data resulting from either the plant being down or the instrumentation malfunctioning. Based on these problems and the need for a continuous time series for dynamic modelling it was decided to analyse the data from the Process, Wipfrag and Malvern systems separately.

**Table 4.2: Summary of statistics calculated for the measured variables**

Variable	Number of data points	Average sampling interval	Mean	Min	Max	Std Dev
x <sub>1</sub>	527045	30 seconds	2315.00	0.00	5000.00	1014.70
x <sub>2</sub>	527045	30 seconds	139.71	0.00	207.18	13.80
x <sub>3</sub>	527045	30 seconds	57.59	0.00	100.00	24.45
x <sub>4</sub>	527045	30 seconds	267.75	0.00	400.00	139.93
x <sub>5</sub>	527045	30 seconds	291.19	0.00	500.00	146.54
x <sub>6</sub>	527045	30 seconds	119.03	0.00	450.00	150.80
x <sub>7</sub>	527045	30 seconds	35.58	0.00	300.00	23.61
x <sub>8</sub>	527045	30 seconds	20.95	0.00	95.00	10.10
x <sub>9</sub>	527045	30 seconds	155.86	0.00	1000.00	325.18
x <sub>10</sub>	527045	30 seconds	268.20	0.00	1000.00	150.06
x <sub>11</sub>	527045	30 seconds	276.77	0.00	1000.00	340.41
x <sub>12</sub>	527045	30 seconds	319.12	0.00	1000.00	268.74
x <sub>13</sub>	45100	1 minute	46.32	9.00	100.00	11.73
x <sub>14</sub>	4210	20 minutes	42.50	0.00	100.00	22.31

When looking at the accuracy of the data, it can be seen that the data contains a great deal of variation making it ideal for modelling with regard to being representative of the process. For the Process dataset, containing the largest amount of data points, more reliable mean values should exist than for the Wipfrag and Malvern datasets, but owing to the large standard deviations calculated for these time series it cannot be said for certain that the calculated mean values are indeed reliable. Similar reasoning leads one to also doubt the reliability of the mean values calculated for the Wipfrag and Malvern datasets. This, combined with the problem of numerous discontinuities in the various time series, led to the selection of subsets from the original datasets. These subsets were constructed from the largest available continuous time series for each variable. The Process subset,  $x_1 - x_{12}$ , consists of data collected for the period 06/03/2001 – 09/03/2001, the Wipfrag subset,  $x_{13}$ , consists of data collected for the period 12/06/2001 – 26/06/2001, and the Malvern subset,  $x_{14}$ , consists of data collected for the period 16/08/2001 – 02/09/2001.

#### 4.1.2 Subset data analysis

Table 4.3 present some basic statistic calculated for each of the selected subsets. It is evident that the data in the subsets contains less variation than that of the full datasets but still covers a sufficiently large range for each variable. These smaller standard deviation values, combined with the fact that these subsets are continuous time series

without any discontinuities in the data, allow these subsets to be used for further analysis while ensuring more reliable results.

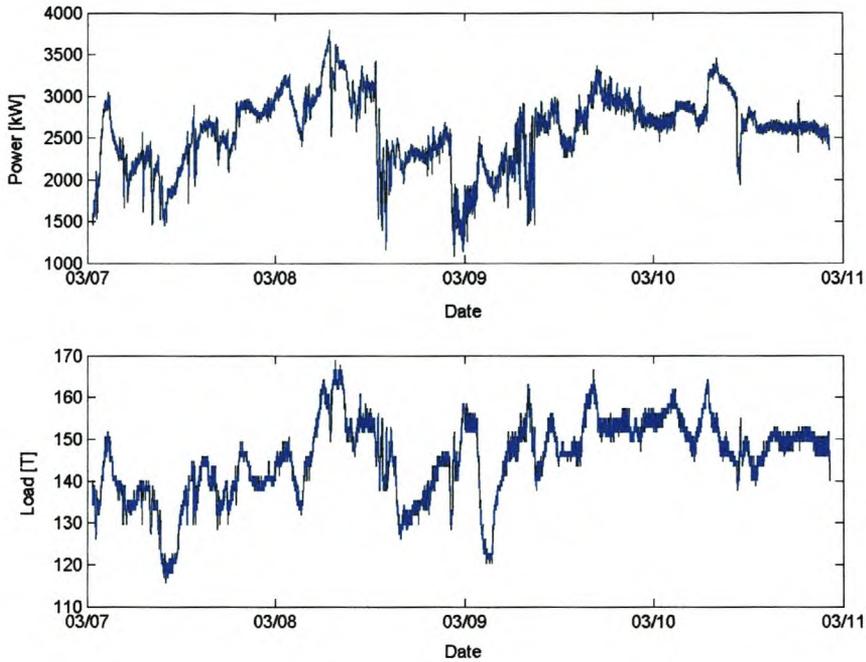
**Table 4.3: Summary of statistics calculated for data subsets**

Variable	Number of data points	Mean	Min	Max	Std Dev
$x_1$	11241	2586.00	1083.60	3800.60	433.02
$x_2$	11241	145.48	115.74	168.98	9.38
$x_3$	11241	60.51	0.00	75.00	13.11
$x_4$	11241	316.93	0.12	400.00	99.20
$x_5$	11241	349.87	0.00	500.00	87.17
$x_6$	11241	238.08	0.26	450.00	69.40
$x_7$	11241	42.96	0.17	100.17	11.73
$x_8$	11241	16.31	15.00	40.00	3.47
$x_9$	11241	40.27	0.00	1000.00	96.70
$x_{10}$	11241	267.29	1.45	722.51	103.47
$x_{11}$	11241	33.63	0.29	563.37	107.89
$x_{12}$	11241	324.08	0.29	587.96	117.22
$x_{13}$	20704	48.70	23.90	94.90	4.84
$x_{14}$	1069	43.39	1.80	100.00	8.15

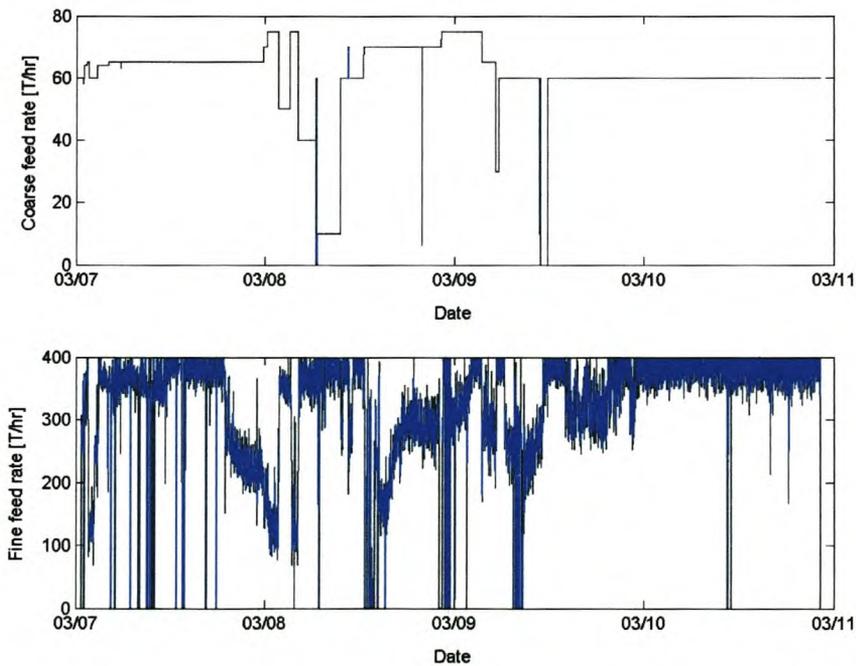
The subset time series in table 4.3 are illustrated in figure 4.1 to figure 4.7. From figure 4.1 it is evident that the mill was run to cover a significant part of its operational range with a high degree of variation. It can also be seen that a strong correlation exists between the FAG mill power drawn ( $x_1$ ) and the FAG mill load ( $x_2$ ), which is as expected (Napier-Munn et al., 1999). The power draw therefore does not change in direct response to feed rate, feed hardness, feed size distribution, or any other external influence, but changes due to subsequent changes in the mill load caused by the aforementioned effects. From figures 4.2 and 4.3 it is clear that the FAG mill fine feed rate ( $x_4$ ) and the FAG mill feed rate ( $x_6$ ) values were often saturated at the top of their respective instrument measurement ranges leading one to believe these values as being unreliable.

For the FAG mill discharge line and the rougher feed line the secondary pumps (variables  $x_{10}$  and  $x_{12}$ ) were predominantly used (figure 4.5 and figure 4.6). Having only one of two possible pumps being mainly used during the time period from which the subset time series was selected may lead to model training problems that will be addressed once encountered. It should also be noted that no pump and its standby

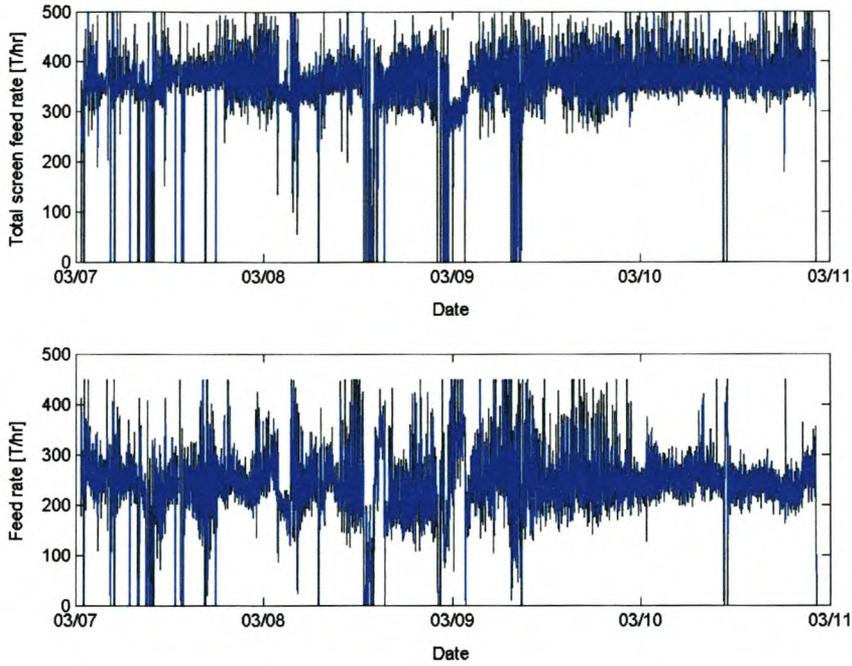
pump is normally run simultaneously. They are flushed before and after use with water; a lower flow rate value usually indicating that the pump is being flushed. In spite of this it is suspected that the irregularities in the FAG mill discharge 1 flow rate ( $x_9$ ) resulted from instrumentation malfunction.



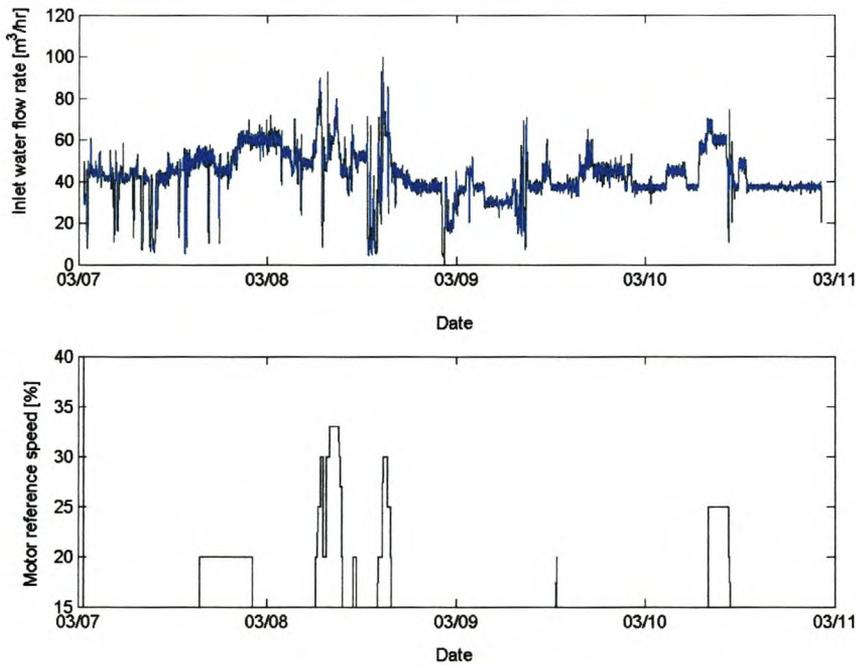
**Figure 4.1: Plot of FAG mill power ( $x_1$ ) and load ( $x_2$ )**



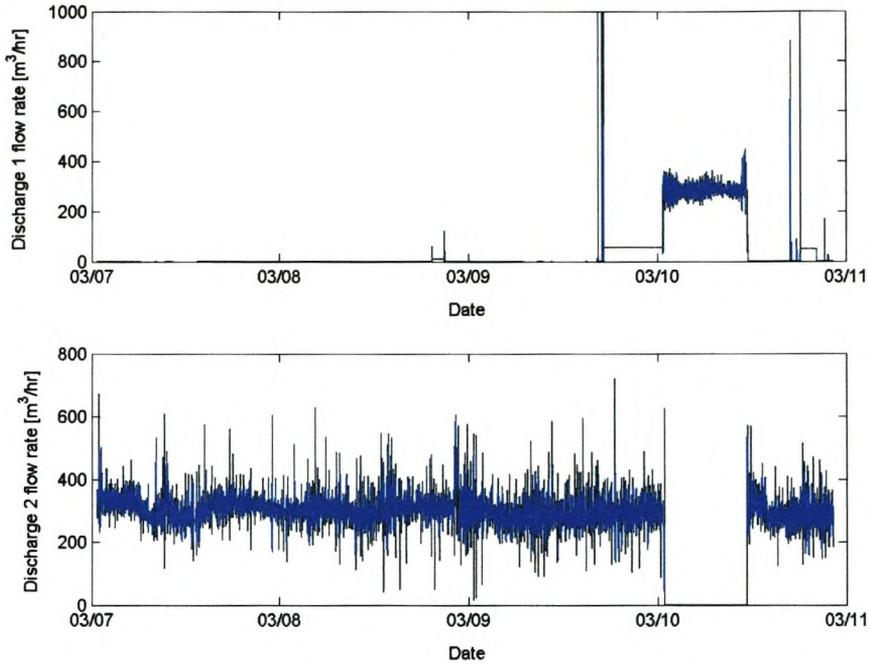
**Figure 4.2: Plot of FAG mill coarse feed rate ( $x_3$ ) and fine feed rate ( $x_4$ )**



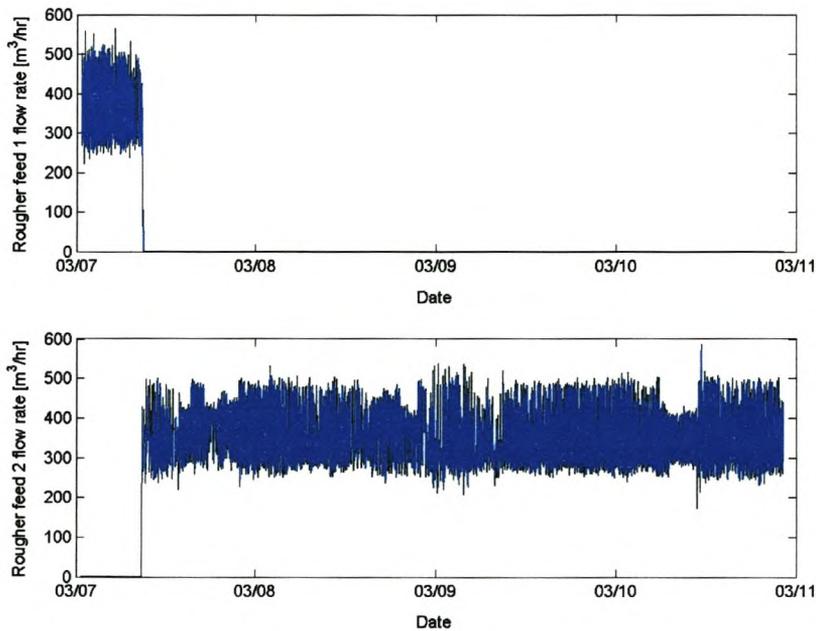
**Figure 4.3: Plot of FAG mill total screen feed rate ( $x_5$ ) and feed rate ( $x_6$ )**



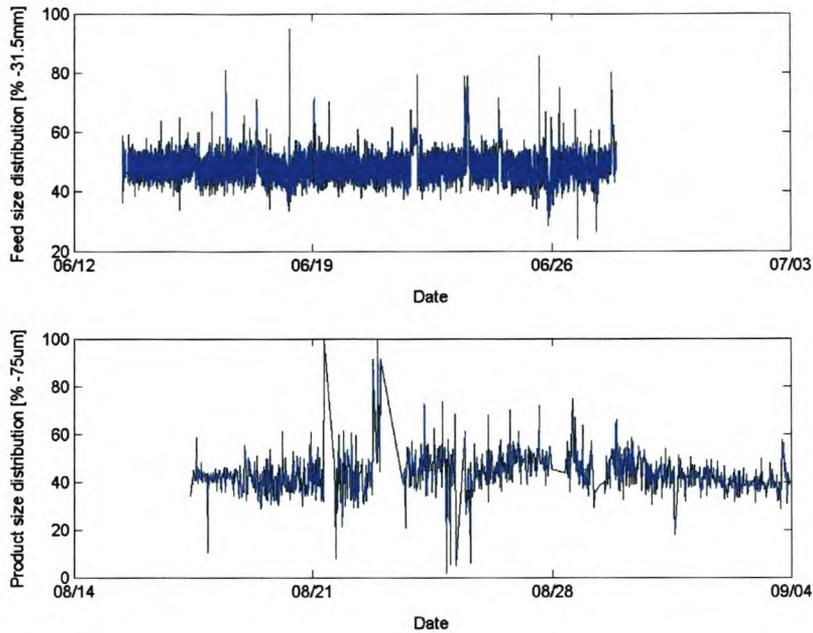
**Figure 4.4: Plot of FAG mill inlet water flow rate ( $x_7$ ) and motor reference speed ( $x_8$ )**



**Figure 4.5: Plot of FAG mill discharge 1 flow rate ( $x_9$ ) and discharge 2 flow rate ( $x_{10}$ )**



**Figure 4.6: Plot of FAG mill rougher feed 1 flow rate ( $x_{11}$ ) and rougher feed 2 flow rate ( $x_{12}$ )**



**Figure 4.7: Plot of FAG mill feed size distribution % -31.5mm ( $x_{13}$ ) and product size distribution % -75 $\mu$ m ( $x_{14}$ )**

From the correlation coefficient matrix calculated from the Process data subset (table 4.4) it can be seen that FAG mill power ( $x_1$ ) is, as previously noticed, significantly correlated to the FAG mill load ( $x_2$ ) and FAG mill inlet water flow rate ( $x_7$ ); in both instances the power drawn increases as the load or inlet water flow rate increases. The FAG mill power ( $x_1$ ) was also found to be significantly correlated to the FAG mill reference speed ( $x_8$ ). Other significant findings include:

- Correlation between FAG mill fine feed ( $x_4$ ) and FAG mill total screen feed ( $x_5$ ); total screen feed being the sum of fine feed and coarse feed.
- Correlation between FAG mill total screen feed ( $x_5$ ) and FAG mill feed rate ( $x_6$ ); the total screen feed after passing through the screening section becoming the mill feed rate.
- Correlation between FAG mill total screen feed ( $x_5$ ) and FAG mill inlet water flow rate ( $x_7$ ); the more ore fed to the circuit the more water is needed to counter accumulation in the circuit and ensure the correct slurry density.

- Correlation between FAG mill inlet water flow rate ( $x_7$ ) and FAG mill reference speed ( $x_8$ ); an increase in water being added to the mill washing more fines from the mill and in so doing reducing the mill load.
- Inverse correlation between FAG mill discharge 1 flow rate ( $x_9$ ) and FAG mill discharge 2 flow rate ( $x_{10}$ ); one pump being on standby while the other is running.
- Inverse correlation between FAG mill rougher feed 1 flow rate ( $x_{11}$ ) and FAG mill rougher feed 2 flow rate ( $x_{12}$ ); one pump being on standby while the other is running.

**Table 4.4: Pearson's correlation coefficient matrix (based on the bivariate method) calculated for the Process data subset. Correlations given in *italics* are significant at the 0.01 level (2-tailed) with correlations shaded in grey discussed earlier in the text.**

Variable	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$	$x_9$	$x_{10}$	$x_{11}$	$x_{12}$
$x_1$	1.00	-	-	-	-	-	-	-	-	-	-	-
$x_2$	<i>0.54</i>	1.00	-	-	-	-	-	-	-	-	-	-
$x_3$	<i>-0.46</i>	<i>-0.42</i>	1.00	-	-	-	-	-	-	-	-	-
$x_4$	<i>0.16</i>	<i>0.22</i>	<i>-0.23</i>	1.00	-	-	-	-	-	-	-	-
$x_5$	<i>0.26</i>	<i>0.15</i>	<i>-0.08</i>	<i>0.65</i>	1.00	-	-	-	-	-	-	-
$x_6$	<i>0.10</i>	<i>0.26</i>	<i>-0.03</i>	<i>0.32</i>	<i>0.48</i>	1.00	-	-	-	-	-	-
$x_7$	<i>0.54</i>	<i>0.18</i>	<i>-0.23</i>	<i>0.22</i>	<i>0.46</i>	<i>0.38</i>	1.00	-	-	-	-	-
$x_8$	<i>0.33</i>	<i>0.14</i>	<i>-0.43</i>	0.01	0.01	<i>0.05</i>	<i>0.45</i>	1.00	-	-	-	-
$x_9$	<i>0.27</i>	<i>0.30</i>	-0.01	<i>0.23</i>	<i>0.09</i>	<i>0.10</i>	<i>0.11</i>	<i>0.08</i>	1.00	-	-	-
$x_{10}$	<i>-0.24</i>	<i>-0.29</i>	<i>0.04</i>	<i>-0.24</i>	<i>-0.10</i>	<i>-0.10</i>	<i>-0.11</i>	<i>-0.09</i>	<i>-0.84</i>	1.00	-	-
$x_{11}$	<i>-0.23</i>	<i>-0.25</i>	<i>0.08</i>	<i>-0.05</i>	<i>-0.06</i>	0.01	<i>-0.07</i>	<i>-0.11</i>	<i>-0.12</i>	<i>0.17</i>	1.00	-
$x_{12}$	<i>0.21</i>	<i>0.17</i>	<i>-0.04</i>	<i>0.03</i>	<i>0.06</i>	<i>-0.05</i>	<i>0.08</i>	<i>0.09</i>	<i>0.08</i>	<i>-0.11</i>	<i>-0.83</i>	1.00

A strong inverse correlation was also found to exist between the FAG mill power ( $x_1$ ) and the FAG mill coarse feed rate ( $x_3$ ). This is consistent with an increase in coarse material, relative to fine material, being fed to the mill. When this occurs, more impact grinding is taking place in the mill as opposed to abrasion grinding. Whereas abrasion grinding is a very power inefficient process, generating much heat, impact grinding requires less energy per ton of material. An increase in coarse material fed to the mill could therefore result in a decrease in power drawn by the mill.

### 4.1.3 Results discussion

Based on the preceding discussion it was decided to reject the use of the FAG mill feed rate ( $x_6$ ) for further analysis. This decision is based on the fact that the instrument was often saturated at maximum range and the unreliability of the measurement due to instrumentation problems experienced. Although the instrument measuring the FAG mill fine feed rate ( $x_4$ ) was also often saturated at maximum range it was decided not to omit this measurement from further analysis owing to its importance in mill control.

It was decided that FAG mill motor reference speed ( $x_8$ ), although vital to mill control, be rejected from further analysis. The time series describing this variable was deemed to contain insufficient variability, the mill being operated at fixed speed and not variable speed during data collection, making it unsuitable to be used for modelling purposes. The use of the FAG mill total screen feed rate ( $x_5$ ) for further analysis was also rejected. This resulted from the variable being the sum of the FAG mill fine feed rate ( $x_4$ ) and the actual coarse ore fed to the mill, and therefore redundant.

## 4.2 Fourier analysis

Fourier analysis is one of the more common ways of finding periodicity, a type of structure, in a time series. Whereas a stochastic time series doesn't contain any periodicity, a chaotic time series may or may not contain periodicity. Fourier analysis can therefore not by itself indicate chaos, but it can also be used to reveal autocorrelation.

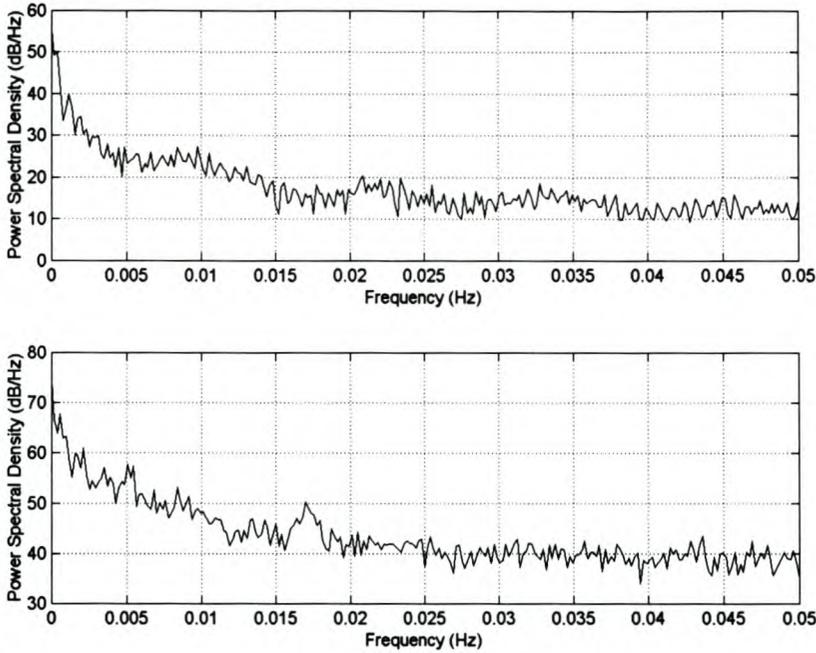
The process of determining the Fourier power spectrum is linear and assumes that a given time series can be decomposed into a sequence of sines and cosines by extracting the composition of these sines and cosines found in the time series. Fourier analysis is further limited by the requirement that the input data to be sampled at evenly spaced intervals. Although methods are available for analyzing irregularly or incompletely sampled data, they are significantly slower than the standard method

used for Fourier analysis. However, due to the nature of the data being analysed, the use of these methods for Fourier analysis is not deemed necessary.

#### **4.2.1 Fourier power spectrum**

The Fourier power spectrum was computed and plotted for all the time series contained in the reduced data subset. Since the results obtained for all the times series analysed were similar, only the Fourier power spectrum for the FAG mill coarse feed rate ( $x_3$ ) and FAG mill fine feed rate ( $x_4$ ) will be discussed in detail. The Fourier power spectra for the remaining time series are included in the appendix.

From figure 4.8 it can be seen that although sharp spectral lines are present at certain frequencies, no pronounced spikes are present in either of the plotted Fourier power spectrums. From this it can be concluded that none of the time series analysed contain definite periodicity (as indicated by pronounced spikes) and they can therefore be classified as either chaotic or stochastic. According to Williams (1999), a characteristic feature of a chaotic time series is a broadband (continuous) Fourier power spectrum (mostly uncorrelated noise) with some periodicities sticking up here and there. It can be seen that the plotted Fourier power spectra (figure 4.8) indeed contain broadband spectral features, this being an immediate consequence of the exponentially decaying autocorrelation function. However, Glass and Mackey (1988) stated that a broadband Fourier power spectrum without pronounced spikes could indicate either a chaotic or a stochastic process. Therefore, with the Fourier power spectrum only having a broadband (a necessary but not sufficient criterion of chaos) and no pronounced periodicities, no conclusions could be drawn as to whether the times series are chaotic or stochastic. However, further analysis of the data using the method of surrogate data could lead to more conclusive results.



**Figure 4.8: Fourier power spectrum for FAG mill coarse feed rate ( $x_3$ ) and FAG mill fine feed rate ( $x_4$ )**

It should be noted that during the Fourier analysis the Fourier power spectra for 1000 continuous observations from the subset time series was also determined. In some instances these contained broadband spectral features, as well as pronounced periodicities, indicating definite chaos. However, the Fourier power spectra for the complete subset time series always contained broadband spectral features, but never pronounced periodicities. From this it was concluded that although a part of a time series may be chaotic, it couldn't be said that the whole time series is then also chaotic.

### 4.3 Statistics for non-linear time series analysis

Calculation of the LA, AMI and FNN statistics is not only useful when analysing the linear or non-linear properties of a time series, but also forms the core when embedding a time series using the MOD. Whereas the LA function and the AMI statistic is used to calculate the embedding delay ( $k$ ), the FNN statistic is used to determine the embedding dimension ( $m$ ). Table 4.5 gives a summary of the results obtained from calculating the aforementioned statistics followed by a comprehensive discussion on the results obtained.

**Table 4.5: Linear Autocorrelation (LA), Average Mutual Information (AMI) and False Nearest Neighbours (FNN) statistics calculated for reduced subset data in table 4.3**

Variable	LA – FNN		AMI – FNN	
	Embedding Delay (LA)	Embedding Dimension (FNN)	Embedding Delay (AMI)	Embedding Dimension (FNN)
$x_1$	100	6	37	8
$x_2$	100	8	106	6
$x_3$	100	5	240	2
$x_4$	16	18	9	12
$x_7$	42	8	42	8
$x_9$	4	4	1	5
$x_{10}$	4	15	3	15
$x_{11}$	5	8	3	9
$x_{12}$	6	11	3	11
$x_{13}$	32	7	22	7
$x_{14}$	2	7	1	7

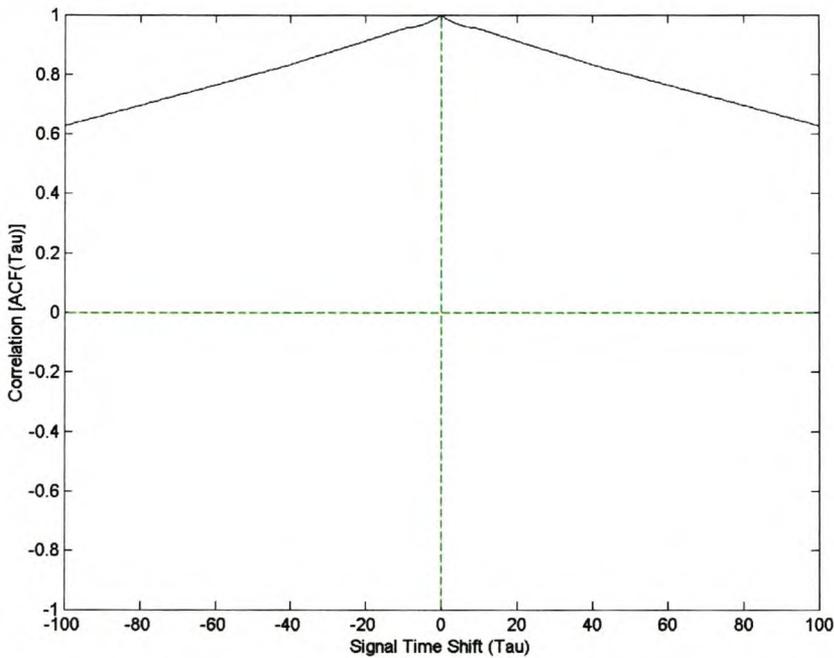
#### 4.3.1 Linear Autocorrelation statistic

The autocorrelation plot is commonly used for checking randomness in data by computing autocorrelations for data values at varying time lags, the result reflecting the extent to which a variable correlates with itself over time. For random data the autocorrelations should be near zero for any and all time-lag separations, whereas for non-random data one or more of the autocorrelations will be significantly non-zero. Autocorrelations were computed for the time series contained in the reduced subsets and the resulting plots assigned to one of three groups. Since the autocorrelation function is a linear statistic and the data highly non-linear, the results can only be seen as indicative of randomness, or lack thereof, in the data and therefore not conclusive. The linear autocorrelation plots for the respective time series not shown below are included in the appendix.

##### 4.3.1.1 Strong autocorrelation

From figure 4.9 it can be seen that the plot starts with a high autocorrelation at a lag (signal time shift) of 1 that slowly declines linearly with little noise. This autocorrelation plot of the FAG mill coarse feed rate ( $x_3$ ), together with those for the FAG mill power ( $x_1$ ) and the FAG mill load ( $x_2$ ), show a strong positive autocorrelation that should prove highly predictable if modelled properly. The

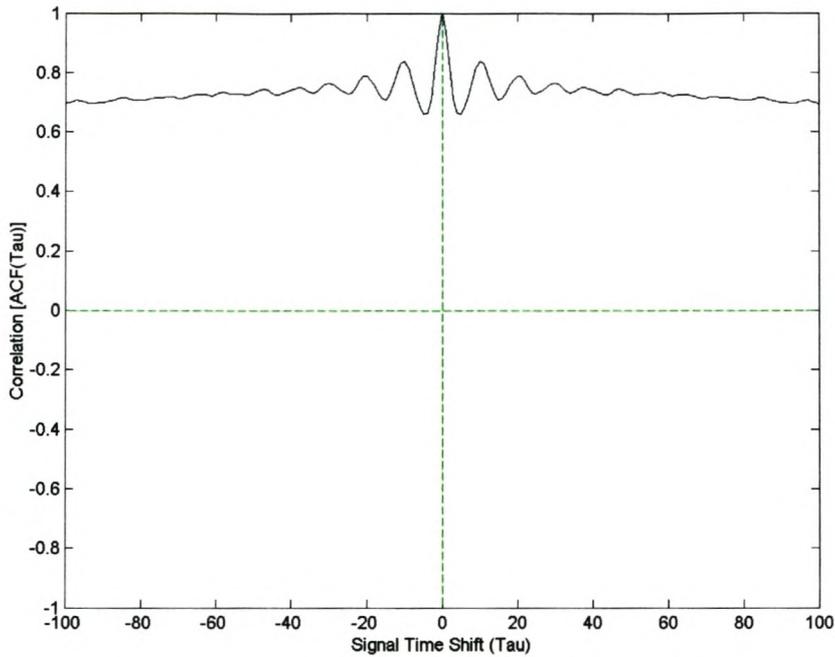
embedding delays for these variables were found to be much larger than the calculated values in table 4.5; the calculated values being an arbitrary maximum value set for the LA function. At a 30 second sampling interval the embedding delays of these variables correspond to a time interval of 50 minutes each. When considering the process at hand, it is evident that these estimated values are much too large. This shows the importance of considering the knowledge available on the process under investigation and not only blindly accepting the calculated statistics as final.



**Figure 4.9: Autocorrelation for FAG mill coarse feed rate ( $x_3$ )**

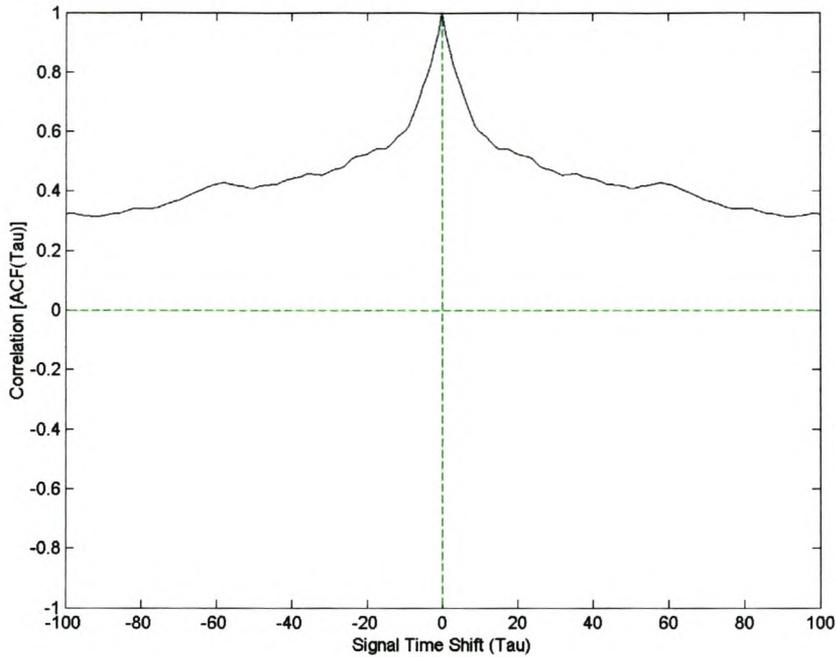
#### 4.3.1.2 Moderate autocorrelation

From figure 4.10 it can be seen that the plot starts with a moderately high autocorrelation at a lag of 1 that at first rapidly decreases but then forms a decreasing oscillatory movement. This autocorrelation plot of the FAG mill rougher feed 2 flow rate ( $x_{12}$ ), together those for the FAG mill rougher feed 1 flow rate ( $x_{11}$ ), the FAG mill discharge 1 flow rate ( $x_9$ ) and the FAG mill discharge 2 flow rate ( $x_{10}$ ), shows a general decay in the autocorrelation function which can be ascribed to moderate noise in the data. Moderate predictability could be expected from modelling this data.



**Figure 4.10: Autocorrelation for FAG mill rougher feed 2 flow rate (x<sub>12</sub>)**

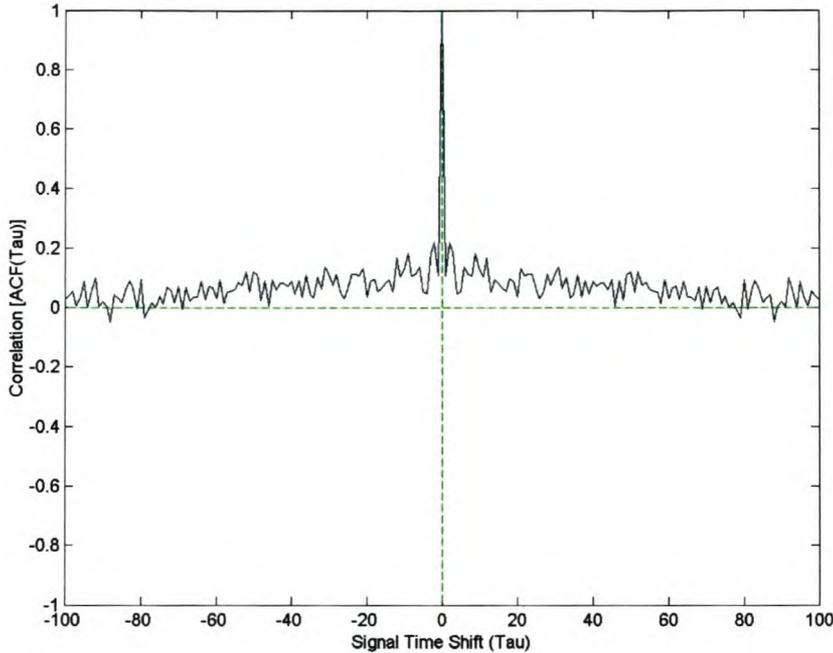
Similar to figure 4.10, figure 4.11 also starts with a moderately high autocorrelation at a lag of 1. However, the difference with this autocorrelation plot of the FAG mill fine feed rate (x<sub>4</sub>), as with the FAG mill inlet water flow rate (x<sub>7</sub>) and the FAG mill feed size distribution (x<sub>13</sub>), is that the autocorrelation function decays exponentially. This type of decrease in the autocorrelation function is indicative of a high degree of noise in the data. Again only moderate predictability could be expected from modelling this data.



**Figure 4.11: Autocorrelation for FAG mill fine feed rate ( $x_4$ )**

#### 4.3.1.3 No autocorrelation

From figure 4.12 it can be seen, that with the exception of the autocorrelation at a lag of 0, which is always 1 by definition, the calculated autocorrelations are all very low. This autocorrelation plot of the FAG mill product size distribution ( $x_{14}$ ) further shows no apparent pattern. Showing no pattern is indicative of uncorrelated random data or data with insignificant correlations and a very high degree of noise. The lack of autocorrelation in the data could also in part be ascribed to the fact that the sampling interval of the data was not always exactly 20 minutes, resulting in the inaccurate calculation of the linear autocorrelation statistic.



**Figure 4.12: Autocorrelation for FAG mill product size distribution % -75 $\mu$ m ( $x_{14}$ )**

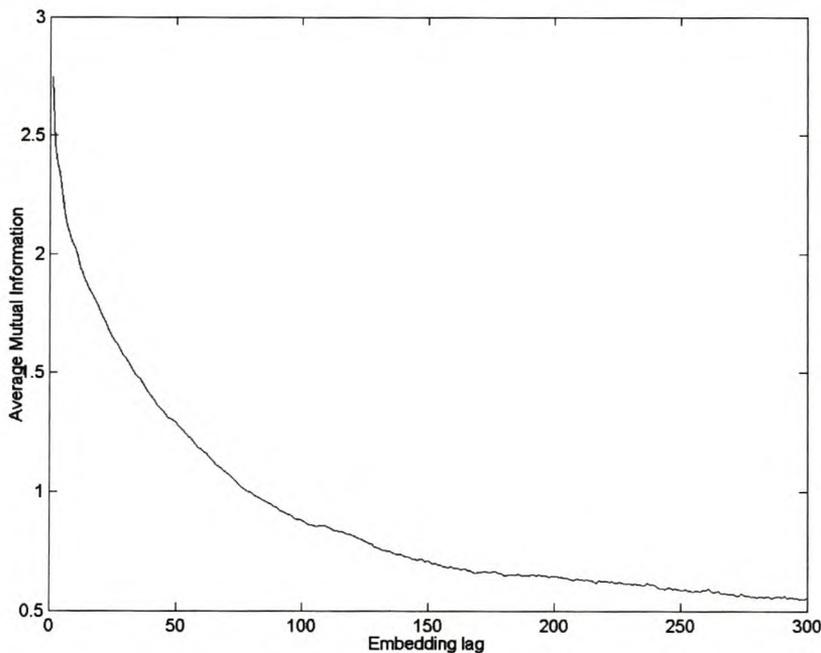
### 4.3.2 Average Mutual Information statistic

The Average Mutual Information statistic acts as a non-linear autocorrelation function, telling how much one learns about a measurement at different times on the average over all the data. For two coordinates of a point having the same value, and thus representing the same information, the one predicts the other exactly. For prediction purposes, having one value providing a lot of information about another is unwanted. However, since a high value for mutual information means strongly correlated values, the first minimum of the Average Mutual Information plot for a time series can be used to tell at which point the values become independent from one another.

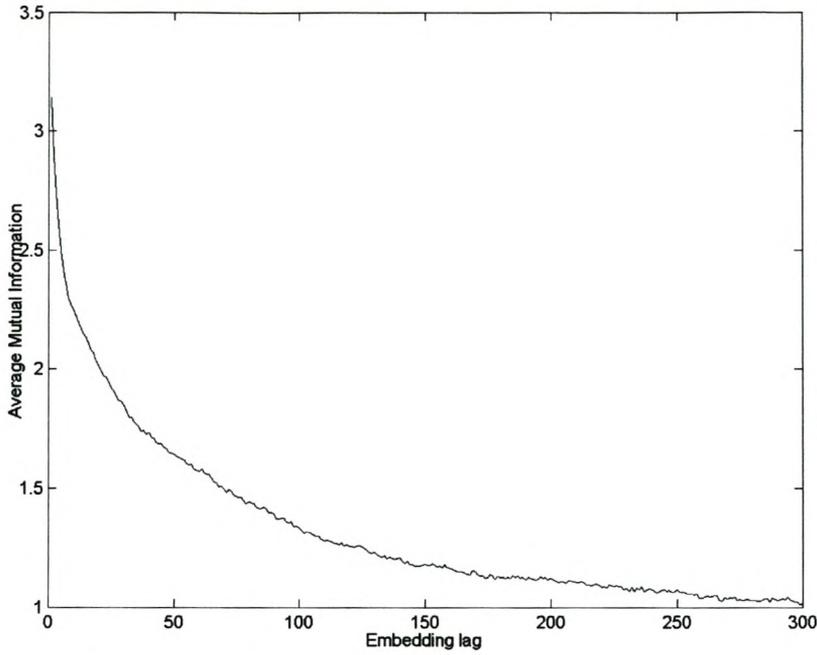
From figures 4.13 – 4.15 it can be seen that the AMI plots can be divided into three distinct groups. The plots contained in the first group, that of AMI for the FAG mill load ( $x_2$ ) and the FAG mill coarse feed rate ( $x_3$ ), is initially represented by a smooth line, only reaching a minimum after a substantial amount of delays; 106 and 240 respectively. Again, as with the LA function, it is evident that these estimated values

are much too large and the incorporation of process knowledge during the interpretation of the results is vital.

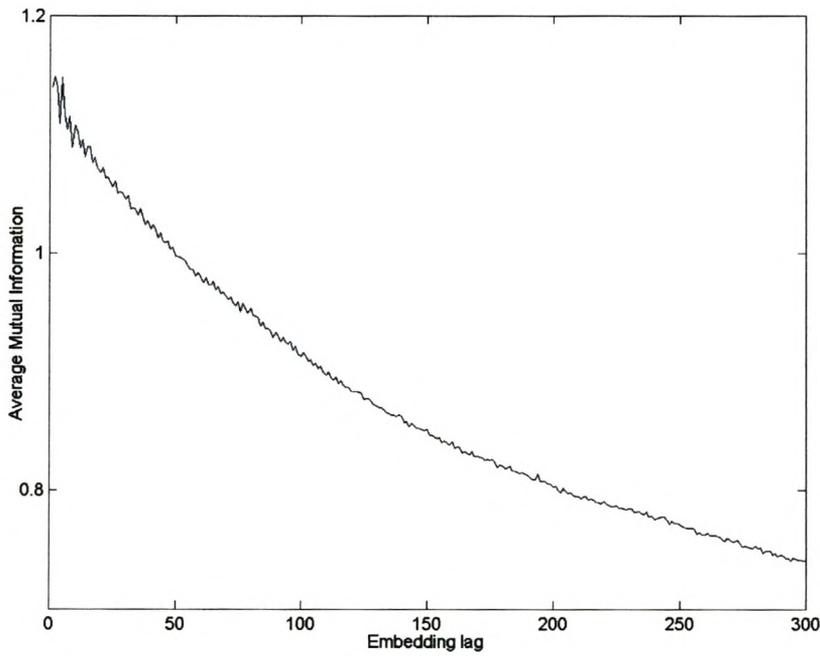
The second group contains the plots of the AMI for the FAG mill power ( $x_1$ ), the FAG mill fine feed rate ( $x_4$ ), the FAG mill inlet water flow rate ( $x_7$ ), and the FAG mill feed size distribution ( $x_{13}$ ). Here a minimum was reached between 9 and 42 delays, which when reached, caused the AMI statistic to start fluctuating (figure 4.14). For the plots contained in the third group, that of the AMI for the FAG mill discharge 1 flow rate ( $x_9$ ), the FAG mill discharge 2 flow rate ( $x_{10}$ ), the FAG mill rougher feed 1 flow rate ( $x_{11}$ ), the FAG mill rougher feed 2 flow rate ( $x_{12}$ ), and the FAG mill product size distribution ( $x_{14}$ ), the fluctuations are much more pronounced (figure 4.15). Here a minimum was reached between only 1 and 3 delays; the plots showing erratic behaviour with highly fluctuating AMI values. Whereas the fluctuations in the AMI plots could be ascribed to noise in the data, the increase in fluctuations between groups two and three, could be ascribed to an increase in the amount of noise in the data. For most of the time series analysed, the AMI results were comparable to the LA results.



**Figure 4.13: AMI plot of FAG mill load ( $x_2$ )**



**Figure 4.14: AMI plot of FAG mill power ( $x_1$ )**



**Figure 4.15: AMI plot of FAG mill discharge 1 flow rate ( $x_9$ )**

From figures 4.13 – 4.14 it is further evident that no distinct minimum AMI value was reached for the time series shown. In each instant only a shallow minimum was reached leading us to suspect that a larger choice of minimum (delay) may be

appropriate. However, a substantial decrease in AMI is noted before the first minimum is reached for groups one and two. This is indicative that the first minimum AMI value reached is sufficient for capturing the independent coordinates of the time series contained in these groups. It can thus be concluded that the minimum AMI values obtained for the time series contained in group three is possibly incorrect and should be adjusted if needed. The AMI plots for the respective time series not shown above are included in the appendix.

### 4.3.3 False Nearest Neighbours criterion

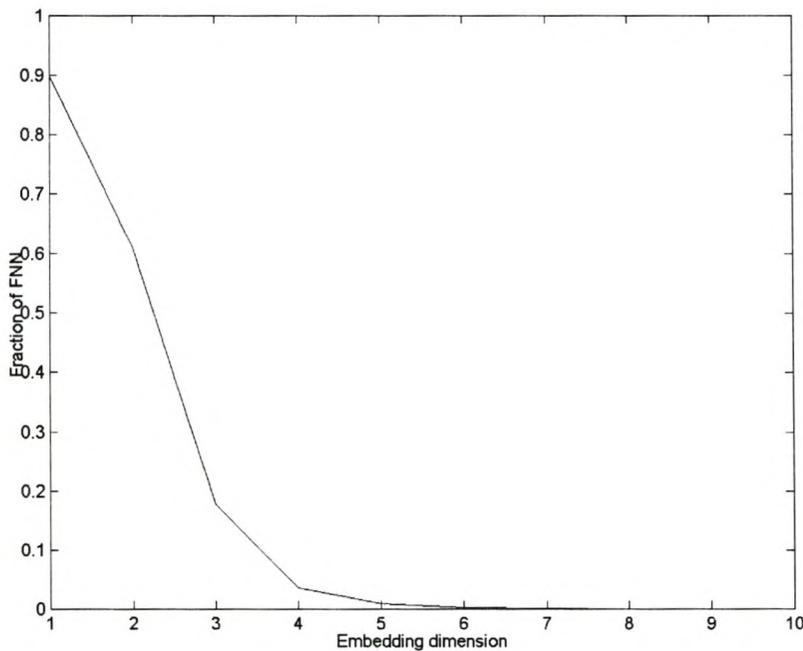
False nearest neighbours are points that seem to be close to each other in phase space because of too small an embedding space. Since such phase space points will be near each other for non-dynamical reasons, because of projection for instance, they are connected with unpredictability. The FNN statistic also provides us with a qualitative way to establish whether or not the time series being analysed has residual noise in it, allowing one to distinguish noise from low dimensional chaos (Abarbanel, 1996). Should noise be present in the time series, the residual FNN can furthermore be indicative of the level of contamination.

When comparing the embedding dimensions estimated using the embedding delays obtained for the LA function and the AMI statistic, it can be seen that similar embedding dimensions were calculated for each time series irrespective of the embedding delay used (table 4.5). From this it would seem as if the choice of embedding delay does not carry a significant weight when using the FNN statistic to calculate the embedding dimension.

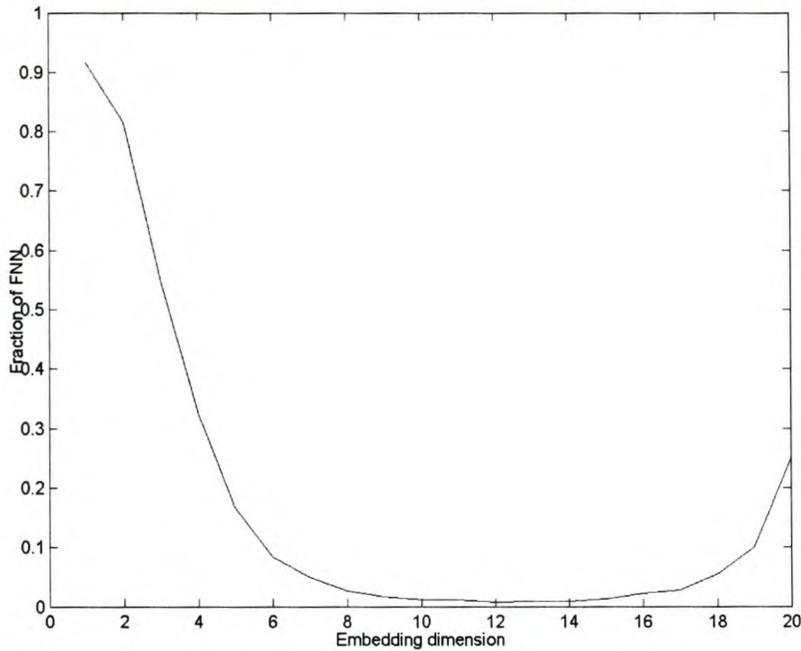
As with the previous statistics, the variables analysed can be divided into two distinct groups. The first group consists of the FAG mill power ( $x_1$ ), the FAG mill load ( $x_2$ ), the FAG mill coarse feed rate ( $x_3$ ), the FAG mill inlet water flow rate ( $x_7$ ), the FAG mill discharge 2 flow rate ( $x_{10}$ ), the FAG mill rougher feed 1 flow rate ( $x_{11}$ ), the FAG mill rougher feed 2 flow rate ( $x_{12}$ ), and the FAG mill product size distribution ( $x_{14}$ ). For all of these variables the fraction of FNN fell to zero once the embedding dimension was increased sufficiently (figure 4.16). However, for the second group, consisting of the FAG mill fine feed rate ( $x_4$ ), the FAG mill discharge 1 flow rate ( $x_9$ ), and the FAG mill feed size distribution ( $x_{13}$ ), the fraction of FNN did not fall to zero

as the embedding dimension was increased (figure 4.17). This is indicative of extremely noisy data, resulting in the calculated embedding dimensions being unreliable. The FNN plots for the respective time series not shown below are included in the appendix.

It should also be noted that based on the LA function and AMI statistic analysis, it was found that most of the time series analysed contain a certain degree of noise. Since FNN for embedding dimension estimation is not robust in the presence of noisy data (Rhodes and Morari, 1997) the estimated embedding dimensions should be treated with caution.



**Figure 4.16: FNN plot of FAG mill power ( $x_1$ ) with  $k = 37$**



**Figure 4.17: FNN plot of FAG mill fine feed rate ( $x_4$ ) with  $k = 9$**

#### 4.3.4 Results discussion

From estimation of the above non-linear time series analysis statistics the following conclusions could be drawn:

1. The data in general is very noisy. This was especially the case for FAG mill fine feed rate ( $x_4$ ), the FAG mill discharge 1 flow rate ( $x_9$ ), and the FAG mill feed size distribution ( $x_{13}$ ) variables as was evident from results obtained from both the LA function and the FNN statistic.
2. Comparable results were obtained between the LA function and the AMI statistic for embedding delay estimation. This would be indicative of a reasonably linear system. Such a system may, however, still be chaotic and or stochastic.
3. Although the embedding delay estimation using the LA function and the AMI statistic gave comparable results, a high degree of uncertainty exists with the reliability of the LA function results. This is a direct result from the AMI statistic not being subject to linear/non-linear constraints, whereas the LA function is linear and thus only the AMI statistic results will be used for embedding purposes prior to modelling.

4. No significant distinction was found between the embedding dimensions calculated using FNN when using the embedding delay calculated by means of either the LA function or the AMI statistic. It thus seemed as if the choice of embedding delay did not carry a significant weight when using the FNN statistic to calculate the embedding dimension.
5. Owing to the various inadequacies of the advanced time series analysis statistics used and the discrepancies found within the results obtained, the importance of incorporating process knowledge into time series analysis is emphasized.

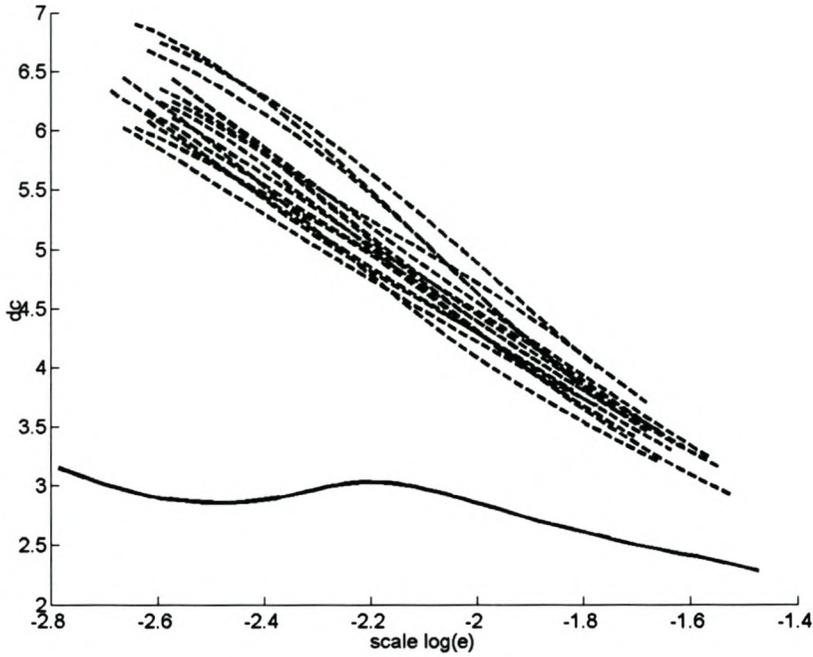
#### **4.4 Data classification**

Using the above estimated embedding parameters it is now possible to evaluate the non-linear determinism of the data by comparing real data characteristics or behaviours with those of a set of surrogate data, having the same statistical properties as the original data but generated under a null hypothesis. The null hypothesis states that the data originates from a Gaussian linear stochastic stationary process and the surrogate data is generated using a randomisation of the phases of the Fourier spectrum of the real data (Theiler et al., 1992).

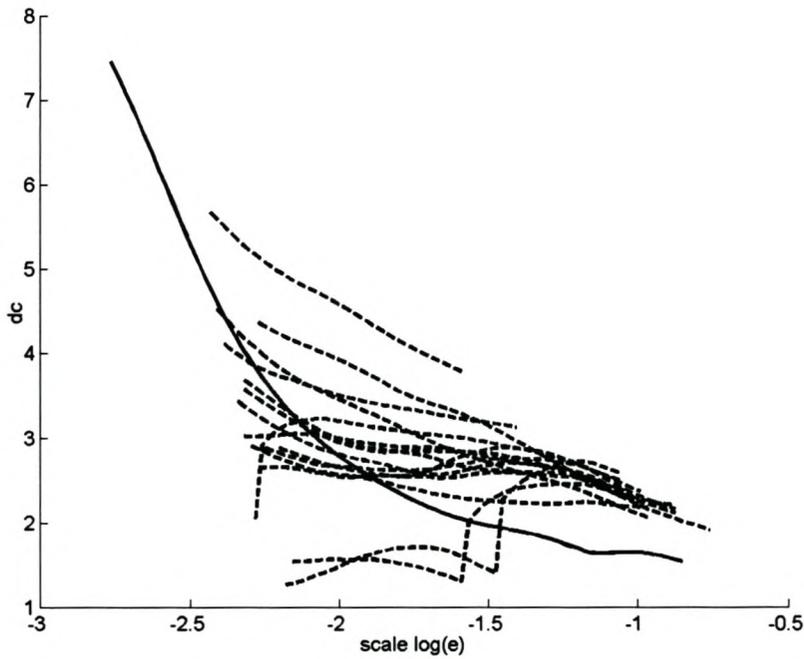
For data classification the correlation dimension for each the time series were calculated, as well as for the 15 surrogate data sets associated with each time series. From the results it is evident that the data could again be divided into two distinct groups. Group one, containing the FAG mill power ( $x_1$ ), the FAG mill load ( $x_2$ ), the FAG mill inlet water flow rate ( $x_7$ ), the FAG mill rougher feed 2 flow rate ( $x_{12}$ ), and the FAG mill product size distribution ( $x_{14}$ ), shows data containing deterministic character (figure 4.18), whereas group two, containing the FAG mill coarse feed rate ( $x_3$ ), the FAG mill fine feed rate ( $x_4$ ), the FAG mill discharge 2 flow rate ( $x_{10}$ ), and the FAG mill feed size distribution ( $x_{13}$ ), shows data containing stochastic character (figure 4.19). The correlation dimension curve plots for the respective time series not shown below are included in the appendix. Due to the short running times for the FAG mill discharge 1 flow rate ( $x_9$ ) and the FAG mill rougher feed 1 flow rate ( $x_{11}$ ) no correlation dimension curves or surrogate data sets could be generated for these variables.

From figure 4.18 it is evident that the correlation dimension curve of the system data is outside the cluster of curves corresponding to the random surrogate data. The magnitude of the displacement between the correlation dimension curves of the surrogate data and that of the observed data is an indication of the difference in complexity between the random surrogates and the dynamic system data (Barnard, 1999). From group one, there is a significant displacement between the correlation dimension curves of the surrogate data and that of the observed data for the FAG mill power ( $x_1$ ), the FAG mill inlet water flow rate ( $x_7$ ), and the FAG mill rougher feed 2 flow rate ( $x_{12}$ ), indicating that the observed data is of much less complexity than the surrogate data and therefore highly deterministic. A much smaller displacement is observed between the correlation dimension curves of the surrogate data and that of the observed data for the FAG mill load ( $x_2$ ) and the FAG mill product size distribution ( $x_{14}$ ), indicating that the observed data is of almost similar complexity to that of the surrogate data. Although the observed data for these time series appear random, the correlation dimension values for the observed data is consistently lower than that of the surrogate data, indicating that some inherent dynamics do in fact exist. This means that the observed data can still be classified as deterministic, though not convincingly so, and that it could prove difficult to define and estimate dynamic models for these systems.

For the time series contained in group two, very little, if any, displacement was observed between the correlation dimension curves of the surrogate data and that of the observed data; the correlation dimension curve for the observed data being located mostly between the correlation dimension curves for the surrogate data (figure 4.19). This stochastic character of the data is indicative of the difficulty that will most likely be experienced during the development of dynamic models for these systems, if possible at all.



**Figure 4.18: Correlation dimension curve for FAG mill power ( $x_1$ ; solid line) and its corresponding surrogate data sets (broken lines) with  $k = 37$  and  $m = 8$**



**Figure 4.19: Correlation dimension curve for FAG mill fine feed rate ( $x_4$ ; solid line) and its corresponding surrogate data sets (broken lines) with  $k = 9$  and  $m = 12$**

It should at this stage be noted that similar correlation dimension curves were obtained when analysing the data embedded using either the AMI calculated embedding parameters or the LA calculated embedding parameters. It was observed that in most cases lower correlation dimension estimates were obtained for the surrogate data and the observed data when using the LA calculated embedding parameters. The displacement between the surrogate data and the observed data was also marginally less than when using the AMI calculated embedding parameters, leading one to believe that the correlation dimension algorithm was exposed to a less developed attractor when using the embedding delay estimated by the LA function.

#### **4.5 Summary**

From the initial data inspection it became apparent that the data from the Process, Wipfrag and Malvern systems needed to be handled separately. Furthermore, owing to dynamic modelling requirements and problems experienced with numerous discontinuities in the datasets, subsets were constructed from the largest available continuous time series for each variable within the dataset. From these, possible correlations were determined between the different variables, such as the FAG mill power drawn ( $x_1$ ) and the FAG mill load ( $x_2$ ), and instrumentation problems, such as the FAG mill fine feed rate ( $x_4$ ) and the FAG mill feed rate ( $x_6$ ) values often being saturated at the top of their respective instrument measurement ranges, identified. This initial data inspection also led to the rejection of the FAG mill total screen feed rate ( $x_5$ ), the FAG mill feed rate ( $x_6$ ), and the FAG mill motor reference speed ( $x_8$ ) from further analysis.

Since all of the Fourier power spectra drawn for the time series analysed were broadband, but none containing pronounced periodicities, no conclusions could be drawn as to whether the time series were chaotic or random. It was also found that although a part of a time series may be chaotic, it couldn't be concluded that the whole time series is then chaotic.

From the calculation of the LA function and the FNN statistic it could clearly be seen that the data in general is very noisy. However, due to the high degree of uncertainty that exists with the reliability of the LA function results, and the fact that no significant difference was noted between the embedding dimensions determined using

the embedding delay calculated by means of either the LA function or the AMI statistic, it was decided that only the AMI statistic results would be used for embedding purposes prior to modelling. From the calculation of the various advanced time series analysis statistics the importance of incorporating process knowledge into the interpretation of the time series analysis results also became evident.

Using the method of surrogate data, all the time series were classified as either exhibiting deterministic or stochastic character. The time series contained in the group showing deterministic character was further split into those time series having a significant displacement between the correlation dimension curves of the surrogate data and that of the observed data, indicating highly deterministic data, and those with a much smaller displacement between the correlation dimension curves of the surrogate data and that of the observed data, indicating towards slight randomness in the data.

## 5 Modelling results

---

Depending on the level of knowledge and the physical insight into a system, all models derived from a system can be classified as either white box, grey box, or black box models (Reuter and Yang, 2001). Whereas white box models can be constructed entirely from existing knowledge and physical insight, for black box models no physical insight is used or needed; grey box models being a mix of white and black box models. Owing to the nature of the data and the process under investigation, the focus of this thesis is on black box dynamic modelling using neural networks.

Based on the time series analysis methodology defined in chapter 3, the following procedure will be followed for modelling of the process data:

- Embed the respective time series using either the MOD or SSA.
- Reconstruct the dynamic attractor.
- Estimate the upper prediction horizon for the variable to be predicted using Lyapunov exponents.
- Model the data using one of the non-linear empirical modelling techniques discussed in chapter 3.
- Evaluate the models using graphical residual analysis and the method of surrogate data.

As stated in chapter 4, this methodology will be applied separately to the data from the Process, Wipfrag and Malvern systems.

### 5.1 Process dataset

As stated in chapter 2, the FAG mill data contained in the Process dataset can be classified as input and output variables as shown in table 5.1.

**Table 5.1: FAG mill variable classification**

Input	Manipulated	Inlet water flow rate ( $x_7$ )
		Coarse ore feed rate ( $x_3$ )
		Fine ore feed rate ( $x_4$ )
Output	Controlled	Power ( $x_1$ )
	Measured	Load ( $x_2$ )
		Discharge flow rate ( $x_9, x_{10}$ )

From these available variables, only the FAG mill power ( $x_1$ ) is classified as a controlled variable and will therefore be the focus of the dynamic modelling. Another contributing factor to the use of the FAG mill power ( $x_1$ ) as the predictor is due to the strong relationship that exists between the energy required for particle size reduction and product particle size relative to the feed particle size; FAG mill power ( $x_1$ ) being the obvious predictor in the absence of any proper particle size measurement. Although not listed as a variable directly related to the FAG mill, the FAG mill rougher feed flow rates ( $x_{11}$  and  $x_{12}$ ) are also available in the Process dataset. Since a stable, continuous feed to the flotation section is very important for optimum mineral extraction, the predictability of these variables will also be investigated.

For the embedding of the data, various embedding strategies were followed making use of embedding parameters estimated using either MOD (AMI was used for embedding delay estimation and FNN for embedding dimension estimation) or SSA. For SSA the maximum embedding dimension was arbitrarily chosen as 20. This resulted from higher embedding dimensions leading to more complex models, with larger input spaces, without improving the performance of the models. These embedding parameters are listed in table 5.2.

**Table 5.2: Process dataset embedding parameters**

Variable	MOD		SSA	
	Embedding Delay	Embedding Dimension	Embedding Delay	Embedding Dimension
$x_1$	37	8	1	20
$x_2$	106	6	1	20
$x_3$	240	2	1	20
$x_4$	9	12	1	16
$x_7$	42	8	1	20
$x_9$	1	5	1	5
$x_{10}$	3	15	1	4
$x_{11}$	3	9	1	5
$x_{12}$	3	11	1	5

Apart from the last 12.5% of data contained in the Process dataset, independent test datasets were also selected for model validation purposes. Details pertaining to the nature of these test datasets are given in the appendices.

### 5.1.1 Linear ARX model

One of the most often-used model structures is the simple linear difference equation relating the current model output to a finite number of past outputs and inputs. When applying this modelling technique, all input and output variables (listed in table 5.1) were used as model inputs with the aim of predicting the FAG mill power ( $x_1$ ). The ARX model was configured to use both input and output variables at time  $t - 1$  to predict the FAG mill power ( $x_1$ ) at time  $t$  and least squares minimisation was used for model optimisation:

$$x_1(t) + a_1 x_1(t-1) = \sum_{i=2,3,4,7,9,10} b_i x_i(t-1) \quad (5.1)$$

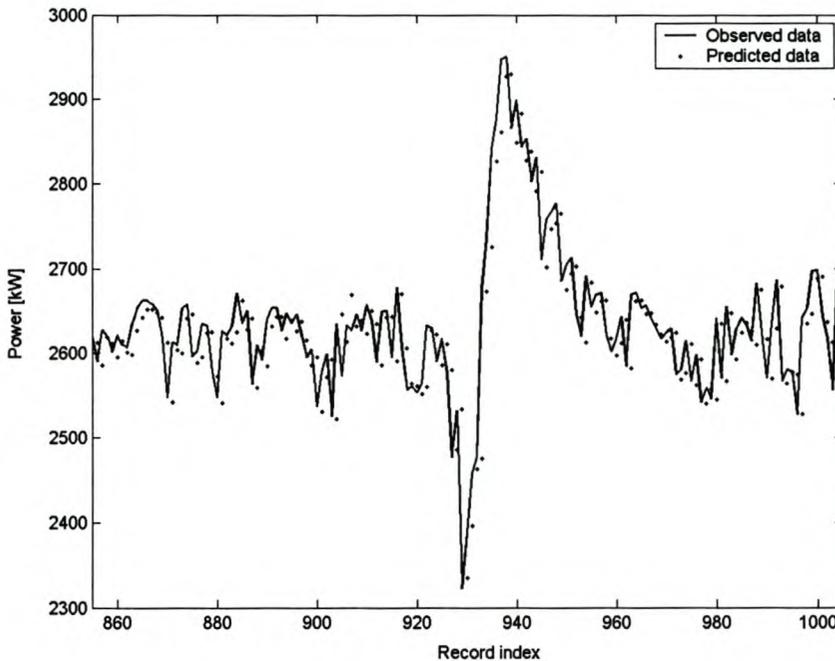
#### 5.1.1.1 Model validation

From the modelling results (table 5.3) it can be seen that the *linearARX* model performed very well when considering the results obtained from one-step prediction on the validation and test datasets.

**Table 5.3: Summary of  $R^2$  [%] statistics for validation and independent test datasets**

Test set	Size	<i>linearARX</i>
Validation data	1407	84.24
Test dataset 1	6571	94.21
Test dataset 2	6091	96.22
Test dataset 3	4501	97.96

However, when examining the one-step prediction graph obtained from applying the *linearARX* model to the validation data (figure 5.1), it seems as if the predicted values for *linearARX* are offset to the right of the actual data. This leads one to suspect that either the model does not generalise sufficiently or it requires more model inputs. It would also seem as if a high  $R^2$  value alone does not guarantee that the model fits the data well, leading to the use of graphical residual analysis as an additional tool in determining the quality of the model.

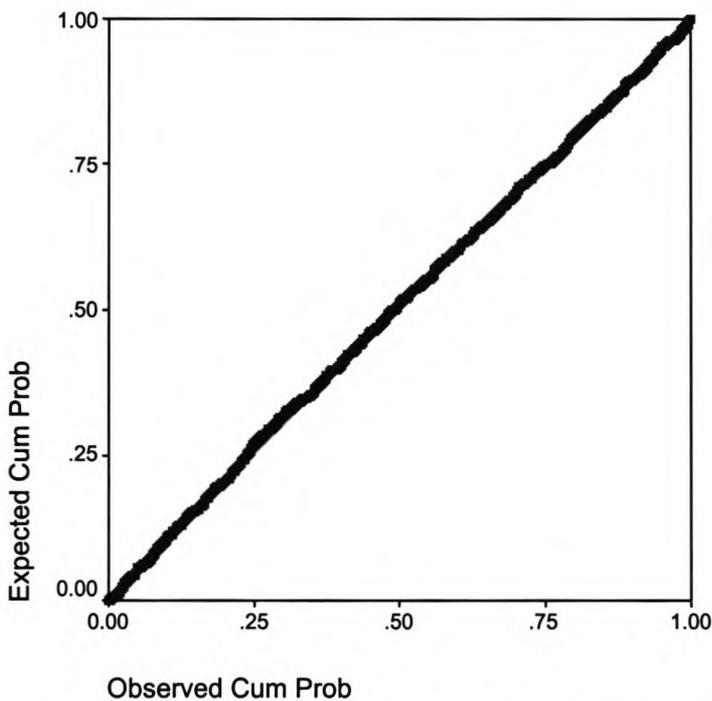
**Figure 5.1: One-step prediction of FAG mill power ( $x_1$ ) using *linearARX* ( $R^2 = 84.24\%$ )**

#### 5.1.1.2 Use of graphical residual analysis for model validation

If one assumes that the model fit to the data is correct, the residuals approximate the random errors that make the relationship between the explanatory variables and the

response variable a statistical relationship; if the residuals appear to behave randomly, it suggests that the model fits the data well. However, if non-random structure is evident in the residuals, it is a clear sign that the model fits the data poorly.

For the graphical residual analysis of the *linearARX* model, the normal P-P plot of the residuals obtained from the application of *linearARX* to one-step prediction (figure 5.2) was constructed. From figure 5.2 it is evident that the points tend towards clustering around a straight line indicating that the sample is from a normal distribution. Since it seems as if the residuals obtained from the use of the *linearARX* model for one-step prediction is normally distributed, it can be said that very little determinism is left in the data to be extracted by the model. This in turn leaves little room for model optimisation using the above applied technique.



**Figure 5.2: Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using *linearARX***

#### 5.1.1.3 Results discussion

From the above it can be seen that given the Process dataset, a model with a very high  $R^2$  can be obtained for the FAG mill power ( $x_1$ ) using a linear ARX model. However,

the one-step predictions obtained from applying the *linearARX* model to unseen data resulted in predictions that seemed to be offset to the right of the actual data. Using residual analysis it was, however, shown that the model generalises well and was successful at capturing the underlying dynamics of the data. From the one-step prediction plot it would seem as if more model inputs are required; the relationship between these inputs and the FAG mill power ( $x_1$ ) possibly being non-linear requiring the use of non-linear modelling techniques.

### **5.1.2 One-dimensional time series modelling of the FAG mill power using a MLP neural network**

Nichols and Nichols (2001) states that the main objective of attractor reconstruction is the use of a single system output in creating a set of new coordinates which preserves the invariant properties of the system. It should therefore in theory be possible to reconstruct the FAG mill systems attractor using only the FAG mill power ( $x_1$ ) and furthermore successfully predict the future values of the FAG mill power ( $x_1$ ), using this reconstruction.

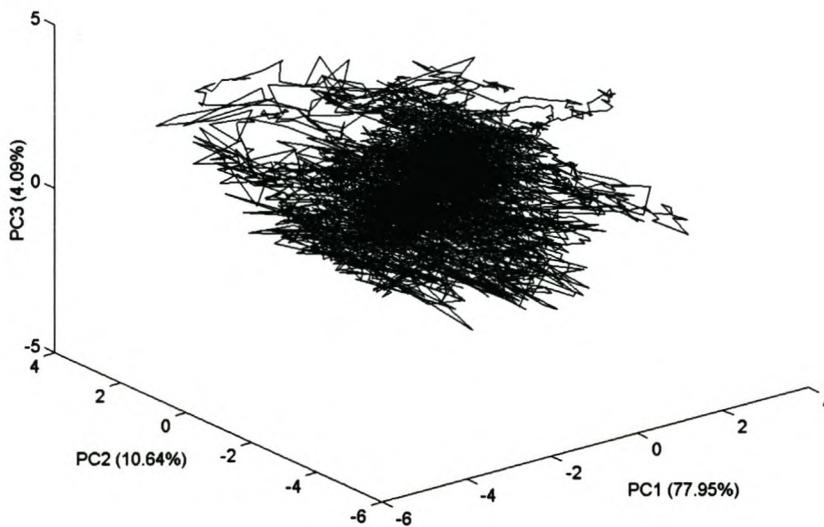
It has already been shown that the FAG mill power ( $x_1$ ) time series contains little, if any, noise, and shows highly deterministic character making it ideal for attractor reconstruction and dynamic modelling purposes. Based on this it was decided to first investigate the use of one-dimensional embedding, using only the FAG mill power ( $x_1$ ) time series, for attractor reconstruction and dynamic modelling.

#### **5.1.2.1 Attractor reconstruction**

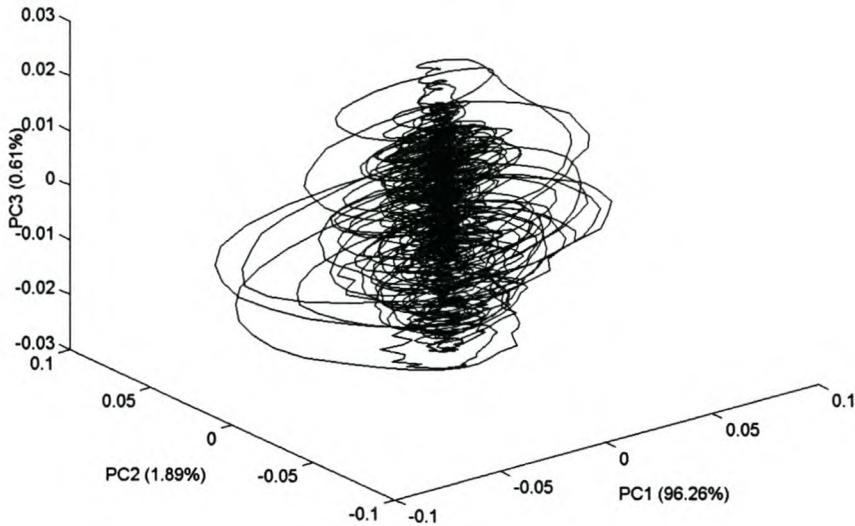
The FAG mill power ( $x_1$ ) time series was subsequently embedded, first using the MOD embedding parameters and secondly using the SSA embedding parameters. The respective reconstructed attractors, together with the variance explained by each of the shown principal components, are shown in figures 5.3 and 5.4.

From figure 5.3 it can be seen that the attractor reconstructed using the MOD embedding parameters is complex when compared to the reconstruction using the SSA embedding parameters (figure 5.4). A complex attractor is indicative of a time series containing dynamic and measurement noise leading one to believe that the little noise contained in the FAG mill power ( $x_1$ ) time series has a marked effect on the

attractor reconstruction if not removed from the data. This is confirmed when looking at the reconstructed attractor using the SSA embedding parameters, where noise is implicitly removed from the data during attractor reconstruction. The attractor in figure 5.4 also shows a definite pattern, occupying a limited region in the state space, when compared to figure 5.3, confirming the determinism found in the data using the method of surrogate data. Based on this, it is expected that the model constructed using the SSA embedded data will perform better than the model constructed using the MOD embedded data.



**Figure 5.3: Reconstructed attractor of FAG mill power ( $x_1$ ) based on MOD embedding parameters**



**Figure 5.4: Reconstructed attractor of FAG mill power ( $x_1$ ) based on SSA embedding parameters**

#### 5.1.2.2 Model structure

A MLP neural network was constructed for each of the respective embeddings methods used with the purpose of predicting only one-step ahead in time. For the MOD embedded data the MLP neural network, *MLPMOD\_0*, consisted of an input layer of 5 nodes, a hidden layer with 15 nodes and a single linear output node. For the SSA embedded data the MLP neural network, *MLPSSA\_0*, consisted of an input layer of 3 nodes, a hidden layer with 7 nodes and a single linear output node. Both networks were trained using the Levenberg-Marquardt algorithm and the number of nodes in the hidden layer estimated using the Schwartz Information Criterion (Schwartz, 1978). A summary of the modelling parameters is given in table 5.4.

**Table 5.4: One-dimensional time series modelling parameters**

	<i>MLP MOD 0</i>	<i>MLP SSA 0</i>
Embedding type	MOD	SSA
Number of principal components selected	5	3
Variance explained [%]	96.14	98.76
Number of hidden nodes	15	7

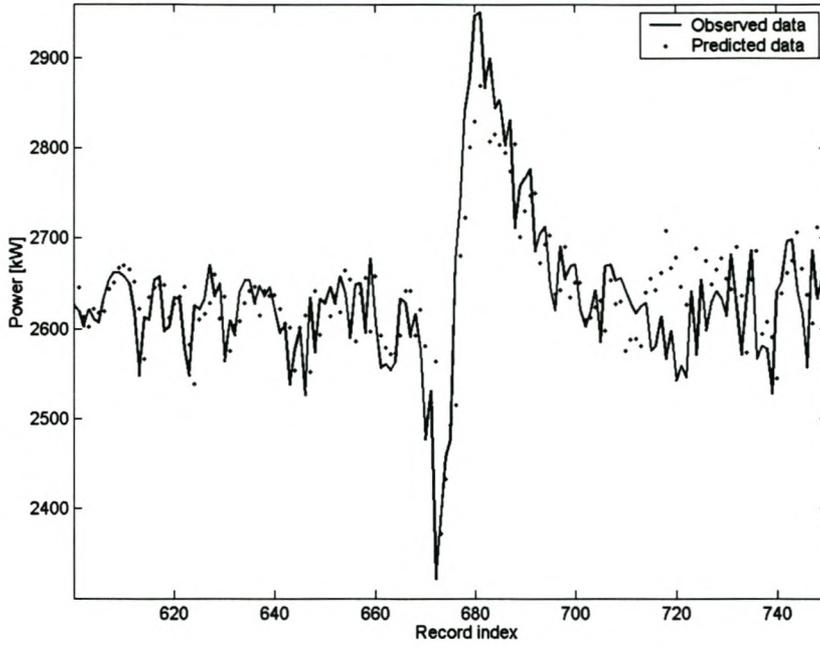
### 5.1.2.3 Model validation

From the modelling results (table 5.5) it can be seen that the *MLPSSA\_0* model not only outperformed the *MLPMOD\_0* model when considering the validation data, but also for two of the three test datasets used.

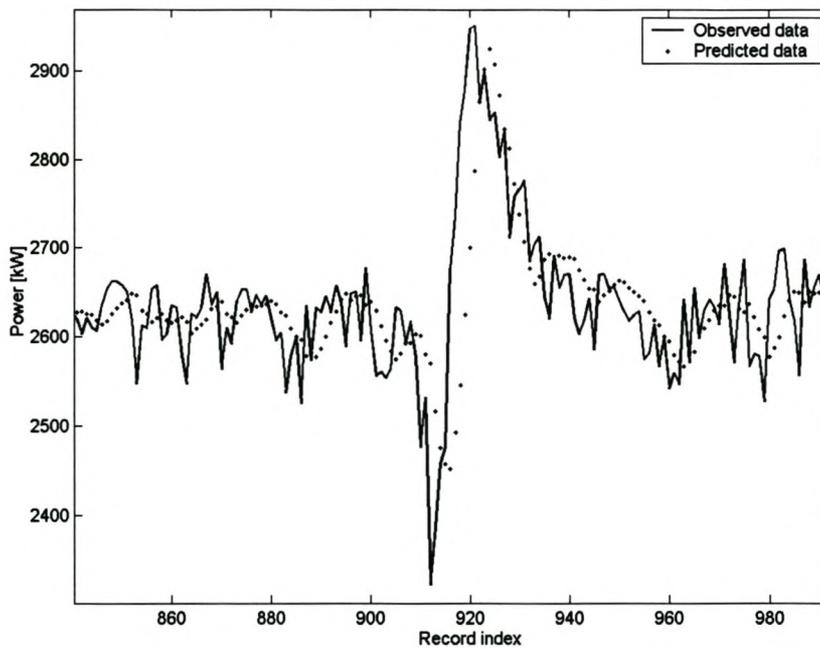
**Table 5.5: Summary of  $R^2$  [%] statistics for validation and independent test datasets**

Test set	Size	<i>MLP MOD_0</i>	<i>MLP SSA_0</i>
Validation data	1407	43.45	82.21
Test dataset 1	6571	92.10	94.10
Test dataset 2	6091	93.80	96.60
Test dataset 3	4501	95.10	94.70

Comparing the one-step prediction graphs obtained from *MLPMOD\_0* (figure 5.5) and *MLPSSA\_0* (figure 5.6) on the validation data, it can be seen that although the  $R^2$  for these two figures differ considerably, prediction in both cases appear not to be very good; both models also being outperformed by the *linearARX* model. It is also evident that most of the *MLPSSA\_0* predicted values (figure 5.6) are offset to the right of the actual data leading one to suspect that either the model does not generalise sufficiently or it requires more model inputs.



**Figure 5.5: One-step prediction of FAG mill power ( $x_1$ ) using *MLPMOD\_0* ( $R^2 = 43.45\%$ )**

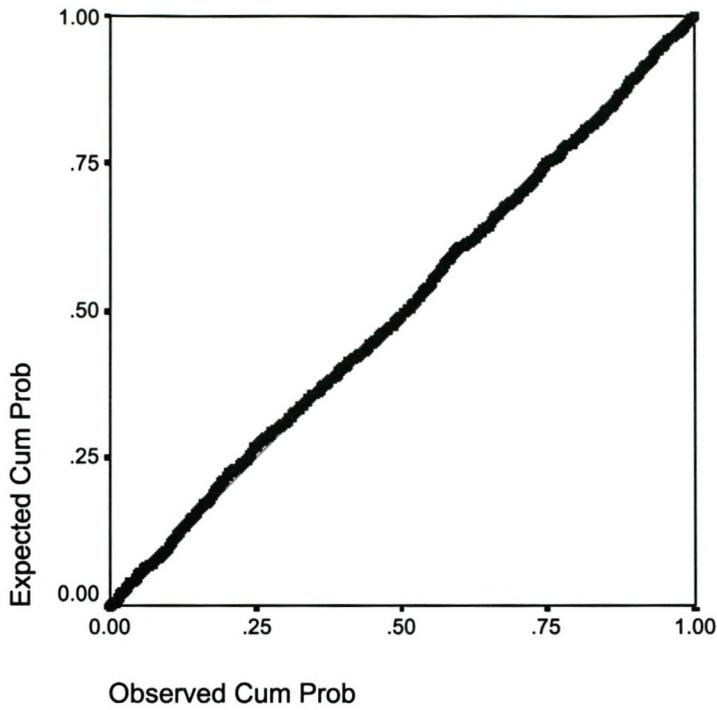


**Figure 5.6: One-step prediction of FAG mill power ( $x_1$ ) using *MLPSSA\_0* ( $R^2 = 82.21\%$ )**

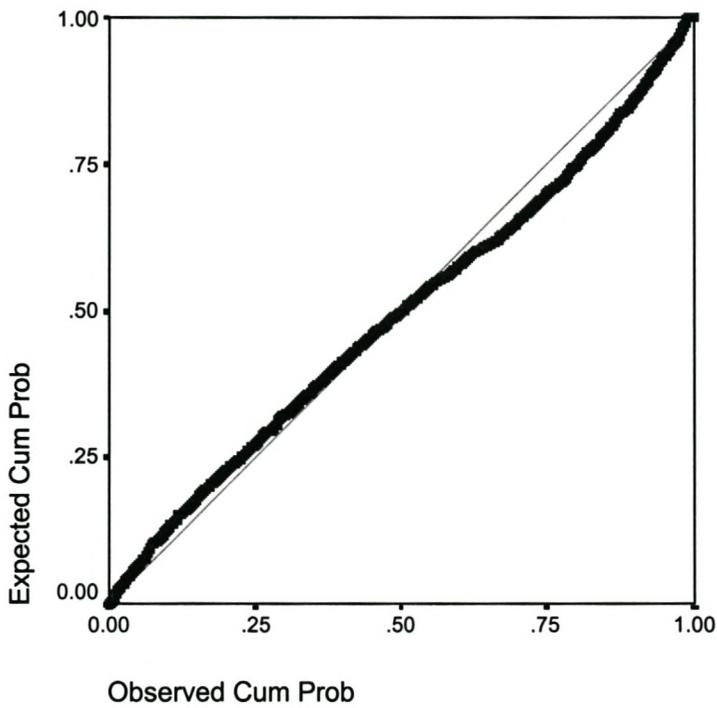
#### 5.1.2.4 Use of graphical residual analysis for model validation

For the graphical residual analysis of the models, the normal P-P plot of the residuals obtained from the application of *MLPMOD\_0* (figure 5.7) and *MLPSSA\_0* (figure 5.8) to one-step prediction was constructed. Here the cumulative proportion for a single numeric variable is plotted against the cumulative proportion expected if the sample were from a normal distribution. From the normal P-P plot of the residuals obtained using *MLPMOD\_0* (figure 5.7) it can be seen that the points tend towards clustering around a straight line, leading one to believe that the sample is from a normal distribution. In contrast to this, for the normal P-P plot of the residuals obtained using *MLPSSA\_0* (figure 5.8) the plot deviates somewhat from being a straight line indicating a non-normal distribution.

Since it seems that the residuals obtained from the use of *MLPMOD\_0* for one-step prediction are normally distributed, it can be said that very little determinism is left in the data to be extracted by the model. This means that the *MLPMOD\_0* model has extracted all available information from the data presented to it, leaving little room for model optimisation should the model input data be restricted to only the one-dimensional time series describing the FAG mill power ( $x_1$ ). Since the residuals obtained using *MLPSSA\_0* for one-step prediction seemed to be non-normally distributed the possibility exists of using these residuals as model inputs for further modelling to try and extract the determinism in the data not captured by *MLPSSA\_0*.



**Figure 5.7: Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using *MLPMOD\_0***



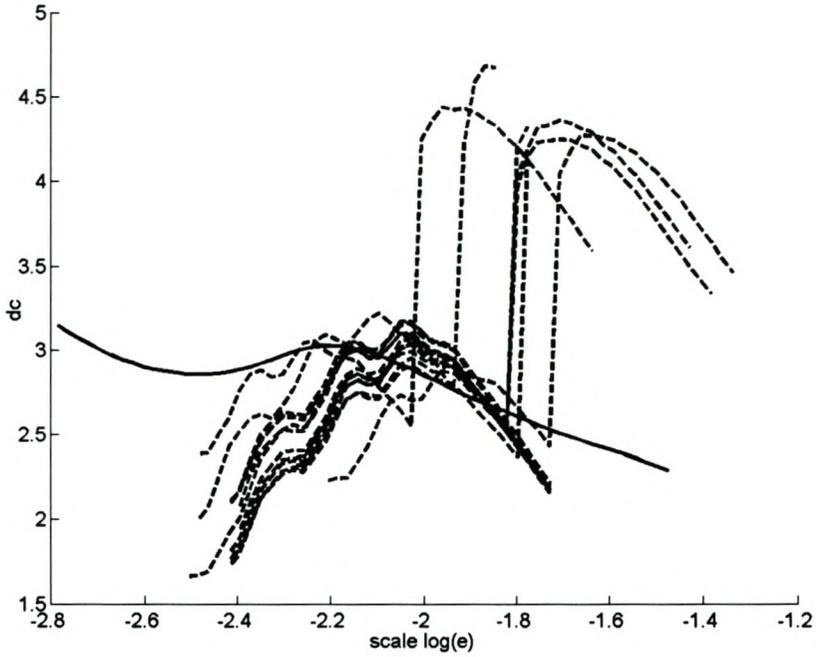
**Figure 5.8: Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using *MLPSSA\_0***

### 5.1.2.5 Use of surrogate data for model validation

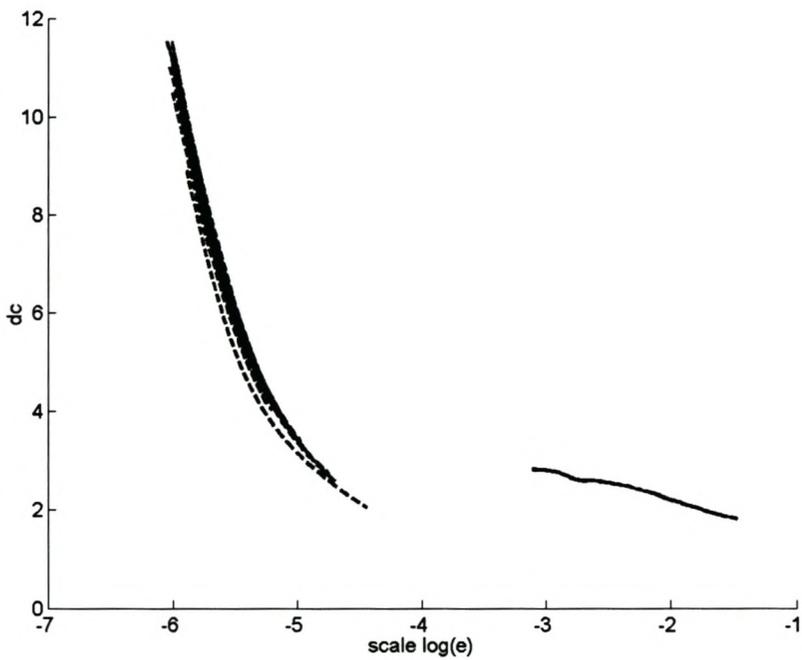
Following graphical residual analysis, both the *MLPMOD\_0* model and the *MLPSSA\_0* model were subjected to validation using the method of surrogate data. Similar to when surrogate data was used to evaluate the non-linear determinism of the data by comparing real data characteristics or behaviours with those of a set of surrogate data, the method of surrogate data is again used to determine the quality of the model by showing whether or not the model has completely captured the structure of the data.

From figure 5.9 it is clear that the *MLPMOD\_0* model has not completely captured the structure of the data, except in the medium-scale region of the dynamic attractor ( $-2.4 < \log \epsilon_0 < -1.75$ ). The peculiar dips in the surrogate curves (figure 5.9) for  $-2.05 < \log \epsilon_0 < -1.75$  can be ascribed to numerical instability in the correlation dimensional algorithm (Barnard, 1999). Compared to this, the actual data from *MLPSSA\_0* model is on average of a significantly lower overall dimension than the surrogate data derived from the same model with no overlap between the actual and surrogate data (figure 5.10). Further more, the actual data from *MLPSSA\_0* only covers the large-scale region ( $\log \epsilon_0 > -3.1$ ) of the dynamic attractor whereas the surrogate data derived from the same model only covers the small-scale region ( $\log \epsilon_0 < -4.5$ ) of the dynamic attractor. It would therefore seem that the surrogate data derived from *MLPSSA\_0* model is very noisy, resulting from an increase in the complexity of the microstructure in the attractor. This is evident from the higher correlation dimension obtained at the small scales caused by dynamic noise (Barnard, 1999), as opposed too a higher correlation dimension at all scales, indicating an overall more complex structure in the attractor.

These results obtained from applying the method of surrogate data to the constructed models were found to be in accordance with those obtained from the graphical residual analysis. Since the use of surrogate data for model validation is dependent on the models ability to be accurately used for free-run prediction, the surrogate data validation results should be analysed in conjunction with the free-run prediction results.



**Figure 5.9:** Correlation dimension curve for FAG mill power ( $x_1$ ; solid line) and of the non-linear surrogates of *MLPMOD\_0* (broken lines)



**Figure 5.10:** Correlation dimension curve for FAG mill power ( $x_1$ ; solid line) and of the non-linear surrogates of *MLPSSA\_0* (broken lines)

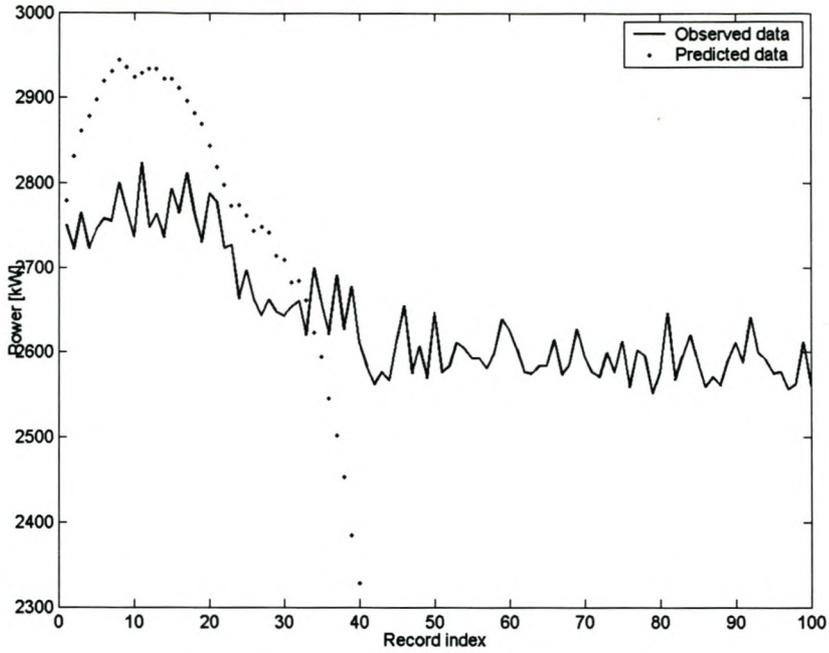
### 5.1.2.6 Free-run prediction

The Lyapunov spectrum for the FAG mill power ( $x_1$ ) was calculated, yielding insight into the predictability of the system, it being a measure of the dynamical stability of the system. Since the 1<sup>st</sup> Lyapunov exponent calculated using the MOD embedding parameters (table 5.6) is positive, the system can be classified as chaotic. However, the sum of all the Lyapunov exponents in the Lyapunov spectrum is negative, confirming that the attractor is bounded and therefore predictable. For the SSA embedding parameters similar conclusions could be drawn.

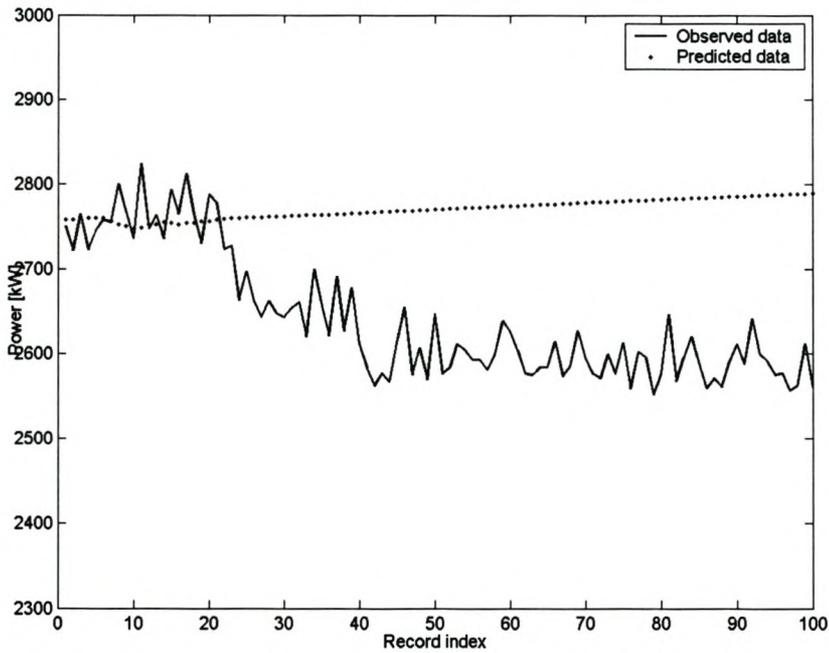
**Table 5.6: Lyapunov spectrum for the FAG mill power ( $x_1$ ) time series**

	<i>MOD embedding</i>	<i>SSA embedding</i>
1 <sup>st</sup> Lyapunov exponent	$3.30e^{-03}$	$5.81e^{-02}$
Prediction horizon [s]	8300	500

When considering the foregoing results, it seems highly unlikely that the model based on the MOD embedding parameters will perform better when applied for free-run prediction than the model based on SSA embedding parameters. However, from the free-run prediction plots (figures 5.11 and 5.12) the findings based on the Lyapunov spectrum was confirmed with the *MLPMOD\_0* model performing better than the *MLPSSA\_0* model, generally better following the initial trend in the data. In spite of this, for the free-run prediction the *MLPMOD\_0* model failed after only 30 predictions and the *MLPSSA\_0* failed after only 15 predictions, both considerably less than that estimated by the Lyapunov spectrum. This lack of ability of these models to be used for free-run prediction could be ascribed to the fact that the models are highly dependent on the initial conditions of the data and the possibility of the FAG mill power ( $x_1$ ) variable not containing all the information of the system being modelled.



**Figure 5.11: Free-run prediction of FAG mill power ( $x_1$ ) using *MLPMOD\_0***



**Figure 5.12: Free-run prediction of FAG mill power ( $x_1$ ) using *MLPSSA\_0***

### 5.1.2.7 Results discussion

From the above it can be concluded that owing to the implicit noise reduction of the SSA algorithm much better attractor reconstruction was obtained when using the SSA embedding when compared to the reconstructed attractor using the MOD embedding.

Furthermore, the model based on the MOD embedded data was also more complex than the model based on the SSA embedded data, *MLPMOD\_0* having 5 input nodes and 15 hidden nodes compared to only 3 input nodes and only 7 hidden nodes for *MLPSSA\_0*. It should be noted that although the expectation that the SSA model will outperform the MOD model was correct, both models extracted almost all possible information from the data without being able to capture the underlying dynamic structure of the data. This led to the conclusion that either:

- for the dynamic system under investigation, the observed component of the system, the FAG mill power ( $x_1$ ), not necessarily being a function of all the state variables of the system; or
- important information regarding the system is lost during the embedding procedure when applying a data reduction technique such as principal component analysis.

In order to further investigate the first possibility, the use of a multi-dimensional embedding (Barnard, 1999), to gather more information regarding the dynamic system, is explored. For the second possibility the use of LS-SVM's for predictive modelling is investigated.

### **5.1.3 Multi-dimensional time series modelling of the FAG mill power using a MLP neural network**

For the multi-dimensional time series modelling, five different embedding strategies for the measurements surrounding the FAG mill (table 5.7) were investigated. This was done in order to ensure that the best model was also the simplest model possible.

The five strategies are:

1. None of the variables were embedded. All input and output variables at time  $t$  is used to predict the FAG mill power ( $x_1$ ) at time  $t + 1$ . Model: *MLPnone*.
2. All variables were embedding using embedding parameters estimated using MOD. Model: *MLPMOD\_1*.
3. All variables were embedding using embedding parameters estimated using SSA. Model: *MLPSSA\_1*.

4. Only output variables were embedding using embedding parameters estimated using SSA. However, input variables were included in the multi-dimensional embedding with an embedding delay and embedding dimension of unity. Model: *MLPSSA\_2*.
5. Only input variables were embedding using embedding parameters estimated using SSA; these input variables also being some of the principal variables that can affect the control of an autogenous mill. However, output variables were included in the multi-dimensional embedding with an embedding delay and embedding dimension of unity. Model: *MLPSSA\_3*.

**Table 5.7: FAG mill variable classification**

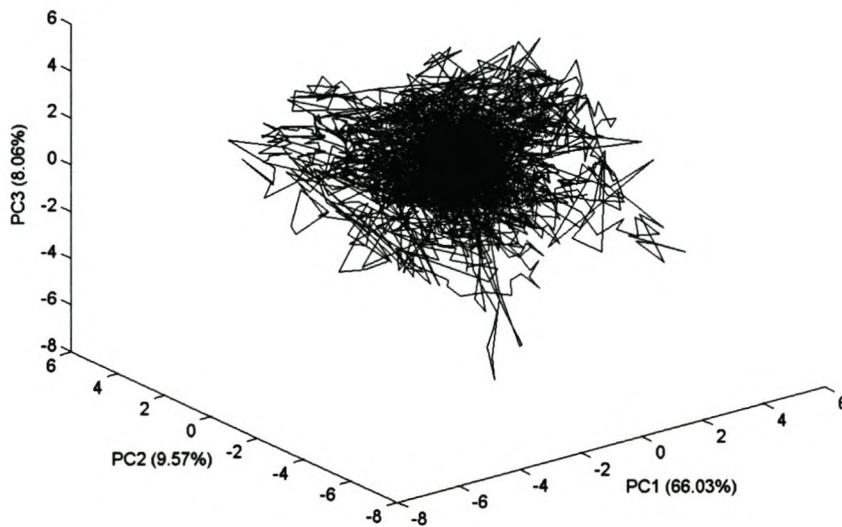
Input	Manipulated	Inlet water flow rate ( $x_7$ )
		Coarse ore feed rate ( $x_3$ )
		Fine ore feed rate ( $x_4$ )
Output	Controlled	Power ( $x_1$ )
	Measured	Load ( $x_2$ )
		Discharge flow rate ( $x_9, x_{10}$ )

#### 5.1.3.1 Attractor reconstruction

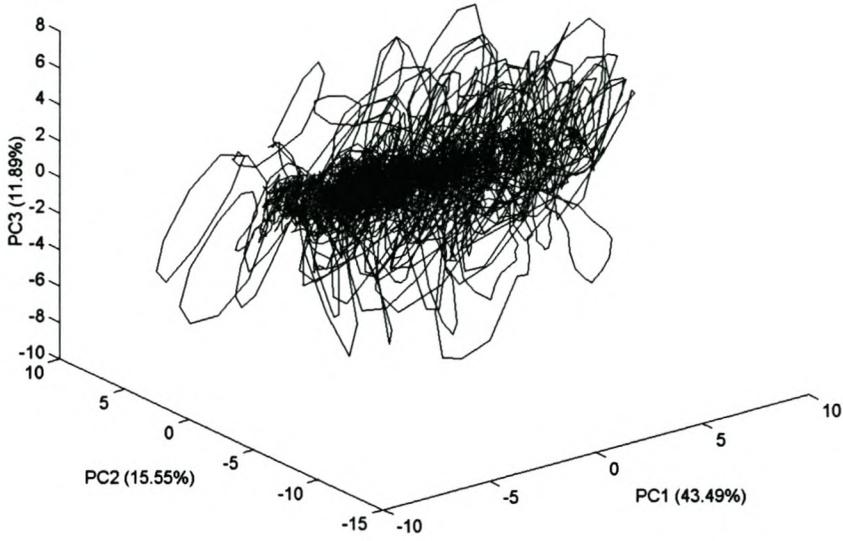
Each of the strategies making use of embedding was used for attractor reconstruction and all the strategies were used for model estimation, the aim being the construction of a model to be used for the successful prediction of future values of the FAG mill power ( $x_1$ ). The respective reconstructed attractors, together with the variance explained by each of the shown principal components, are shown in figures 5.13 through 5.16.

Similar to the reconstructed attractors from the embedding of the one-dimensional FAG mill power ( $x_1$ ) time series, the reconstructed attractor from the multi-dimensional embedding using the MOD embedding parameters (embedding strategy 2; figure 5.13) is again much more complex when compared to the reconstructed attractor from the multi-dimensional embeddings using the SSA embedding parameters (embedding strategies 3-5; figures 5.14-5.16). Although it might not seem like it, the reconstructed attractor based on embedding strategy 5 (figure 5.16) is, when rotated at the correct angle in state space, very similar to those based on embedding strategies 3 and 4 (figures 5.14 and 5.15).

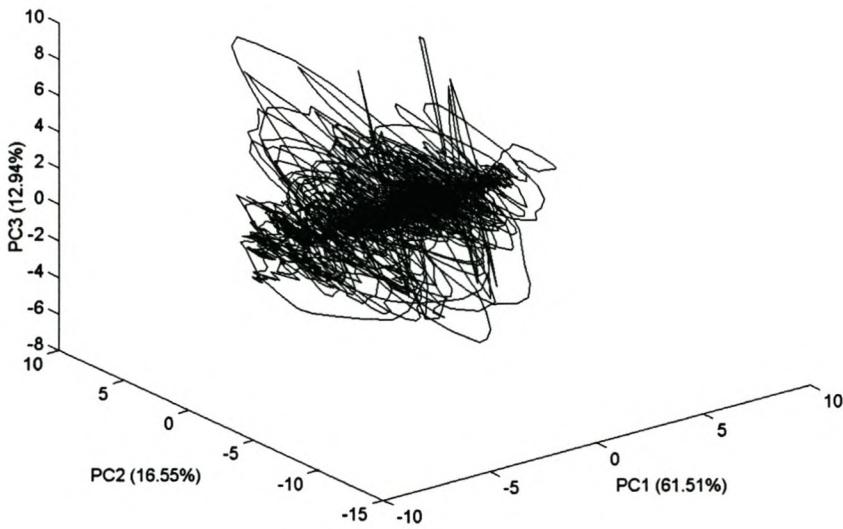
Following similar reasoning as before, it is again expected that the models constructed based on the SSA embedding strategies will perform better than the model constructed based on the MOD embedding strategy. It should also be noted at this point that the reconstructed attractor based on embedding strategy 5 covers the largest area in state space, making it the embedding strategy most likely to produce the best model, which is closely followed by the reconstructed attractor based on embedding strategy 3 and then the reconstructed attractor based on embedding strategy 4.



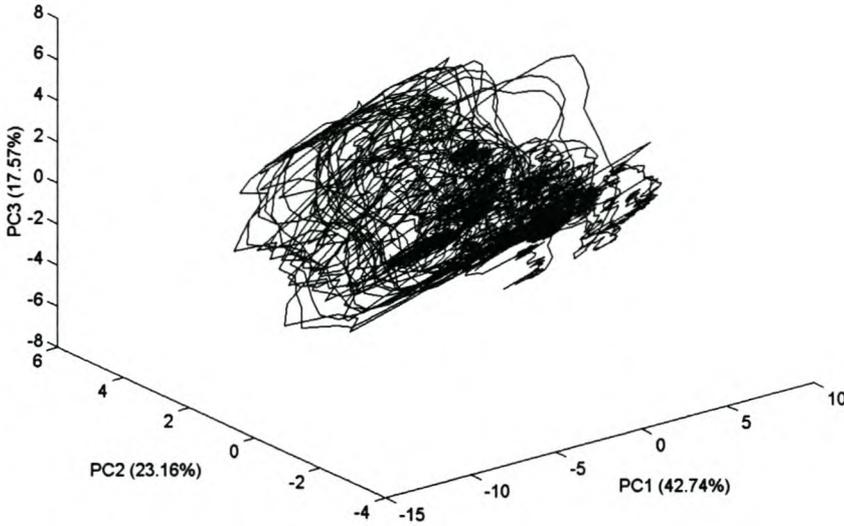
**Figure 5.13: Reconstructed attractor based on embedding strategy 2**



**Figure 5.14: Reconstructed attractor based on embedding strategy 3**



**Figure 5.15: Reconstructed attractor based on embedding strategy 4**



**Figure 5.16: Reconstructed attractor based on embedding strategy 5**

5.1.3.2 Model structure

A MLP neural network was constructed for each of the proposed embedding strategies with the purpose of predicting only one-step ahead in time. As before, all networks were trained using the Levenberg-Marquardt algorithm and the number of nodes in the hidden layers estimated using the Schwartz Information Criterion (Schwartz, 1978). A summary of the modelling parameters is given in table 5.8.

**Table 5.8: Multi-dimensional time series modelling parameters**

	<i>MLP none</i>	<i>MLP MOD 1</i>	<i>MLP SSA 1</i>	<i>MLP SSA 2</i>	<i>MLP SSA 3</i>
Embedding strategy	1	2	3	4	5
Number of PC's selected	-	13	16	11	15
Variance explained [%]	-	96.82	97.04	97.93	97.28
Number of hidden nodes	8	7	6	5	3

5.1.3.3 Model validation

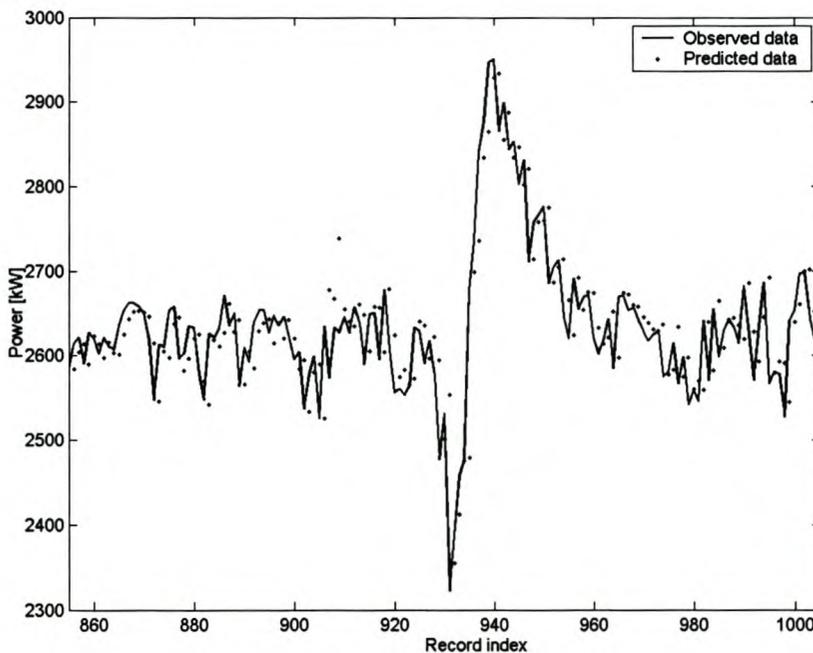
From the modelling results (table 5.9) it can be seen that the *MLPSSA\_3* model outperformed all other models constructed based on the multi-dimensional embedding strategies. Although the *MLPSSA\_3* model required 15 principal components, based on the Scree test (Cattell, 1966), to explain most of the variance in the data, the

ensuing model structure had the least number of nodes in the hidden layer when compared to the other models being tested, effectively resulting in the least complex model.

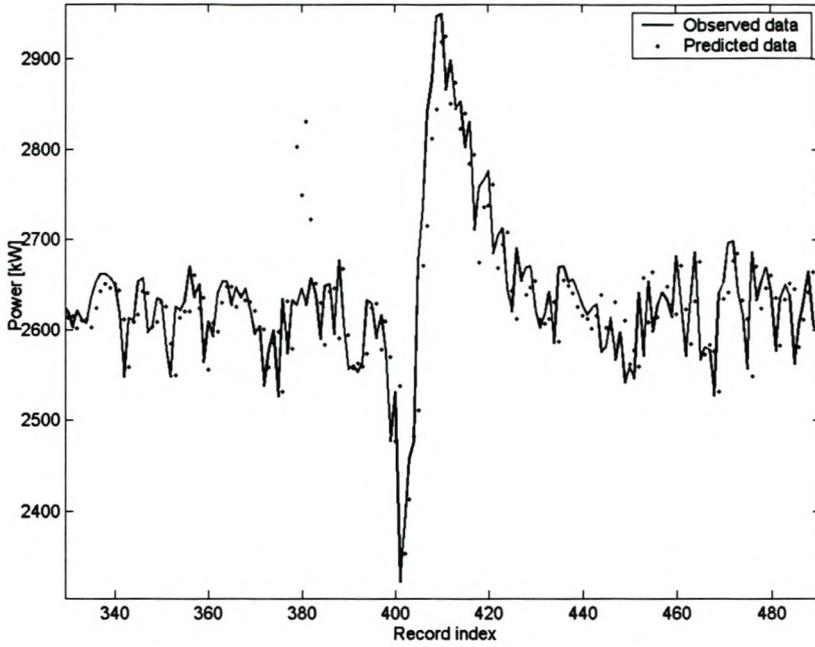
**Table 5.9: Summary of  $R^2$  [%] statistics for validation and independent test datasets**

Test set	Size	<i>MLP none</i>	<i>MLP MOD 1</i>	<i>MLP SSA 1</i>	<i>MLP SSA 2</i>	<i>MLP SSA 3</i>
Validation data	1407	81.94	37.72	80.03	71.26	82.34
Test dataset 1	6571	91.80	70.80	89.10	87.00	94.20
Test dataset 2	6091	96.20	53.30	93.90	94.60	96.40
Test dataset 3	4501	89.70	37.30	94.60	93.90	97.50

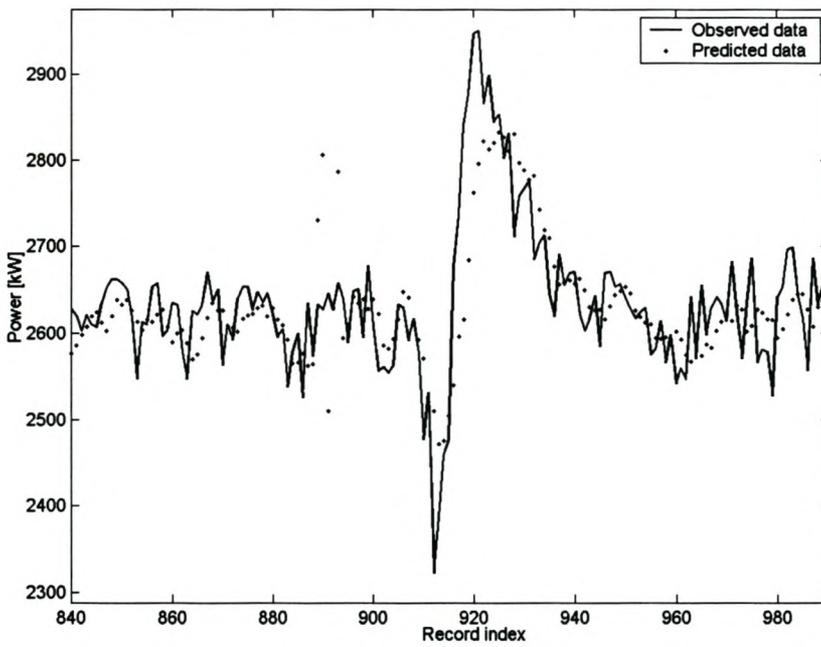
When comparing the one-step prediction graphs (figures 5.17 – 5.21) obtained from testing the various models with validation data, it can be visually confirmed that the *MLPSSA\_3* model is indeed the best of the tested models. However, as found previously, the predicted values for the *MLPSSA\_3* model (figure 5.21) are offset to the right of the actual data leading one to suspect that either the model does not generalise sufficiently or it requires more model inputs.



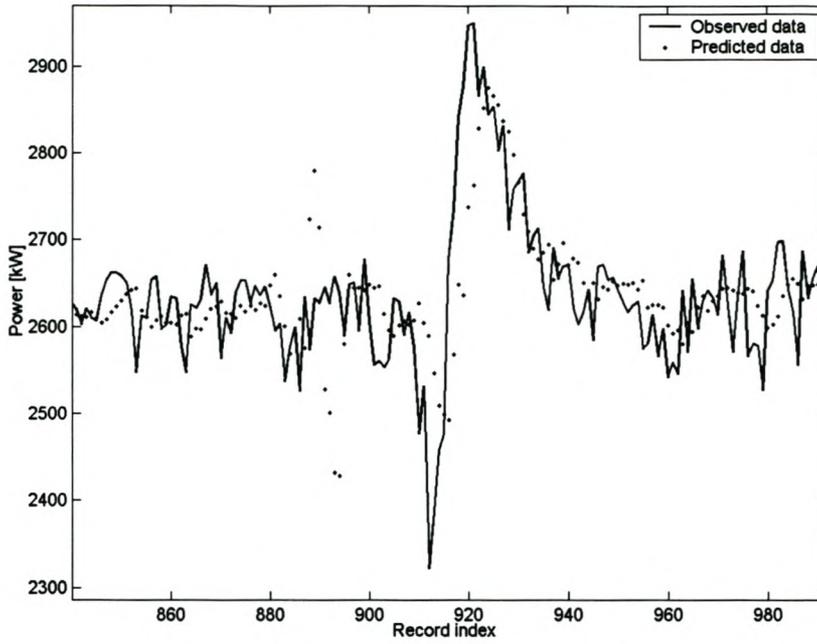
**Figure 5.17: One-step prediction of FAG mill power ( $x_1$ ) using *MLPnone* ( $R^2 = 81.94\%$ )**



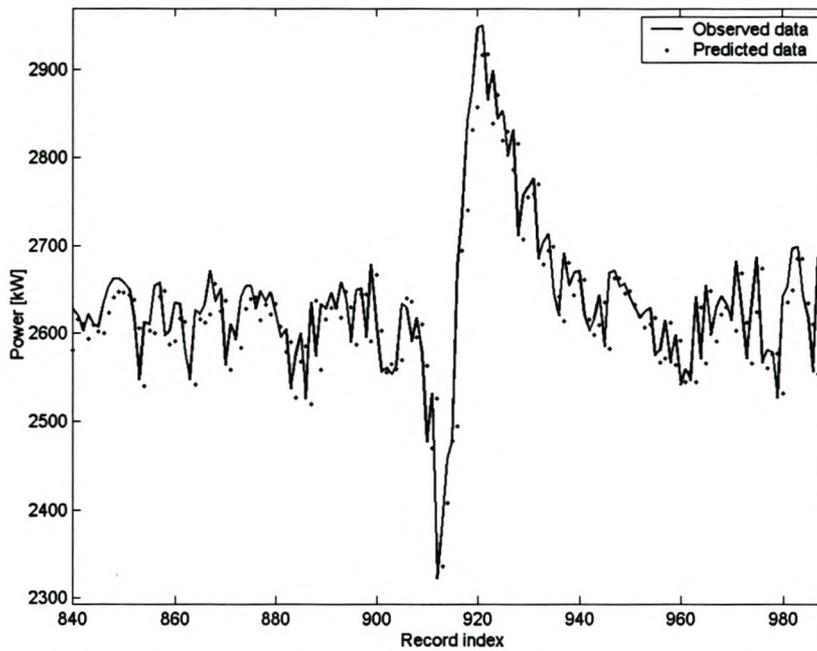
**Figure 5.18: One-step prediction of FAG mill power ( $x_1$ ) using *MLP*MOD\_1 ( $R^2 = 37.72\%$ )**



**Figure 5.19: One-step prediction of FAG mill power ( $x_1$ ) using *MLP*SSA\_1 ( $R^2 = 80.03\%$ )**



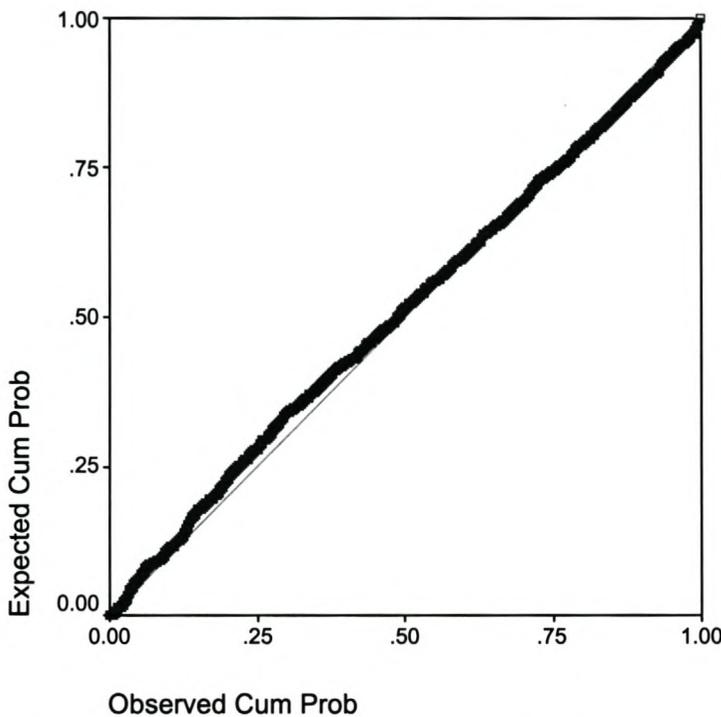
**Figure 5.20: One-step prediction of FAG mill power ( $x_1$ ) using *MLPSSA\_2* ( $R^2 = 71.26\%$ )**



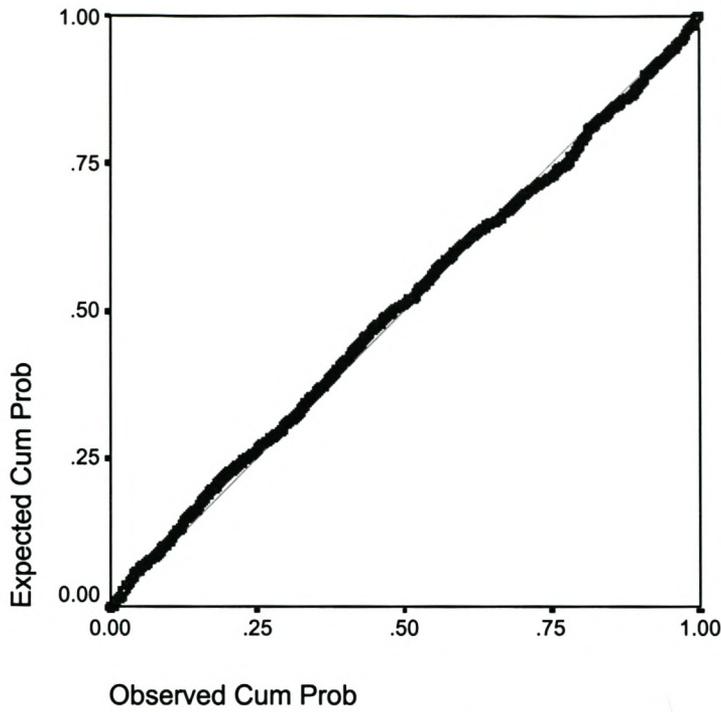
**Figure 5.21: One-step prediction of FAG mill power ( $x_1$ ) using *MLPSSA\_3* ( $R^2 = 82.34\%$ )**

#### 5.1.3.4 Use of graphical residual analysis for model validation

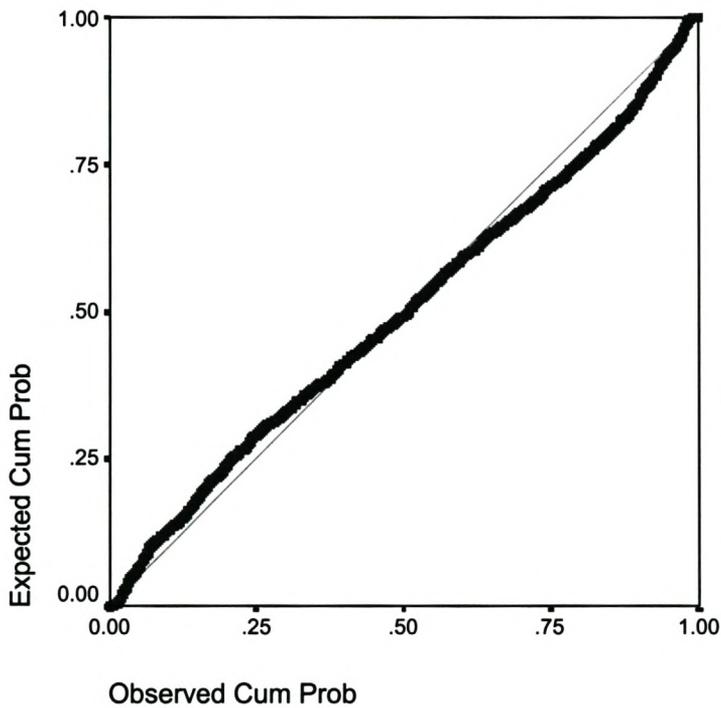
Normal P-P plots of the residuals obtained from the application of the four tested models to one-step prediction of the validation data are shown in figures 5.22 through 5.26. From these figures it is evident that only the residuals obtained from *MLPSSA\_3* is normally distributed, leading one to believe that very little determinism is left in the data to be extracted. From all the other models, specifically *MLPSSA\_2*, non-normally distributed residuals were obtained, which is indicative of available information within the data that was not extracted by the models.



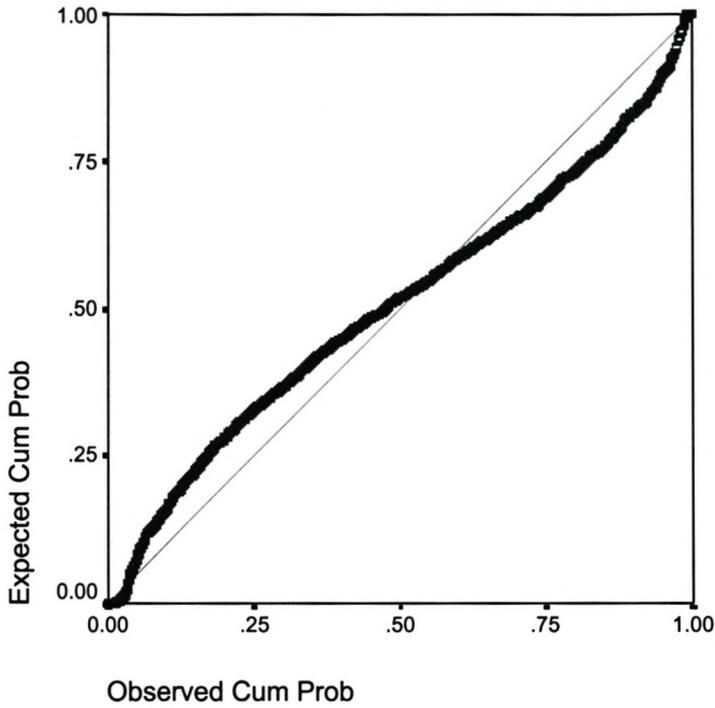
**Figure 5.22: Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using *MLPnone***



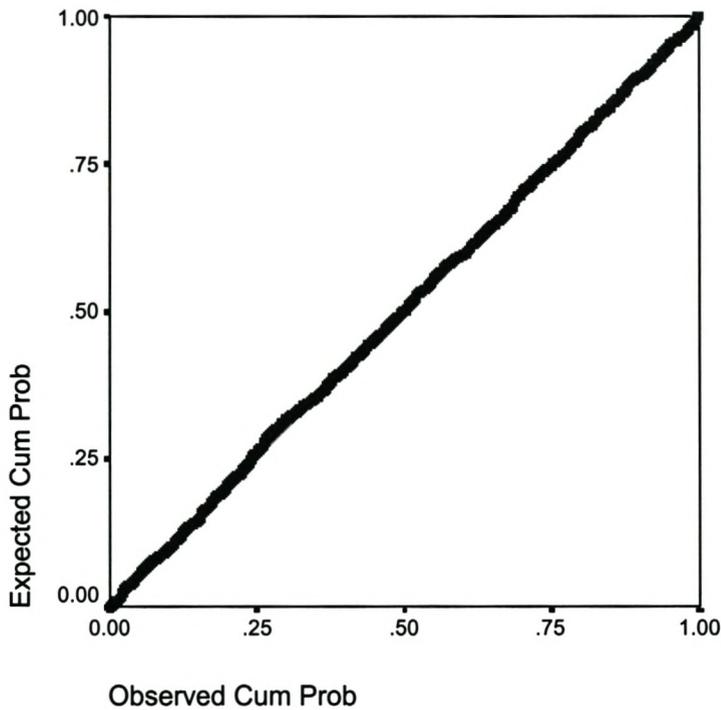
**Figure 5.23: Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using *MLPMOD\_1***



**Figure 5.24: Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using *MLPSSA\_1***



**Figure 5.25:** Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using *MLPSSA\_2*



**Figure 5.26:** Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using *MLPSSA\_3*

### 5.1.3.5 Use of surrogate data for model validation

Before discussing the results of this section it is important to note the following regarding the use of surrogate data for model validation when the model is based on multi-dimensional embedded time series. The method of surrogate data requires a free-run prediction of the model output variable. For a free-run prediction using a model based on a one-dimensional time series (only one variable) where the model inputs are also the values to be predicted, the first prediction,  $y_{(t+1)}$ , is made using the actual process data,  $(x_t, x_{(t-1)}, x_{(t-2)}, \dots, x_{(t-n)})$ , as model inputs. For the second prediction,  $y_{(t+2)}$ , both the actual process data and the previously predicted model output,  $(y_{(t+1)}, x_t, x_{(t-1)}, \dots, x_{(t-n+1)})$ , is used as model inputs. This procedure is continued; at some stage the model input will be comprised entirely of previously model output values. Now, when considering the use of a multi-dimensional time series (more than one variable) as model inputs, but the model only capable of predicting one of these variables as an output, a true free-run prediction of the selected variable is impossible. Since, as it was stated earlier, the method of surrogate data requires a true free-run prediction of the model output variable, this technique is not entirely suited to be used for the validation of the current models and the results should be treated with extreme care.

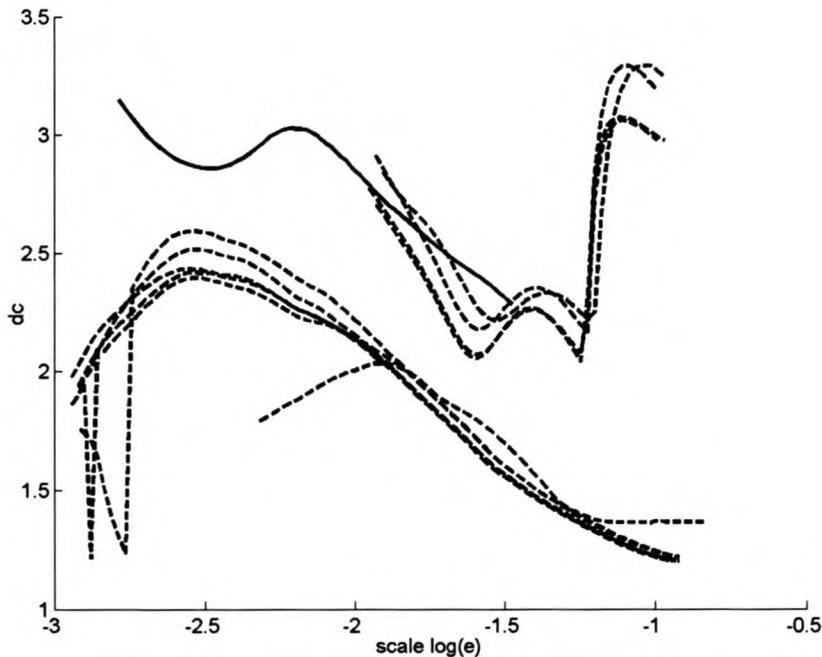
For the *MLPMOD\_1* model it can be seen that the surrogate data generated for the model formed two distinct clusters (figure 5.27). The purpose of generating more than one surrogate dataset is to get a good indication of where most of the surrogate datasets would lie for the model, should one be classified as an “outlier”. However, since two distinct clusters of surrogate datasets have formed, it could not be said for certain where the surrogates should lie, resulting in the method of surrogate data for the validation of the *MLPMOD\_1* model being ineffective.

From figures 5.28 and 5.29 it can be seen that the correlation dimension curves estimated from the *MLPSSA\_1* and *MLPSSA\_2* models are of similar, but consistently higher, correlation dimension than that of the actual process data. This indicates the possibility of the developed models being too complex, possibly due to overfitting of

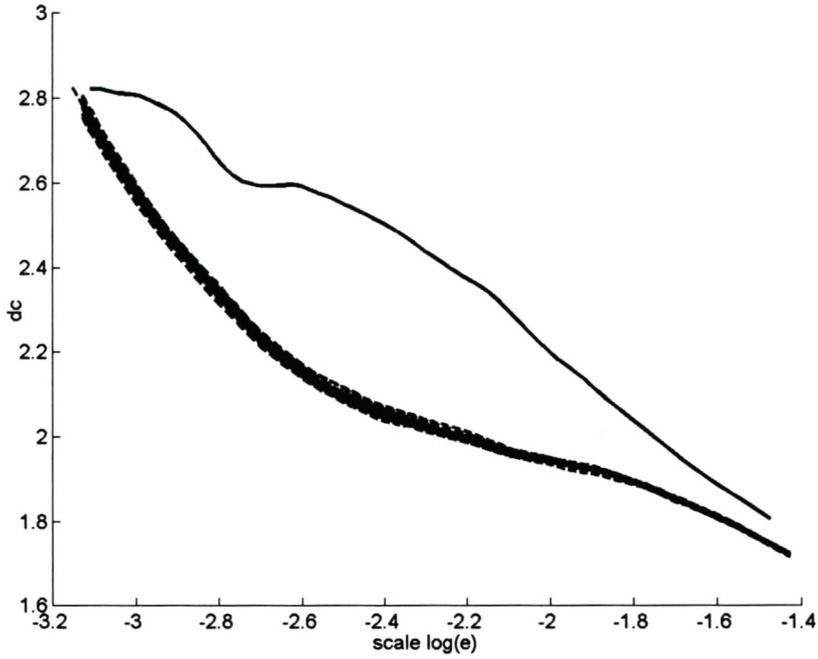
the training data, for the efficient modelling of the data using the second and third embedding strategies.

Figure 5.30 show that the *MLPSSA\_3* model was successful in capturing the structure of the data in the small-scale region of the dynamic attractor ( $\log \epsilon_0 < -2.6$ ). Although an intersection exist between the correlation dimension curve of the model data and that of the surrogate datasets at  $\log \epsilon_0 = -1.7$ , the *MLPSSA\_3* model was unsuccessful in capturing the structure of the data in the medium and large-scale region of the dynamic attractor. It should be noted that the correlation dimension curves for the *MLPSSA\_3* model and its associated surrogate datasets are of less overall value than those of the other models. This confirms that the *MLPSSA\_3* model, having only three hidden nodes, is a less complex model than the other models, having hidden nodes greater than four, evaluated here.

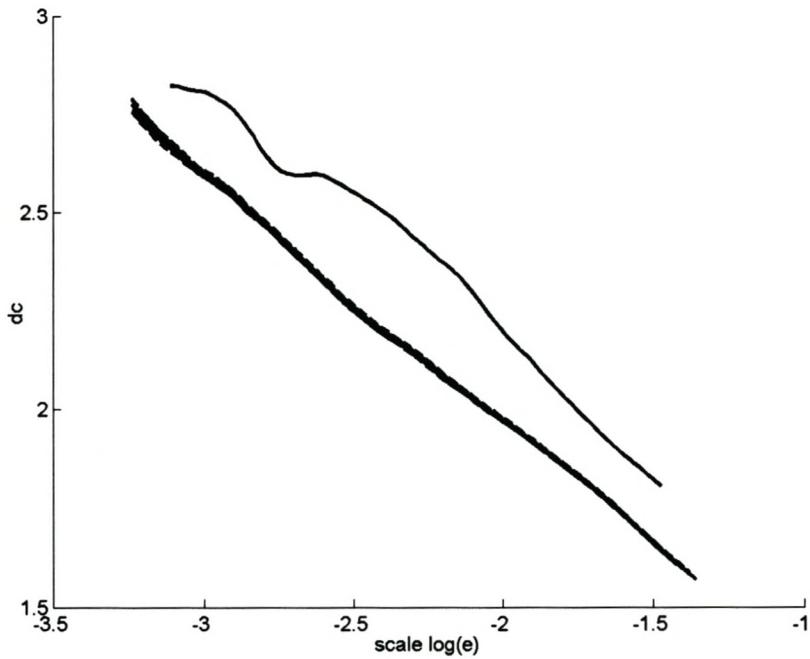
Although mention has been made of the dangers of applying the method of surrogate data for the validation of these models, the results, as before, were found to be in accordance with those obtained from the graphical residual analysis.



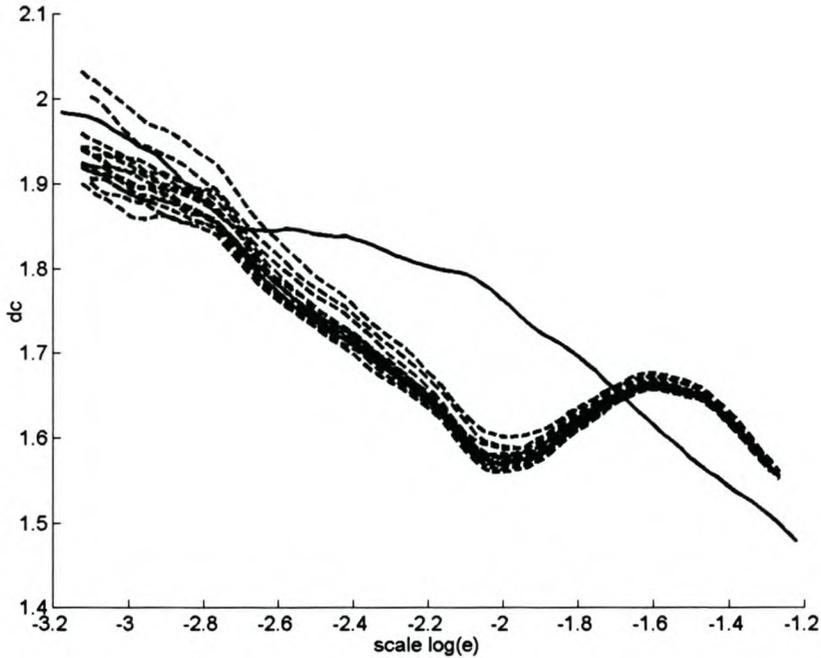
**Figure 5.27: Correlation dimension curve for FAG mill power ( $x_1$ ; solid line) and of the non-linear surrogates of *MLPMOD\_1* (broken lines)**



**Figure 5.28: Correlation dimension curve for FAG mill power ( $x_1$ ; solid line) and of the non-linear surrogates of *MLPSSA\_1* (broken lines)**



**Figure 5.29: Correlation dimension curve for FAG mill power ( $x_1$ ; solid line) and of the non-linear surrogates of *MLPSSA\_2* (broken lines)**



**Figure 5.30: Correlation dimension curve for FAG mill power ( $x_1$ ; solid line) and of the non-linear surrogates of *MLPSSA\_3* (broken lines)**

#### 5.1.3.6 Results discussion

From the above it became evident that, as suspected, a direct connection exists between the embedding strategy used, the quality of the reconstructed attractor, and the quality of the resulting model. It was concluded that for the system under investigation the better the quality of the attractor reconstruction and the more of the state space it covered, the more the attractor was unfolded and the better the quality of the resulting model; embedding strategy 5's reconstructed attractor, being the most unfolded, resulting in the best model.

As before, the SSA models all outperformed the MOD model for multi-dimensional embeddings; the *MLPMOD\_1* model performing much worse than even the previously developed *MLPMOD\_0* model. Owing to the noisy nature of the data being analysed, it was previously found that the AMI statistic produced uncertain values for the FAG mill discharge 1 flow rate ( $x_9$ ) and the FAG mill discharge 2 flow rate ( $x_{10}$ ) embedding delays. Also, the FFN statistic produced uncertain values for the FAG mill fine feed rate ( $x_4$ ) and the FAG mill discharge 1 flow rate ( $x_9$ ) embedding dimensions. Due to these doubtful values, the wrong information could have been

included in the MOD multi-dimensional embedding, leading to a worse predictive model when using multi-dimensional embedded data than when using one-dimensional embedded data.

Similar to the MOD multi-dimensional embedding results, the SSA multi-dimensional embedded model, *MLPSSA\_1*, also performed worse than the one-dimensional embedded SSA model. Therefore, the possibility that for the dynamic system under investigation, the FAG mill power ( $x_1$ ) not necessarily being a function of all the state variables of the system is remotely slim. This leaves the possibility that important information regarding the system is lost during the embedding procedure when applying data reduction techniques, such as principal component analysis, prior to modelling.

It was found that by only embedding the input variables of the system using the embedding parameters estimated by SSA, but still including the output variables in the multi-dimensional embedding with an embedding delay and embedding dimension of unity, a model, *MLPSSA\_3*, was found that outperformed the one-dimensional embedded model, *MLPSSA\_0*. From a control perspective, this model shows the importance of including the input variables of the system when predicting one of the systems output variables. Whereas the *MLPSSA\_0* model could be regarded as the best model for purely predictive purposes, based on the fact that its accuracy is very similar to that of the *MLPSSA\_3* model and it being the simplest model, the *MLPSSA\_3* model could be regarded as the best overall model, being the most accurate with regard to prediction and including the manipulated variables of the system, making it ideal for use in an advanced control system. It must nonetheless be kept in mind that the predicted values for the *MLPSSA\_3* model were offset to the right of the actual data leading to the suspicion that either the model did not generalise sufficiently or it required more model inputs. As before, this suspicion of insufficient generalisation was not confirmed through the results obtained from residual analysis.

#### **5.1.4 Modelling of the FAG mill power using LS-SVM's**

Based on the modelling results obtained using MLP neural networks and the associated embedding strategies, it was noted that the possibility existed whereby valuable information regarding the system is lost when the size of the model input

space is reduced during the embedding procedure. Since it has been found that LS-SVM's scale very well to high dimensional input spaces (Suykens, 2001) it was decided to compare the results obtained from LS-SVM models, where the size of the input space is not reduced during the embedding procedure, with those of the MLP neural network models, where the size of the input space was reduced during the embedding procedure. Furthermore, since experimental work done at the University of Stellenbosch has shown that trivial embeddings of the data prior to LS\_SVM modelling generally resulted in better models, only embedding strategies using embedding delays ( $k$ ) of unity will be investigated.

Based on the foregoing results and the above discussion it was decided to use the following embedding strategies for LS-SVM modelling:

1. None of the variables were embedded. All input and output variables at time  $t$  were used to predict the FAG mill power ( $x_1$ ) at time  $t+1$ . Model: *SVMnone*.
2. Only the FAG mill power ( $x_1$ ) time series was used as model input having an embedding delay ( $k$ ) of unity and embedding dimension ( $m$ ) equal to the point of linear decorrelation for the time series. Model: *SVM\_0*.
3. All variables were embedding using an embedding delay ( $k$ ) of unity and embedding dimension ( $m$ ) equal to the point of linear decorrelation for the respective variables. Model: *SVM\_1*.
4. All input variables were embedding using an embedding delay ( $k$ ) of unity and embedding dimension ( $m$ ) equal to the point of linear decorrelation for the respective variables. All output variables were also included in the multi-dimensional embedding with an embedding delay ( $k$ ) and embedding dimension ( $m$ ) of unity. Model: *SVM\_3*.
5. All variables were embedding using an embedding delay ( $k$ ) of unity and embedding dimension ( $m$ ) equal to the arbitrarily chosen maximum embedding dimension of 20 (the maximum point of linear decorrelation for any of the variables). Model: *SVM\_4*.

#### 5.1.4.1 Model structure

LS-SVM models were constructed for each of the proposed embedding strategies with the purpose of predicting only one-step ahead in time. For each embedding strategy the model parameters,  $\gamma$  and  $\sigma$ , were optimised by testing a set of LS-SVMs corresponding to any one combination of  $(\gamma, \sigma)_i$  in a proposed grid of  $(\gamma, \sigma)$ , where  $\sigma = [0.5, 1, 2, 4, 8, 16, 32]$  and  $\gamma = [0.1, 1, 10, 100, 1000, 10000, 100000]$ . The optimal  $(\gamma, \sigma)_i$  combination resulted from a minimal error on the validation data set, leading to a model with, hopefully, a good ability to generalise well. A summary of the modelling parameters is given in table 5.10.

**Table 5.10: LS-SVM modelling parameters**

	<i>SVM</i> <i>none</i>	<i>SVM_0</i>	<i>SVM_1</i>	<i>SVM_3</i>	<i>SVM_4</i>
$\sigma$	32	32	16	32	16
$\gamma$	100	100	1000	1000	1000

#### 5.1.4.2 Model validation

From the modelling results (tables 5.11) it can be seen that for predictive purposes the *SVM\_0* model would be best, it being the simplest model, with the *SVM\_1* and *SVM\_4* models, for possible use in an advanced control system, performing similarly.

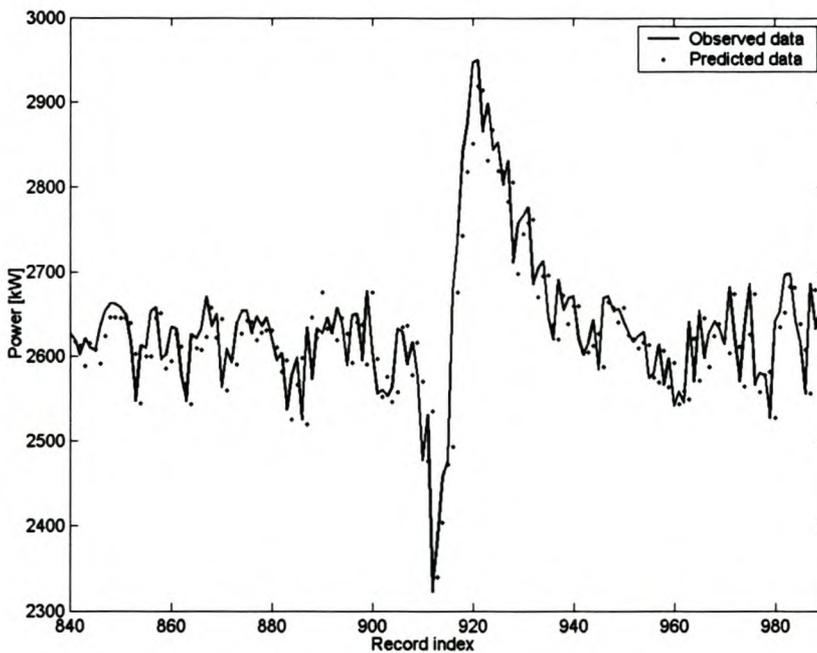
**Table 5.11: Summary of  $R^2$  [%] statistics for validation and independent test datasets**

Test set	Size	<i>SVM</i> <i>none</i>	<i>SVM_0</i>	<i>SVM_1</i>	<i>SVM_3</i>	<i>SVM_4</i>
Validation data	1407	84.17	85.46	85.82	82.23	85.21
Test dataset 1	6571	94.10	95.40	94.90	93.50	95.10
Test dataset 2	6091	96.20	97.20	97.50	96.30	97.50
Test dataset 3	4501	97.90	97.00	97.10	97.60	97.10

The reason for the *SVM\_1* and *SVM\_4* models performing similarly can be ascribed to the fact that their embeddings are very similar; four out of the seven variables being embedded in 20 dimensions (20 being the arbitrarily chosen maximum embedding dimension) for both strategies with a fifth variable being embedded in 16 dimensions (very close to 20) for the *SVM\_1* model. This leaves only two variables to be significantly more embedded for the *SVM\_4* model than for the *SVM\_1* model,

leading to inconclusive results regarding the effect of embedding all variables similarly at a chosen maximum embedding dimension when modelling using LS-SVM's.

When comparing the one-step prediction graphs for the various models obtained from testing with validation data (found in the appendix), it was found that visually the *SVM\_3* model looked to be the best model (figure 5.31). However, as found previously, the predicted values for the *SVM\_3* model seems to be offset to the right of the actual data leading one to suspect that either the model does not generalise sufficiently or it requires more model inputs. The fact that the model might not generalise sufficiently should be further investigated through the use of residual analysis.

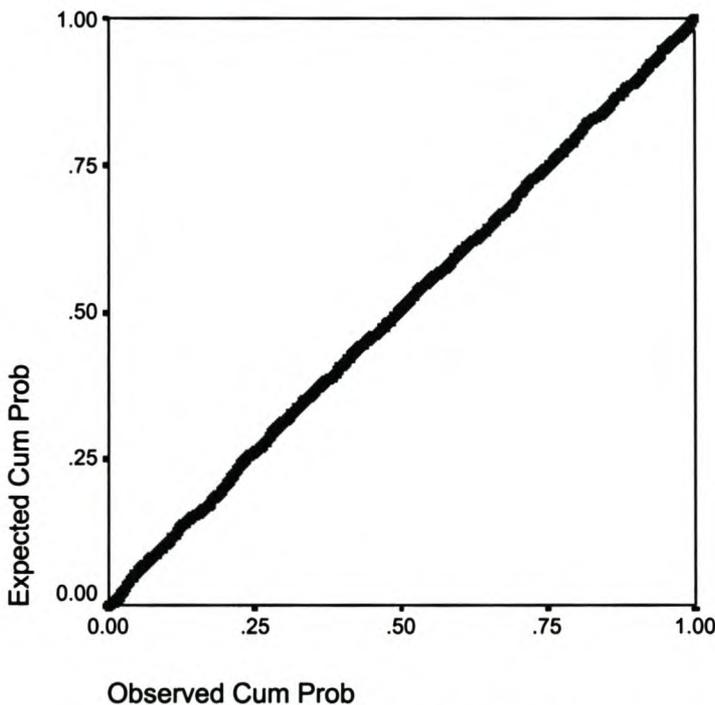


**Figure 5.31: One-step prediction of FAG mill power ( $x_1$ ) using *SVM\_3* ( $R^2 = 82.23\%$ )**

#### 5.1.4.3 Use of graphical residual analysis for model validation

Normal P-P plots of the residuals obtained from the application of all the LS-SVM models to one-step prediction of the validation data were plotted (found in the appendix). From most of the plots it was evident that the points tended towards

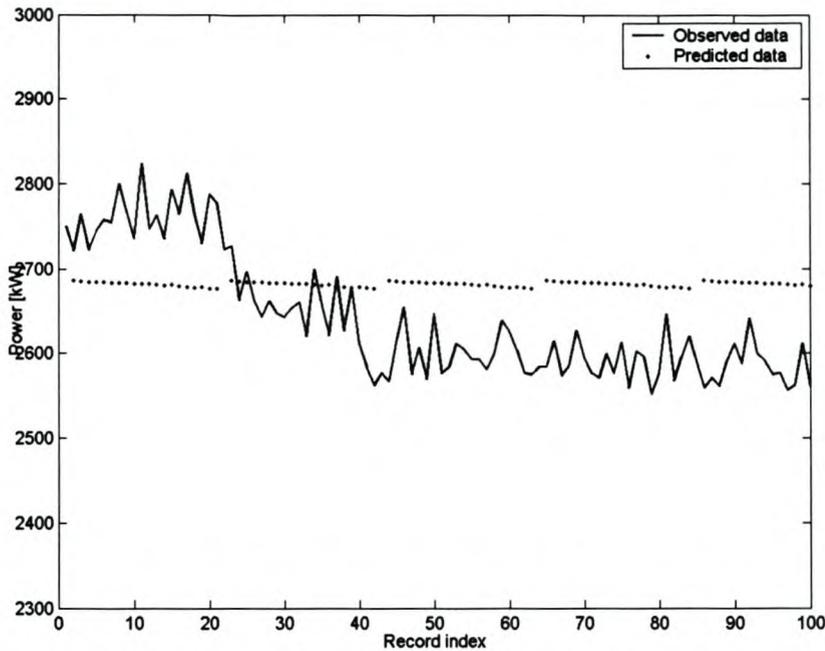
clustering around a straight line, showing better results than those obtained when modelling using MLP neural networks. Of the normal P-P plots, the one for the residuals obtained from the application of the *SVM\_3* model (figure 5.32) proved to be the straightest, indicating that either this model was best at extracting the available information from the data presented to it or that the data was of such a nature that it was more easily extracted than for the other embedding strategies. It is evident that the residuals obtained from the application of *SVM\_3* are normally distributed, indicating that the model does indeed generalise to a certain degree.



**Figure 5.32: Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using *SVM\_3***

#### 5.1.4.4 Free-run prediction

Using the *SVM\_0* model, a free-run prediction of the FAG mill power ( $x_1$ ) proved impossible (figures 5.33). Similar to the results obtained from the application of the MLP neural networks models for free-run prediction, the lack of ability of these LS-SVM models to be used for free-run prediction could be ascribed to the fact that the models are highly dependent on the initial conditions of the data and the possibility of the FAG mill power ( $x_1$ ) variable not containing all the information of the system being modelled.



**Figure 5.33: Free-run prediction of FAG mill power ( $x_1$ ) using  $SVM_0$**

Since the free-run prediction results proved to be very poor, the use of surrogate data for model validation would lead to inaccurate results. This is a direct result from the fact that, as stated earlier, the method of surrogate data requiring a free-run prediction of the model output variables. Therefore, should the free-run predicted time series clearly differ from the actual time series, the surrogate data set based on the free-run predicted time series will also differ markedly from the surrogate data set based on the actual time series. The use of surrogate data for model validation on a model delivering poor free-run prediction results would therefore not add significantly to the interpretation of the modelling results and is deemed unnecessary in this instance.

#### 5.1.4.5 Results discussion

From the above it became evident that the LS-SVMs did not fare significantly better than the MLP neural networks at modelling the given data. Furthermore, it was found that although there was no need to accurately determine the embedding parameters or to apply data reduction to the data when using LS-SVMs for modelling, model development (in terms of computer processing time) took much longer than when using MLP neural networks, without significantly increasing prediction accuracy.

Based on these findings it would therefore not be sensible to use LS-SVMs for the modelling of a metallurgical system similar to the one being investigated in this thesis without first attempting to use state space reconstruction and MLP neural networks for the purpose. However, should MLP neural networks fail, modelling using LS-SVMs could prove invaluable.

### **5.1.5 Modelling of the FAG mill power using a RBF neural network**

From both the MLP neural network and LS-SVM modelling it became evident that although reasonable one-step predictions were obtainable using the specified embedding strategies, much work is still required when considering the free-run prediction results. Owing to the advantages of RBF neural networks over MLP neural networks, as discussed in chapter 3, and the fact that their application is rather straightforward when using the Matlab functions as developed by Orr (1999b), it was decided to test the ability of RBF neural networks as an alternative to MLP neural networks and LS-SVMs with the aim of improving free-run prediction results.

Similar embedding strategies to those for the MLP neural network model development were employed for RBF neural network model building, and numbering of the models was done accordingly.

#### **5.1.5.1 Model structure**

RBF neural network models were constructed for each of the proposed embedding strategies with the initial aim of predicting only one-step ahead in time; an accurate one-step prediction improves the chances of an accurate free-run prediction.

Based on work done by Orr (1999b) and trial-and-error testing by the author of this thesis, the following model parameters were fixed prior to model training:

- The initial number of RBF units and their centres were set as a random subset of the columns of the input vector to reduce the computational burden (Orr, 1999b).

- The nominal size of the RBF units was set at the default value, the value being the difference between the maximum and minimum value for each of the model inputs (Orr, 1999b).
- Since the default setting was used for the nominal size of the RBF units, the scales value was set at a value smaller than 1 (Orr, 1999b). Through trial-and-error testing, the scales value was fixed at 0.1.
- It was decided to include a bias unit in each of the neural network models.
- Based on Orr (1999b) and through trial-and-error testing, the model selection criterion was set as the Bayesian Information Criterion (BIC).
- RBFs were stopped being added to the neural network model once the model selection criterion decreased by less than 10000 for 2 consecutive iterations (Orr, 1999b).
- The type of RBF was set as multiquadric through trial-and-error testing.
- The sharpness of the RBFs was set as 2 (Orr, 1999b).
- The initial value for the regularisation parameters was set at  $10^{-9}$  (Orr, 1999b).
- Regularised forward selection was turned on.

A summary of the modelling parameters subsequent to training, giving an idea as to the size of the RBF neural network models, is given in table 5.12.

**Table 5.12: RBF modelling parameters**

	<i>RBF MOD 0</i>	<i>RBF SSA 0</i>	<i>RBF none</i>	<i>RBF MOD 1</i>	<i>RBF SSA 1</i>	<i>RBF SSA 2</i>	<i>RBF SSA 3</i>
Number of RBF units (model inputs x hidden units)	5x84	3x25	7x96	13x99	16x112	11x113	15x72
Bias	35	17	1	20	3	2	3
RBF sharpness	2	2	2	2	2	2	2
$\lambda$	0.181	0.021	0.750	3.012	2.211	2.502	2.695
$\gamma$	83.22	25.89	90.01	94.62	103.4	94.61	67.08

From table 5.12 it is evident that a strong correlation exists between the model complexity and  $\gamma$ ; and increase in model complexity generally leading to an increase in  $\gamma$ . Also, in contrast with the discussion in chapter 3, the RBF neural network models require significantly more hidden units than the MLP neural network models.

A possible cause for this could be the fact that for the RBF neural network models the inputs aren't weighted before being distributed to the hidden layer whereas for the MLP neural network models the inputs are weighted.

#### 5.1.5.2 Model validation

From the modelling results (tables 5.13 and 5.14) it can be seen that the models based on the embedding strategies using SSA for embedding parameter estimation again outperformed those based on the embedding strategies using MOD for embedding parameter estimation.

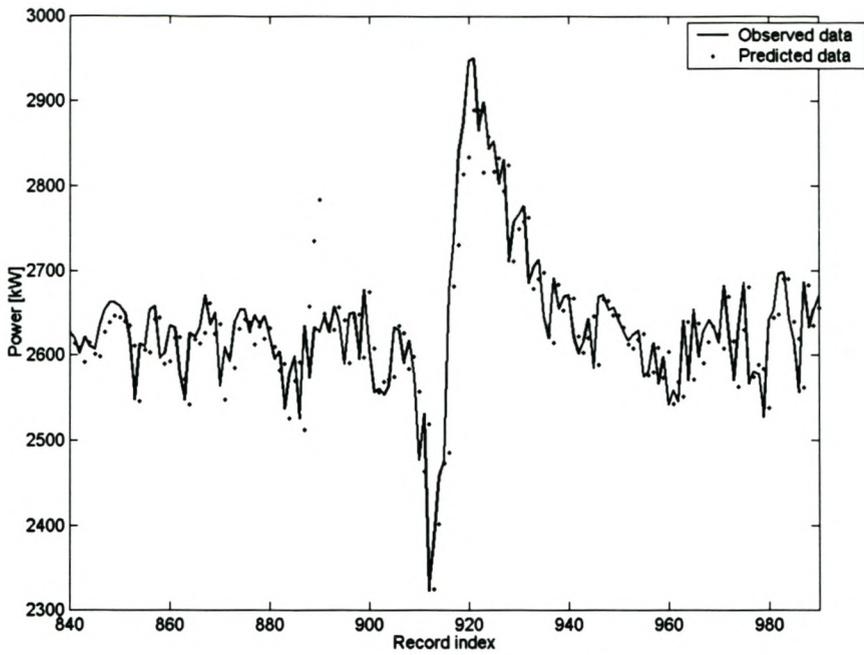
**Table 5.13: Summary of  $R^2$  [%] statistics for validation and independent test datasets**

Test set	Size	<i>RBF</i> <i>MOD 0</i>	<i>RBF</i> <i>SSA 0</i>
Validation data	1407	46.37	82.41
Test dataset 1	6571	92.28	94.23
Test dataset 2	6091	94.03	96.57
Test dataset 3	4501	94.76	94.82

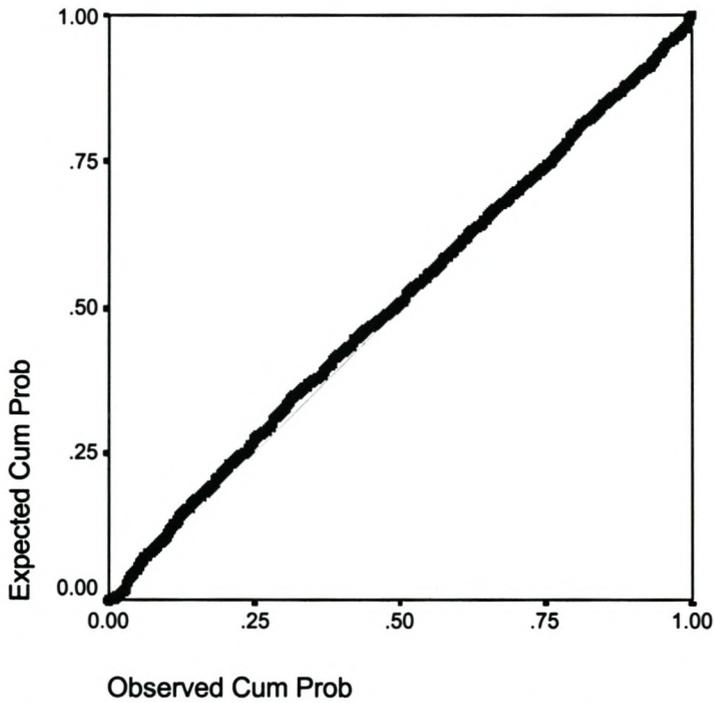
**Table 5.14: Summary of  $R^2$  [%] statistics for validation and independent test datasets**

Test set	Size	<i>RBF</i> <i>none</i>	<i>RBF</i> <i>MOD 1</i>	<i>RBF</i> <i>SSA 1</i>	<i>RBF</i> <i>SSA 2</i>	<i>RBF</i> <i>SSA 3</i>
Validation data	1407	77.99	39.91	82.69	78.44	80.60
Test dataset 1	6571	88.97	89.94	94.90	93.12	90.35
Test dataset 2	6091	93.02	68.60	96.98	94.42	95.59
Test dataset 3	4501	96.12	96.48	95.49	93.12	97.37

When comparing the one-step prediction graphs for the various models obtained from testing with validation data (found in the appendix) to those obtained from testing the MLP neural network models, it was found that the RBF neural network results proved very similar to that of the MLP neural network results. For the normal P-P plots of the residuals obtained from the application of the various RBF and MLP neural network models to one-step prediction of the validation data this also seemed to be the case. For comparison purposes figure 5.34 shows the one-step prediction results obtained from application of the *RBFSSA\_3* model and figure 5.35 shows the normal P-P plot of the associated residual values.



**Figure 5.34: One-step prediction of FAG mill power ( $x_1$ ) using *RFSSA\_3* ( $R^2 = 80.60\%$ )**



**Figure 5.35: Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using *RFSSA\_3***

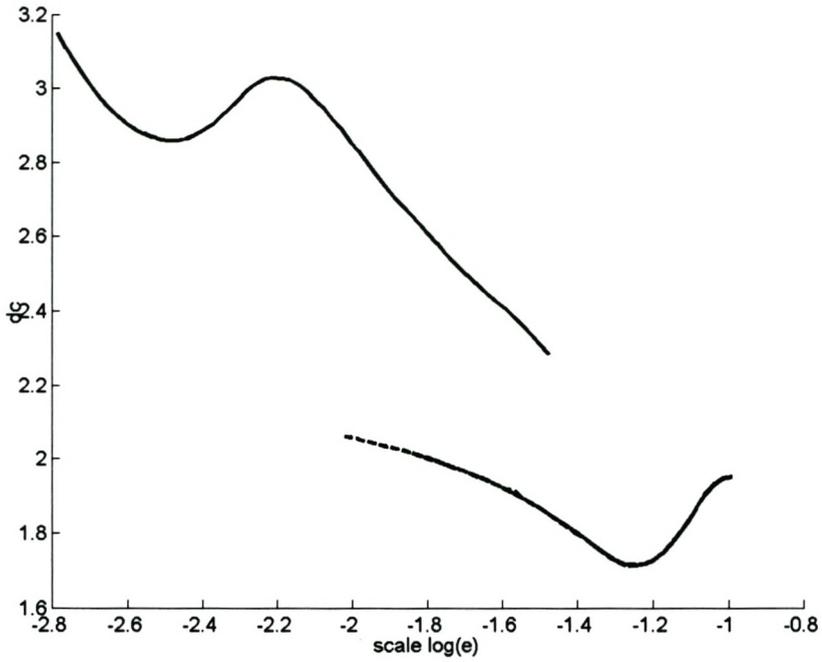
### 5.1.5.3 Use of surrogate data for model validation

Based on the reasons discussed earlier in this chapter, it was decided to only validate the RBF neural network models based on the one-dimensional time series using surrogate data.

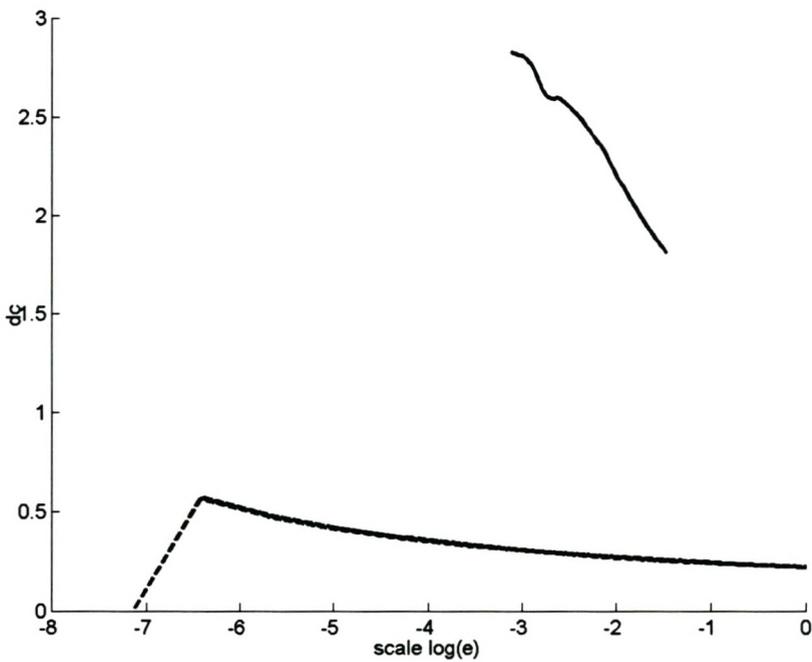
From figures 5.36 and 5.37 it can be seen that the correlation dimension curves estimated from the *RBFMOD\_0* and *RBFSOA\_0* models are of a consistently lower correlation dimension than that of the actual process data. This indicates the possibility of the developed models being too simple for the accurate modelling of the complex data using the specified embedding strategies.

Whereas the actual data used for the training of the *RBFMOD\_0* model covers the small-scale region ( $\log \epsilon_0 < -1.45$ ) of the dynamic attractor, the surrogate data derived from the same model mainly covers the large-scale region ( $\log \epsilon_0 > -2$ ) of the dynamic attractor. In contrast to this, the surrogate data derived from the application of the *RBFSOA\_0* model to free-run prediction covers a very large region ( $-7 < \log \epsilon_0 < 0$ ) compared to the region covered by the model training data ( $-3 < \log \epsilon_0 < -1.5$ ).

Again, these findings should be treated with caution. As stated previously, model validation using surrogate data on a model delivering poor free-run prediction results would not add significantly to the interpretation of the modelling results. The validation results should therefore be analysed in conjunction with free-run prediction results.



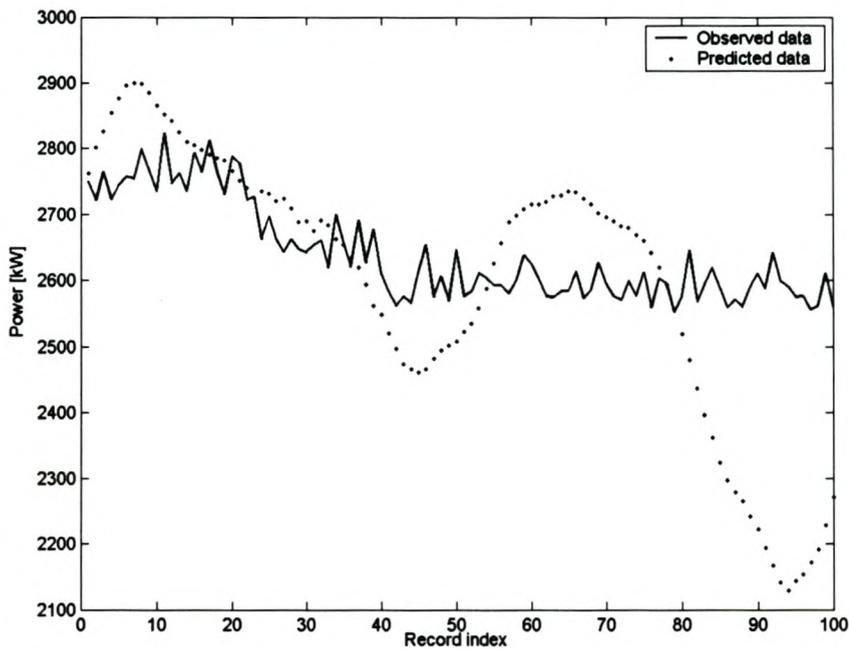
**Figure 5.36: Correlation dimension curve for FAG mill power ( $x_1$ ; solid line) and of the non-linear surrogates of *RBFMOD\_0* (broken lines)**



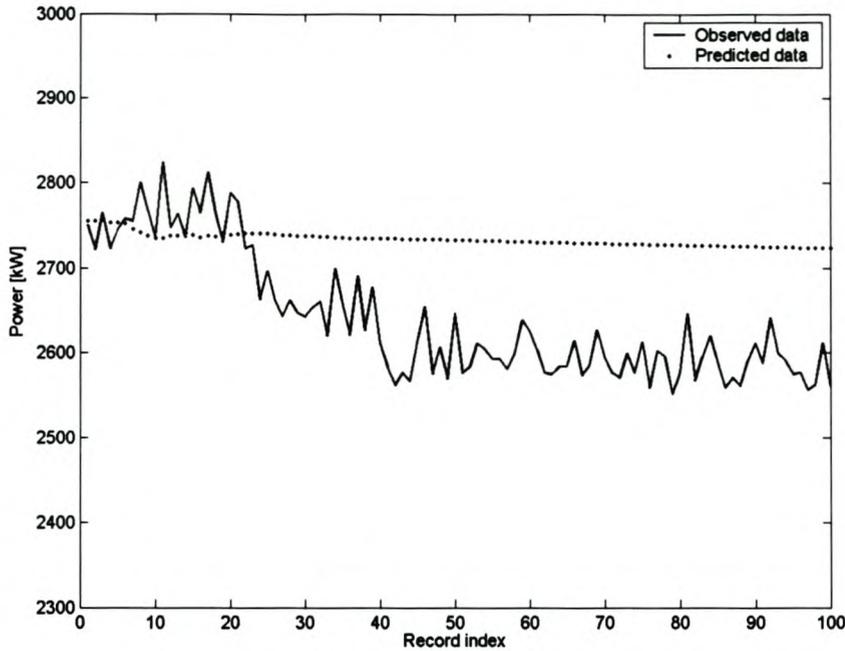
**Figure 5.37: Correlation dimension curve for FAG mill power ( $x_1$ ; solid line) and of the non-linear surrogates of *RBFSSA\_0* (broken lines)**

#### 5.1.5.4 Free-run prediction

From the free-run prediction plots (figures 5.38 and 5.39) it can be seen that, as before, the RBF neural network model based on the MOD embedding strategies outperformed the RBF neural network model based on the SSA embedding strategy. However, neither of the free-run prediction plots were very good. Again, the lack of ability of these models to be used for free-run prediction could be ascribed to the fact that the models are highly dependent on the initial conditions of the data and the possibility of the FAG mill power ( $x_1$ ) variable not containing all the information of the system being modelled.

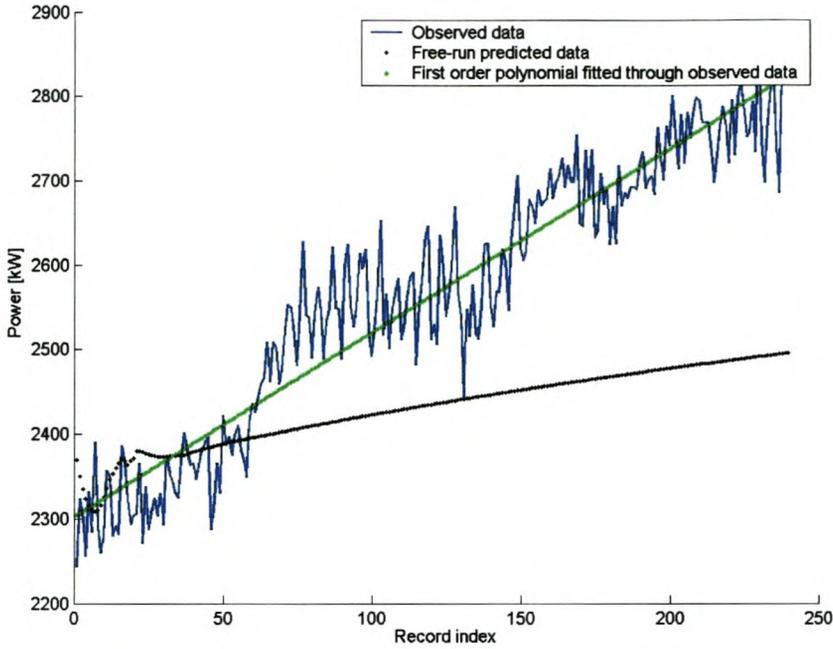


**Figure 5.38: Free-run prediction of FAG mill power ( $x_1$ ) using *RBFMOD\_0***

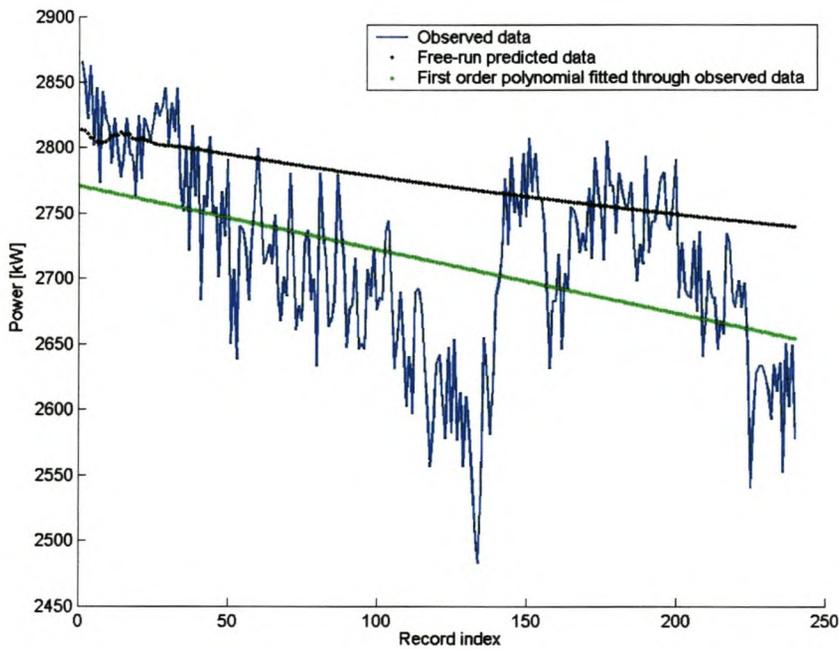


**Figure 5.39: Free-run prediction of FAG mill power ( $x_1$ ) using *RBFS*<sub>0</sub>**

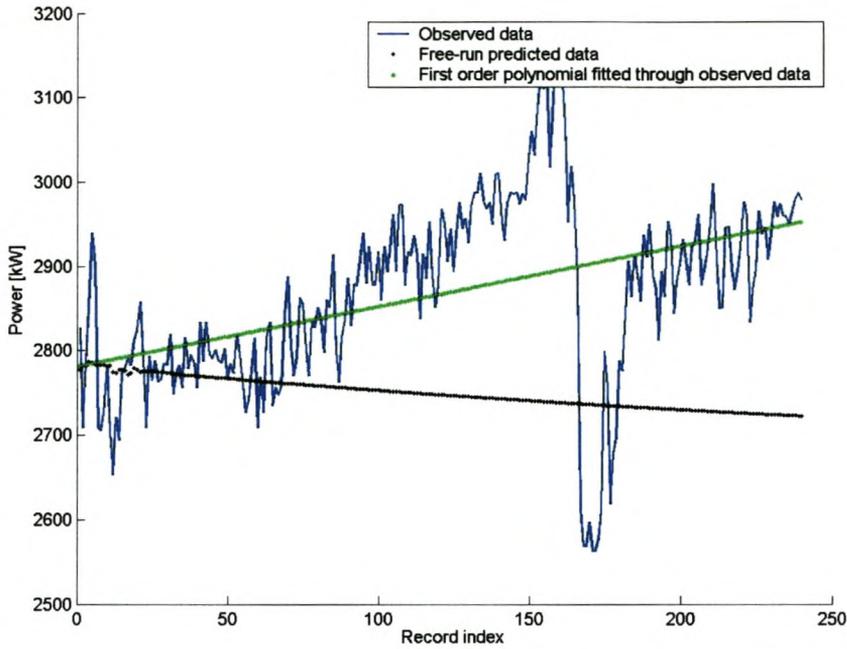
It was also noted that whereas the free-run prediction of the *RBFS*<sub>0</sub> model tried to follow the general trend in the data, the *RBFS*<sub>0</sub> model gave an indication as to the direction in which the process is going to move, the FAG mill power ( $x_1$ ) either increasing or decreasing on average over the prediction interval. Subsequently sixty preliminary free-run prediction tests were completed using the *RBFS*<sub>0</sub> model in order to determine if anything useful could be gained from this information. Based on the test results, figures 5.40 – 5.43 showing typical results, it was found that in excess of 70% correctly predicted whether or not the FAG mill power ( $x_1$ ) was increasing or decreasing on average over the prediction interval. However, these findings aren't conclusive and further investigation into this phenomenon is required.



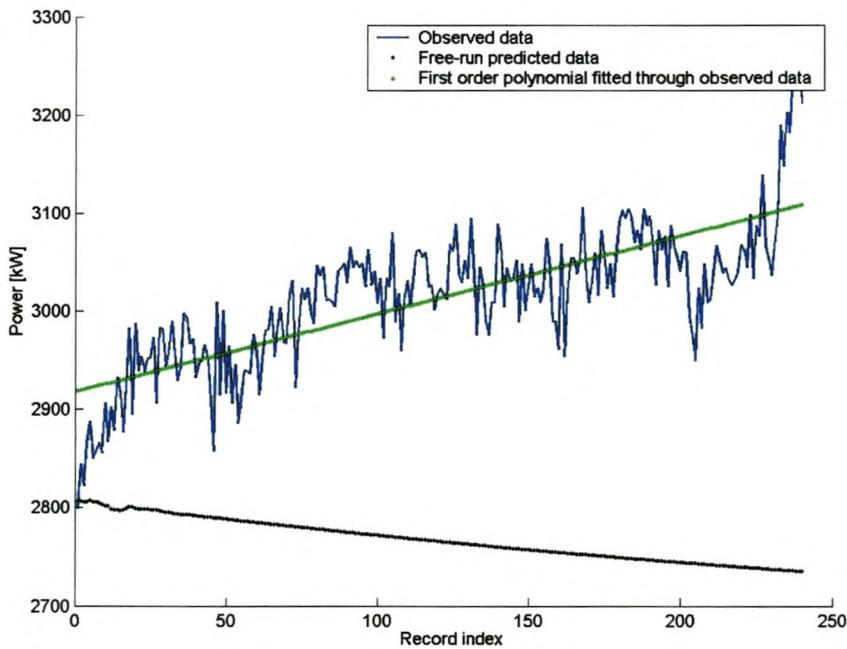
**Figure 5.40: Free-run prediction using *RFSSA\_0*; correctly predicted an increase in FAG mill power ( $x_1$ ) over the prediction interval**



**Figure 5.41: Free-run prediction using *RFSSA\_0*; correctly predicted a decrease in FAG mill power ( $x_1$ ) over the prediction interval**



**Figure 5.42: Free-run prediction using *RFSSA\_0*; incorrectly predicted a decrease in FAG mill power ( $x_1$ ) over the prediction interval**



**Figure 5.43: Free-run prediction using *RFSSA\_0*; incorrectly predicted a decrease in FAG mill power ( $x_1$ ) over the prediction interval**

#### 5.1.5.5 Results discussion

Based on the preceding discussing it can be concluded that the RBF neural network models performed very similarly to the MLP neural network models when considering the one-step prediction results.

When considering the free-run prediction results, it would seem as if the one-dimensional RBF neural network models were better suited than either the MLP neural network or LS-SVM models. Using the RBF neural network model based on only the FAG mill power ( $x_1$ ) time series, *RBFSSA\_0*, preliminary results indicated that the model could successfully predict in excess of 70% of the time whether or not the FAG mill power ( $x_1$ ) was increasing or decreasing on average over the prediction interval. Based on these findings, it can be concluded that RBF neural networks seems to be better at capturing the underlying dynamics in the data than either MLP neural networks or LS-SVMs.

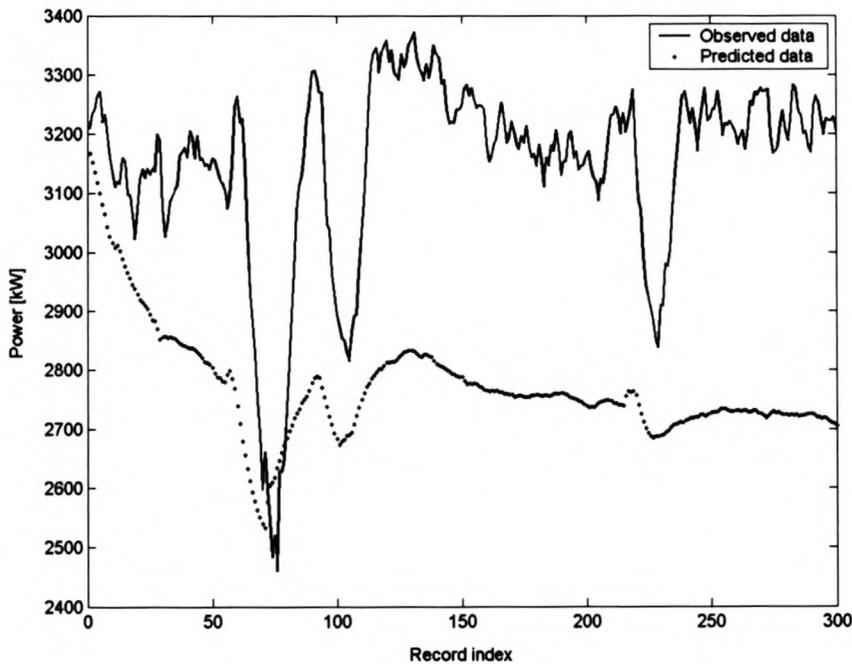
#### **5.1.6 Application of multi-dimensional time series models of the FAG mill power ( $x_1$ ) to free-run prediction**

From the above analysis it can clearly be seen that in all instances the one-dimensional time series models proved to be very good when considering one-step prediction of the FAG mill power ( $x_1$ ) time series. Unfortunately these models completely failed to closely follow the FAG mill power ( $x_1$ ) time series during free-run predictions. It also became evident that the multi-dimensional time series models did not greatly improve, if at all, on the one-step prediction results obtained for the one-dimensional time series models. Additionally, no clear distinction could be made between some of the results obtained from the various multi-dimensional time series models investigated, necessitating the use of some other criterion for better classification of the results.

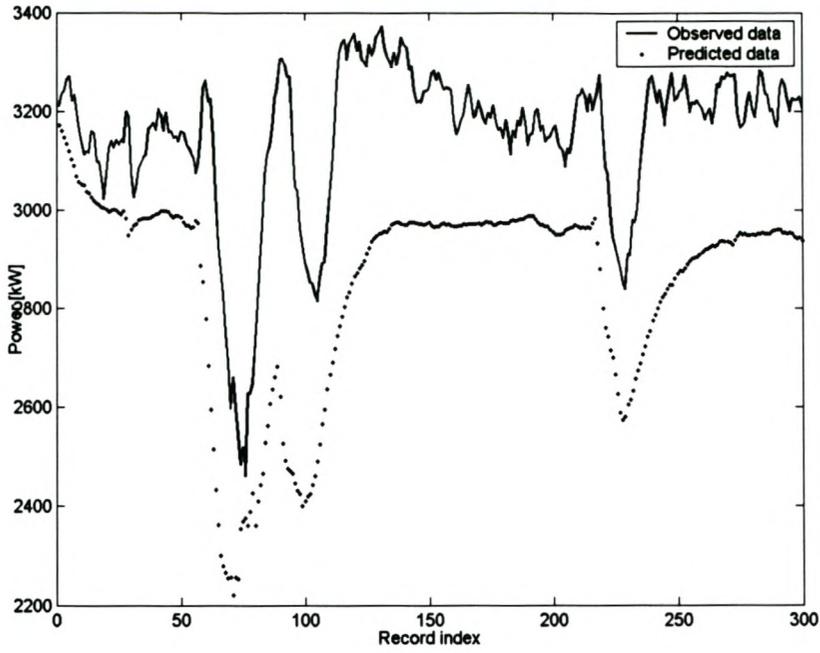
As an alternative to one-step predictions, free-run predictions are considered a more rigorous test of the validity of a model. However, since the multi-dimensional time series models under investigation have both process input and output variables as model inputs, and a single process output variable as model output, a true free-run prediction is not obtainable. When applying the developed multi-dimensional time series models to free-run prediction the results obtained could be considered useful in

terms of a soft sensor for predicting the process output variable given by the model output e.g. during instrument failure. Should similar multi-dimensional time series models be developed for prediction of the other process output variables, a true free-run prediction could be completed where the results obtained could be considered useful in terms of simulation.

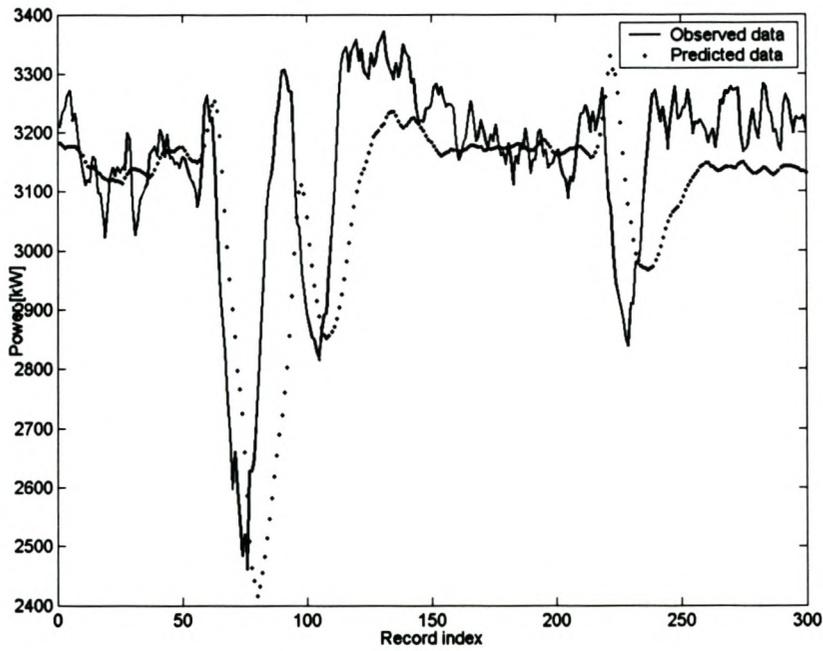
Figures 5.44 – 5.50 shows the results obtained from applying the various multi-dimensional time series models to free-run prediction. The data on which the free-run predictions are based is a subset selected from test dataset 3. Due to the nature of the free-run prediction results obtained no quantitative measure was used to compare the results, only visual inspection of the plots obtained were relied upon.



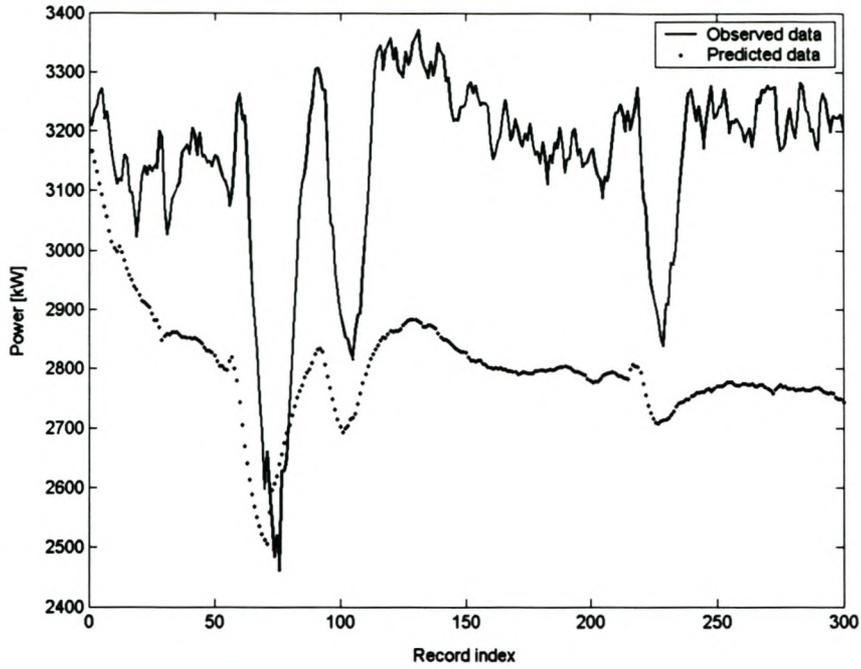
**Figure 5.44: Free-run prediction using *linearARX***



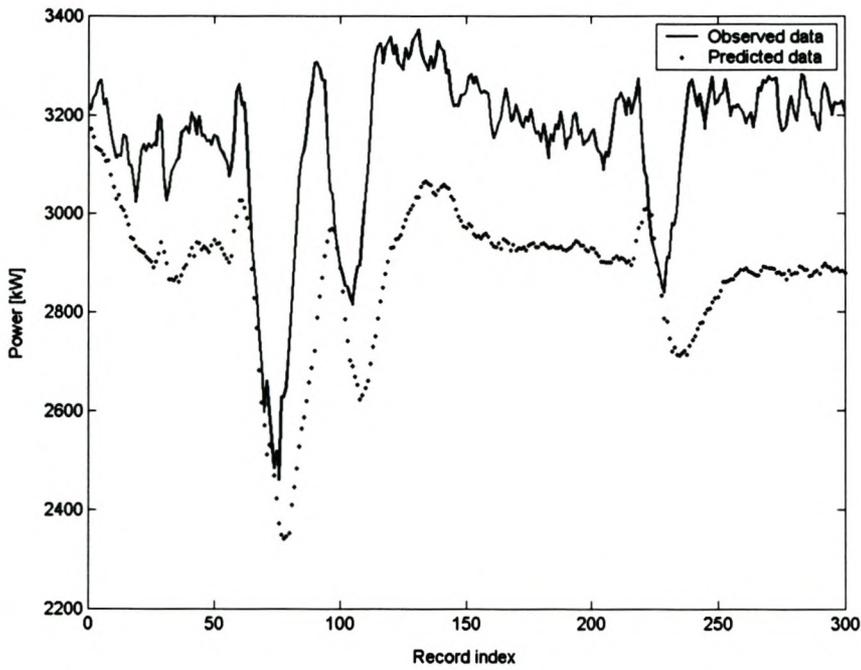
**Figure 5.45: Free-run prediction using *MLPnone***



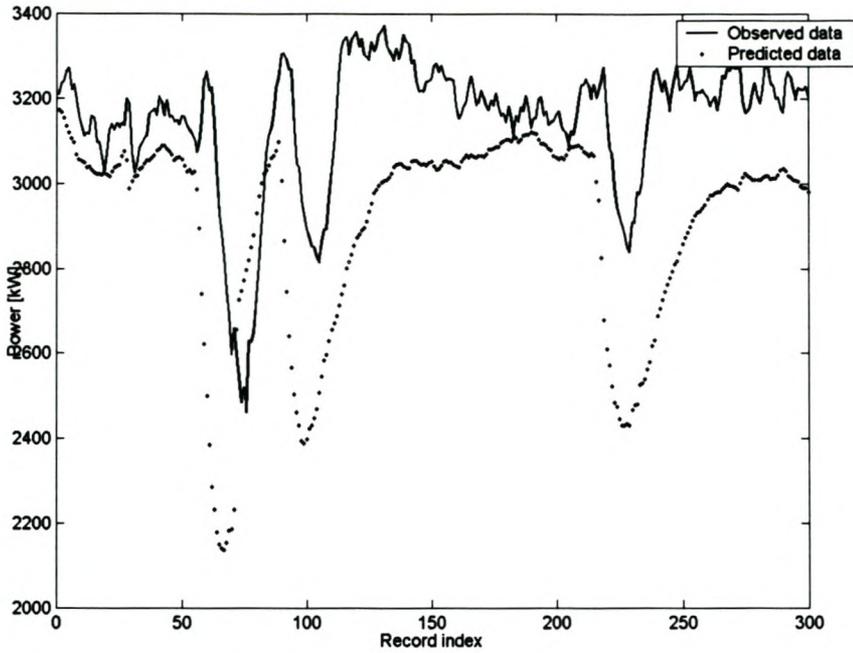
**Figure 5.46: Free-run prediction using *MLPSSA\_3***



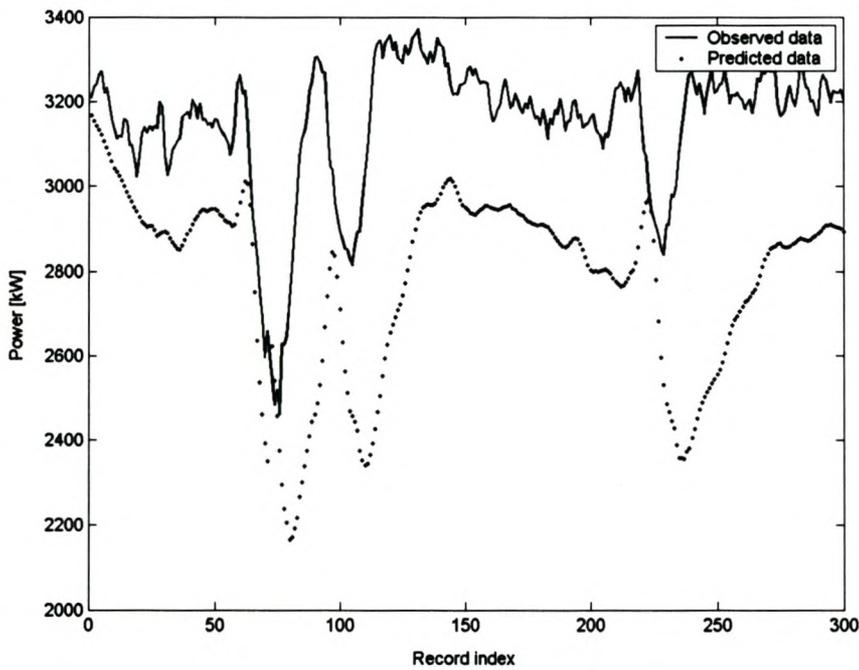
**Figure 5.47: Free-run prediction using *SVMnone***



**Figure 5.48: Free-run prediction using *SVM\_3***



**Figure 5.49: Free-run prediction using *RBFnone***



**Figure 5.50: Free-run prediction using *RBFSSA\_3***

From the above it can clearly be seen that whereas the *linearARX* model (figure 5.44) performed the worst when applied to free-run prediction, the *MLPSSA\_3* model (figure 5.46) performed best.

Although it can be seen that the *SVM\_3* model (figure 5.48) outperformed the *SVMnone* model (figure 5.47), even better free-run prediction results were obtained from the *SVM\_1* and *SVM\_4* models (found in the appendix). This could be ascribed to the much larger input space used for both the *SVM\_1* and *SVM\_4* models due to the embedding parameters used. In turn, this also confirms the one-step prediction results obtained as well as the fact that for LS-SVM models valuable information regarding the system is lost when a reduced model input space is used. In contrast with this, the *RBFnone* model seemed to outperform the *RBFSSA\_3* model, with the *RBFSSA\_1* model seemingly being the best performing RBF neural network model. Furthermore, it was stated that for the one-dimensional time series models using a RBF neural network the models seemed to be better at capturing the underlying dynamics in the data than either MLP neural networks or LS-SVMs. This is clearly not the case for the multi-dimensional time series models where the *MLPSSA\_3* model performed the best during free-run prediction.

When analysing the plots obtained from applying the various developed models to one-step prediction it was initially suspected that the generalisation of the models were poor. Using residual analysis, this was found not to be the case for many models; the above free-run prediction results serving as further confirmation. It is, however, very important to note that these results again only show the potential of the various models when applied to free-run prediction. They are dependent on similarly accurate models for the other process output variables and the current results aren't consistently obtainable, only showing the expected capability of the various models.

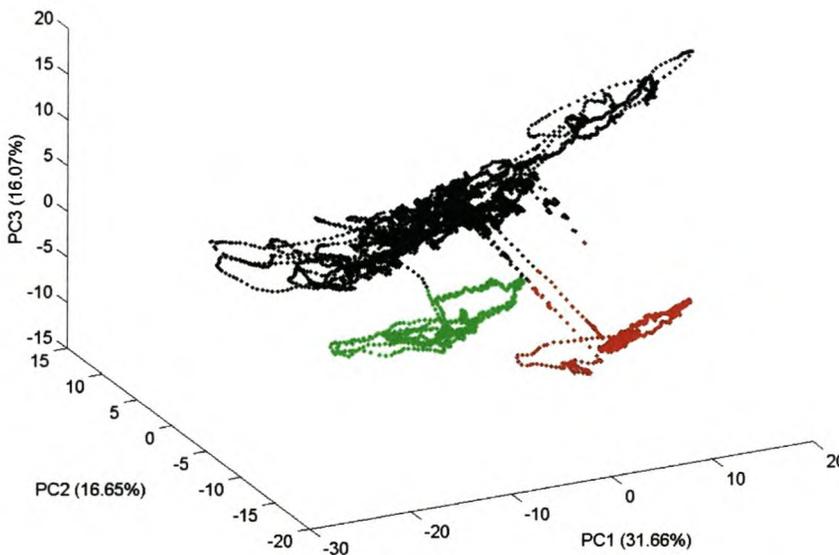
The above free-run prediction results clearly shows the promise of embedding the data prior to non-linear modelling. Improved models, resulting in improved free-run prediction results, could be expected when using more training data covering a larger region of the state space. The use of training data that homogeneously populate the state space could also be investigated to improve the modelling results.

#### **5.1.7 Prediction of FAG mill rougher feed 2 flow rate ( $x_{12}$ )**

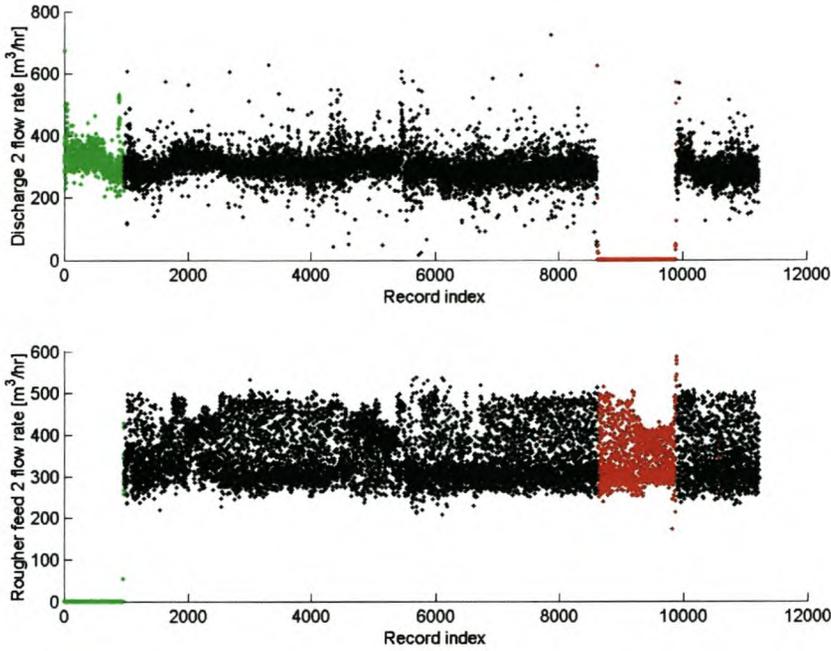
Based on the foregoing LS-SVM modelling results, it was decided to test the findings regarding the use of a fixed embedding dimension and delay strategy when modelling the plant data contained in the Process dataset using LS-SVMs; making full use of the

fact that LS-SVMs scale very well to high dimensional input spaces. Since a stable, continuous feed to the flotation section is very important for optimum mineral extraction, it was decided to use all the available information to construct a prediction model for the FAG mill rougher feed 2 flow rate ( $x_{12}$ ) with the aim of possibly using this model in an advanced control system.

As with the *SVM\_4* model, all variables were embedded using an embedding dimension of 20 and an embedding delay of unity. When looking at the reconstructed attractor (figure 5.51), a definite pattern is visible. It further shows the reconstructed attractor occupying three distinct areas in the state space, the largest containing most of the observations. The origin of the three areas occupied by the reconstructed attractor can clearly be traced back to the FAG mill discharge 2 flow rate ( $x_{10}$ ) and the FAG mill rougher 2 flow rate ( $x_{12}$ ) (figure 5.52).



**Figure 5.51: Reconstructed attractor of data using an embedding dimension of 20 and an embedding delay of unity**

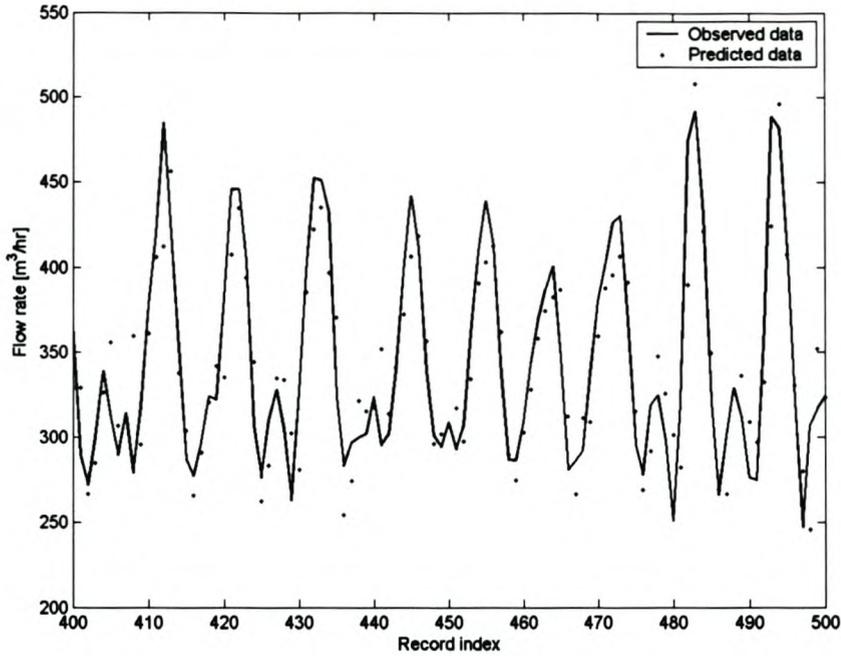


**Figure 5.52: Plot of FAG mill discharge 2 flow rate ( $x_{10}$ ) and rougher feed 2 flow rate ( $x_{12}$ )**

From the embedded data, a LS-SVM model, *SVM\_5*, was constructed having modelling parameters  $\gamma = 1000$  and  $\sigma = 16$ . The modelling results (table 5.15 and figure 5.53) show that the *SVM\_5* model for predicting the FAG mill rougher feed 2 flow rate ( $x_{12}$ ) performed fairly well when considering the result achieved using a similar strategy for the modelling of the FAG mill power ( $x_1$ ). However, the model completely failed to predict the FAG mill rougher feed 2 flow rate ( $x_{12}$ ) when the pump was not running. This resulted from the data used for the model training not containing enough information for switching between the FAG mill rougher feed 2 flow rate ( $x_{12}$ ) pump and the FAG mill rougher feed 1 flow rate ( $x_{11}$ ) pump.

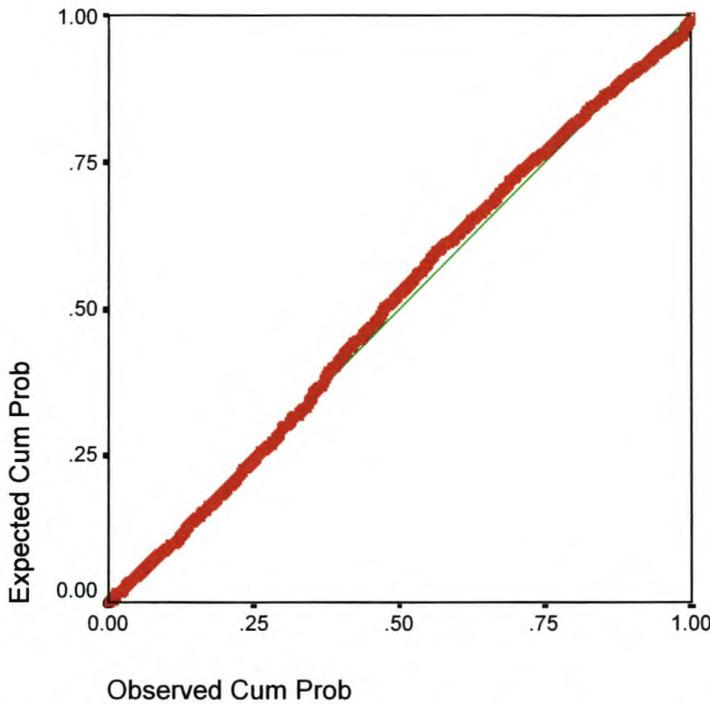
**Table 5.15: Summary of  $R^2$  [%] statistics for validation and independent test datasets**

Test set	Size	<i>SVM_5</i>
Validation data	1407	77.89
Test dataset 1	2800	80.40
Test dataset 2	6091	N/A
Test dataset 3	1180	79.80



**Figure 5.53: One-step prediction of FAG mill rougher feed 2 flow rate ( $x_{12}$ ) using *SVM\_5* ( $R^2 = 77.89\%$ )**

Figure 5.54 shows that as with the models for predicting the FAG mill power ( $x_1$ ), the LS-SVM model, *SVM\_5*, extracted nearly all the available information presented to it during training. This can clearly be seen from the normal P-P plot of the residual values for one-step prediction of the FAG mill rougher feed 2 flow rate ( $x_{12}$ ) using *SVM\_5* by the points, as before, tending towards clustering around a straight line. It is, however, clear that the residual values do deviate somewhat from a straight line, indicating that they are non-normally distributed. The possibility of using these residuals as model inputs for further modelling to try and extract the determinism in the data not captured by *SVM\_5* therefore exists.



**Figure 5.54: Normal P-P plot of the residual values for one-step prediction of the FAG mill rougher feed 2 flow rate ( $x_{12}$ ) using *SVM\_5***

From the above analysis it would therefore seem possible, if given a dataset of important inputs and outputs around a fully autogenous mill having a sampling interval of 30 seconds, to “quickly” construct a model of the mill to predict one of the mills measured outputs; the model being a LS-SVM built without explicitly selecting the input variables for the model and fixing the embedding delay at unity and the embedding dimension at the arbitrarily chosen value of 20. The resulting model should be fairly accurate with regard to control purposes and may or may not be improvable.

## 5.2 Wipfrag and Malvern datasets

For a milling circuit, from a control perspective, an important consideration is often the control of the mill product size distribution (Woollacott and Eric, 1994). For the FAG mill under consideration we have the FAG mill product size distribution (% -  $75\mu\text{m}$ ;  $x_{14}$ ) available for modelling. During the initial data analysis of this variable, using the method of surrogate data for data classification, it was found that this time series was almost purely stochastic for all practical purposes, indicating that it could

prove very difficult, if possible at all, to model this data. This stochastic nature of the FAG mill product size distribution (% -75 $\mu$ m;  $x_{14}$ ) time series could be ascribed to inadequate measurements and variable sampling rates. Similarly, the time series for the FAG mill feed size distribution (% -31.5mm;  $x_{13}$ ), one of the principal variables that can affect the control of an autogenous mill, was found to be stochastic. This was expected as the time series originated from a stochastic process. The stochastic nature of both these time series are indicative that, in all probability, it will be impossible to model this data.

### 5.2.1 Embedding the data

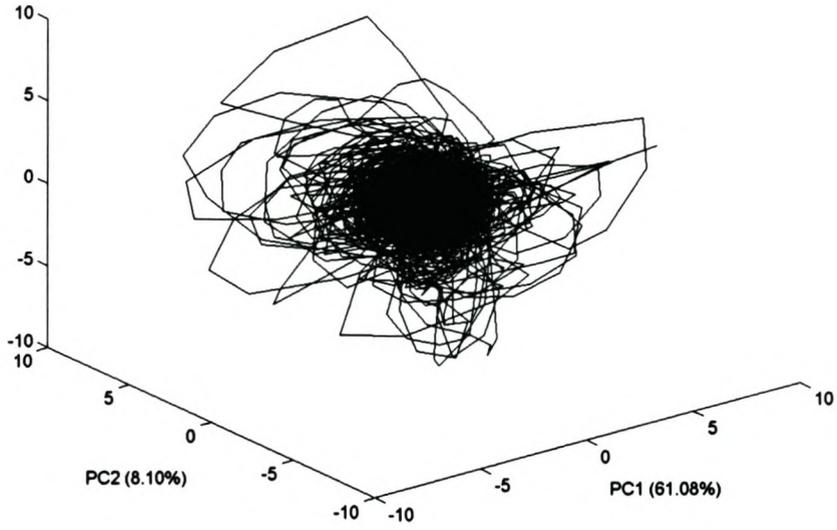
Since the use of SSA for embedding of the time series has thus far proven to be superior to the MOD for the data being analysed, only the use of SSA for the embedding the FAG mill feed size distribution (% -31.5mm;  $x_{13}$ ) and the FAG mill product size distribution (% -75 $\mu$ m;  $x_{14}$ ) will be investigated. Table 5.16 shows the respective variables embedding parameters.

**Table 5.16: Embedding parameters**

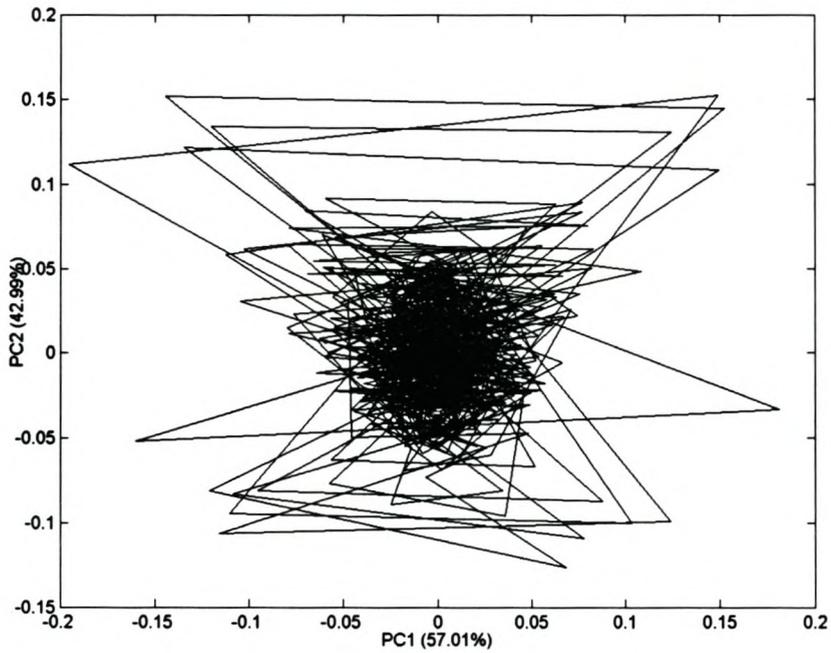
Variable	SSA	
	Embedding Delay	Embedding Dimension
$x_{13}$	1	20
$x_{14}$	1	2

### 5.2.2 Attractor reconstruction

From figures 5.55 and 5.56 it can be seen that the reconstructed attractors of both the FAG mill feed size distribution (% -31.5mm;  $x_{13}$ ) and the FAG mill product size distribution (% -75 $\mu$ m;  $x_{14}$ ) are very complex. Visual comparison of the reconstructed attractors reveals that the attractor for the FAG mill product size distribution (% -75 $\mu$ m;  $x_{14}$ ) is less chaotic than the attractor for the FAG mill feed size distribution (% -31.5mm;  $x_{13}$ ); segregation in the mill feed silo leading to the FAG mill feed size distribution (% -31.5mm;  $x_{13}$ ) being more unpredictable. This is in accordance with the findings of the data classification using the method of surrogate data and further emphasizes the difficulty that could be experienced when trying to model these time series.



**Figure 5.55: Reconstructed attractor of FAG mill feed size distribution (% - 31.5mm;  $x_{13}$ )**



**Figure 5.56: Reconstructed attractor of FAG mill product size distribution (% - 75 $\mu$ m;  $x_{14}$ )**

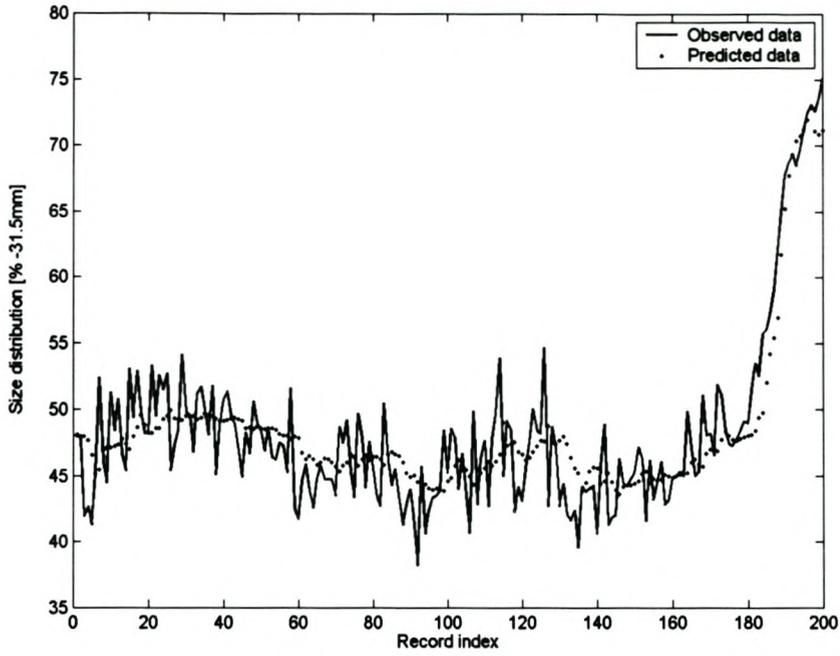
### 5.2.3 Model structure and validation

A separate MLP neural network was constructed for each of the time series with the purpose of predicting only one-step ahead in time. Both networks were trained using the Levenberg-Marquardt algorithm and the number of nodes in the hidden layer estimated using the Schwartz Information Criterion (Schwartz, 1978). A summary of the modelling parameters is given in table 5.17.

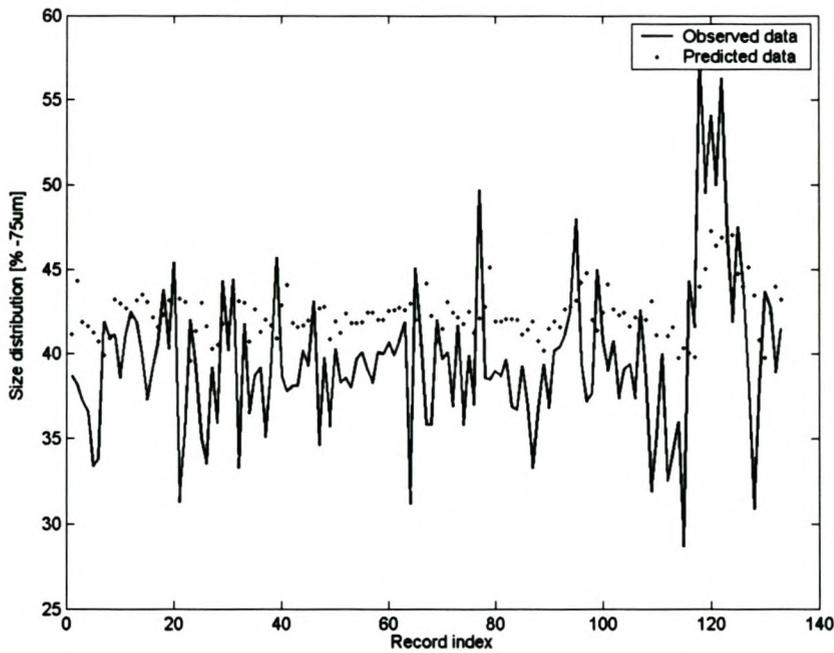
**Table 5.17: Modelling parameters and results**

	<i>wipfrag</i> SSA	<i>malvern</i> SSA
Number of PC's selected	6	2
Variance explained [%]	78.60	100
Number of hidden nodes	7	1
Validation data $R^2$	72.15%	-21.92%

From the modelling results (table 5.17) it would seem that, contrary to expectation, the *wipfragSSA* model managed to be successfully applied for the one-step prediction of the validation data whereas the *malvernSSA* model, as expected, proved to be inadequate for prediction purposes. When considering the one-step prediction graph obtained from *wipfragSSA* (figure 5.57) it is observed that the predicted values only follow the general trend in the data, almost always offset to the right of the actual process data. In contrast to this, the one-step prediction graph obtained from *malvernSSA* (figure 5.58), show no distinct pattern.



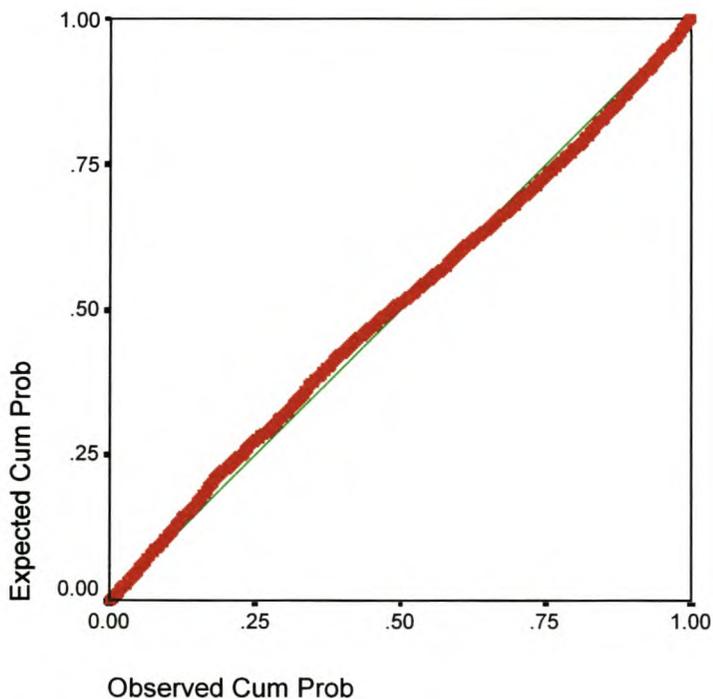
**Figure 5.57: One-step prediction of FAG mill feed size distribution (% -31.5mm;  $x_{13}$ ) ( $R^2 = 72.15\%$ )**



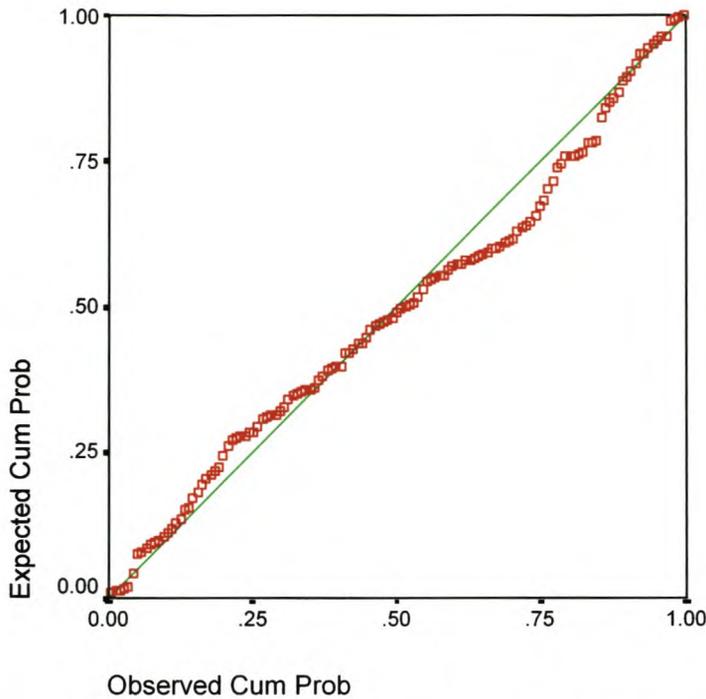
**Figure 5.58: One-step prediction of FAG mill product size distribution (% -75µm;  $x_{14}$ ) ( $R^2 = -21.92\%$ )**

#### 5.2.4 Use of graphical residual analysis for model validation

The normal P-P plots of the residuals obtained from the application of *wipfragSSA* to one-step prediction (figure 5.59) and of *malvernSSA* to one-step prediction (figure 5.60) was constructed. From these figures it can be seen that the *wipfragSSA* model was more successful at extracting information from the data than the *malvernSSA* model. In both instances the models were nevertheless close to the best model possible, given the data. Whereas the residual values obtained from the *wipfragSSA* model only slightly deviated from a straight line, the residuals values obtained from the *malvernSSA* model deviated considerably more from a straight line, a characteristic of non-normally distributed residuals. It would therefore seem that the problem with modelling this data does not lie with the modelling techniques applied, but instead with the data. The fact that the instruments measuring the FAG mill feed size distribution (% -31.5mm;  $x_{13}$ ) and the FAG mill product size distribution (% -75 $\mu$ m;  $x_{14}$ ) experienced a lot of calibration and maintenance problems seems to support this notion. It also shows the importance of “good” quality data when modelling, reminding one that the model is only as good as the data.



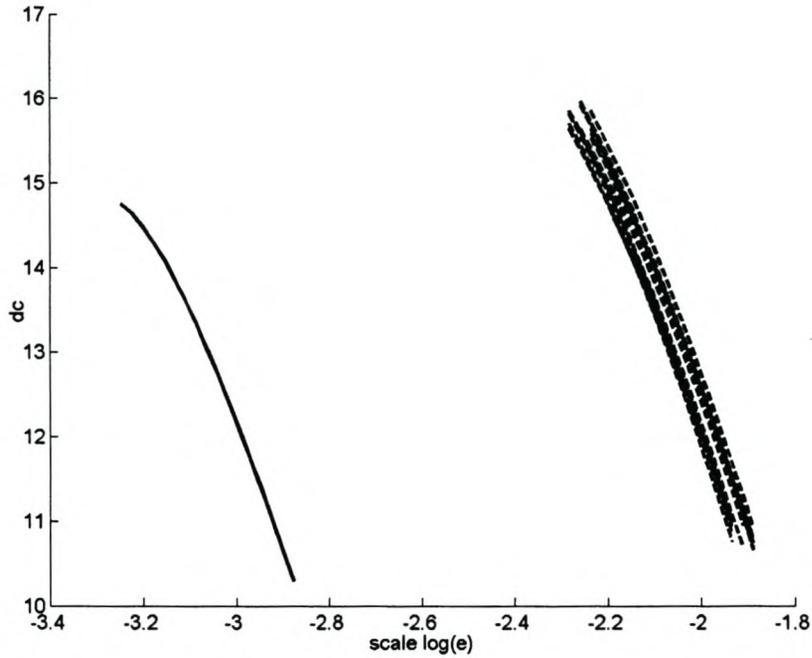
**Figure 5.59: Normal P-P plot of the residual values for one-step prediction of the FAG mill feed size distribution (% -31.5mm;  $x_{13}$ )**



**Figure 5.60: Normal P-P plot of the residual values for one-step prediction of the FAG mill product size distribution (% -75 $\mu$ m;  $x_{14}$ )**

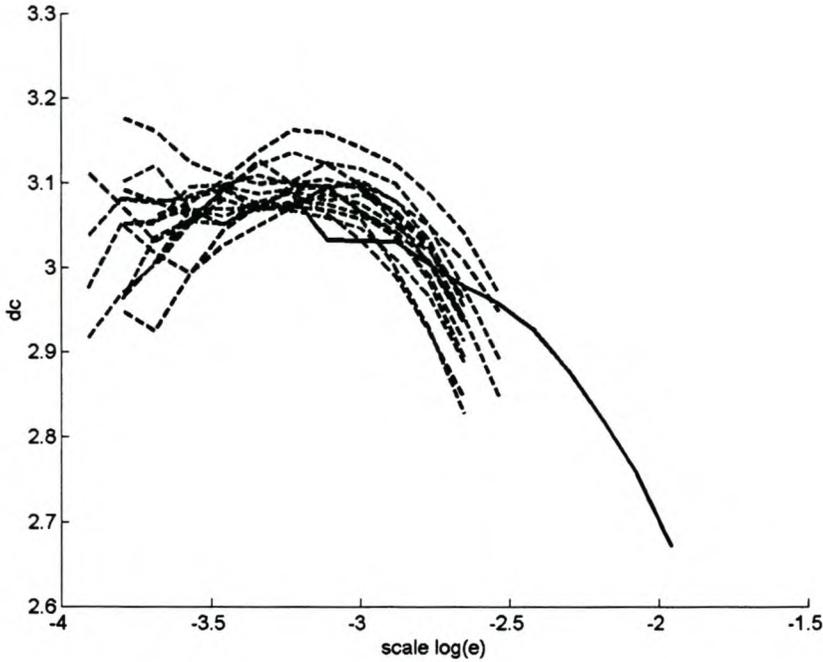
### 5.2.5 Use of surrogate data for model validation

From the correlation dimension curve for the FAG mill feed size distribution (% - 31.5mm;  $x_{13}$ ) and its associated non-linear surrogates (figure 5.61) it would seem as if the model failed to capture any of the underlying dynamics of the system. Although the plots for both the process data and the non-linear surrogates are of similar shape and correlation dimension, it can be seen that both the micro- and macrostructure of the dynamic attractor for the process data is much more complex than that of the non-linear surrogates, as evident from the correlation dimension curve for the process data lying in the  $\log \epsilon_0 < -2.85$  region and that of the non-linear surrogates in the  $\log \epsilon_0 > -2.3$  region. This increased complexity in the structure of the process data's dynamic attractor is a direct consequence of the high degree of noise in the data, which subsequently could not successfully be modelled.



**Figure 5.61: Correlation dimension curve for FAG mill feed size distribution (% -31.5mm;  $x_{13}$ ; solid line) and of the non-linear surrogates (broken lines)**

From figure 5.62 it can be seen that the small-scale region of the dynamic attractor ( $\log \epsilon_0 < -2.55$ ) of the FAG mill product size distribution (% -75 $\mu\text{m}$ ;  $x_{14}$ ) overlaps with the large-scale region ( $\log \epsilon_0 > -3.25$ ) of the non-linear surrogates associated with the *malvernSSA* model. Since a higher correlation dimension obtained at the small-scale region of the dynamic attractor of data is usually caused by dynamic noise (Barnard, 1999), it could be concluded that the model was only successful in modelling the noise contained in the data. This is further confirmed by the fact that the average correlation dimension value for the non-linear surrogates are higher than that of the process data, meaning that the dynamic attractor of the process data is less complex than that of the generated non-linear surrogates.



**Figure 5.62: Correlation dimension curve for FAG mill product size distribution (% -75 $\mu$ m;  $x_{14}$ ; solid line) and of the non-linear surrogates (broken lines)**

### 5.2.6 Free-run prediction

Since the results obtained from the application of the developed models for one-step prediction of the validation data were poor, it was suspected that free-run predictions would be impossible. Before doing the free-run predictions the Lyapunov spectrum was calculated for both the FAG mill feed size distribution (% -31.5mm;  $x_{13}$ ) and the FAG mill product size distribution (% -75 $\mu$ m;  $x_{14}$ ), yielding insight into the predictability of the system. For both systems the 1<sup>st</sup> Lyapunov exponent was positive (table 5.18), indicative of a chaotic system, with the sum of all the Lyapunov exponents in each Lyapunov spectrum being negative, indicative of a bounded, and therefore predictable, attractor.

**Table 5.18: Lyapunov spectrum for the FAG mill power ( $x_1$ ) time series**

	<i>FAG mill feed size distribution</i>	<i>FAG mill product size distribution</i>
1 <sup>st</sup> Lyapunov exponent	$8.06e^{-02}$	$7.74e^{-02}$
Prediction horizon [s]	360	600

Contradictory to the optimistic prediction horizon estimations (table 5.18), free-run prediction of the data proved impossible (found in the appendix). Previously the lack of ability of models to be used for free-run prediction were ascribed to the fact that the models are highly dependent on the initial conditions of the data and the possibility that the time series used for modelling not containing all the information of the system being modelled. However, for this instance it would rather seem as if the Lyapunov exponent is not a good measure of the predictability for these highly noisy systems. This is confirmed by the fact that the FAG mill power ( $x_1$ ) was also only predictable a fraction of the length of the Lyapunov estimate.

### 5.2.7 Results discussion

When considering the above results, it can be concluded that the FAG mill product size distribution (% -75 $\mu$ m;  $x_{14}$ ) and the FAG mill feed size distribution (% -31.5mm;  $x_{13}$ ) data did not allow for accurate predictive modelling, at least not for automated control, which was the ultimate purpose of this thesis. The lack of these variables to successfully be modelled also confirmed the findings in chapter 4, where the FAG mill product size distribution (% -75 $\mu$ m;  $x_{14}$ ) was found to be almost purely stochastic for all practical purposes and the FAG mill feed size distribution (% -31.5mm;  $x_{13}$ ) was found to be stochastic.

Possible solutions to ensure better modelling of the FAG mill product size distribution (% -75 $\mu$ m;  $x_{14}$ ) and the FAG mill feed size distribution (% -31.5mm;  $x_{13}$ ) include:

- Increasing the quality of the data by ensuring that the instrumentation measuring these variables is properly calibrated and maintained.
- Ensuring the availability of more FAG mill product size distribution (% -75 $\mu$ m;  $x_{14}$ ) observations for data analysis and modelling purposes.
- The use of multi-dimensional embedding, making use of other process observations to assist in the modelling of the above-mentioned variables.
- Investigating the use of stochastic modelling techniques (Gradišek et al., 2002; Gradišek et al., 2000; Kantz and Schreiber, 2000; Siegert et al., 1998).

### 5.3 Summary

Based on the modelling results it was found that overall, the MLP neural network models performed the best, with comparable results from the LS-SVM, RBF neural network and linear ARX models. It was also found that the models based on SSA embeddings always outperformed those based on MOD embeddings; this being a direct result from better attractor reconstruction using the SSA embeddings compared to the reconstructed attractors using the MOD embeddings. For the system under investigation it was thus concluded that the better the quality of the attractor reconstruction (based on the embedding strategy) and the more of the state space it covered, the more the attractor was unfolded and the better the quality of the resulting model.

From a one-step ahead prediction perspective, adding new variables as model inputs did not significantly improved the modelling results; the additional variables not presenting any new, useful information for the modelling of the system. However, the best MLP neural network model was found by embedding all the input variables of the system, using the embedding parameters estimated by SSA, and including all the output variables in the multi-dimensional embedding with an embedding delay and embedding dimension of unity. From a control perspective, this model showed the importance of including the input variables of the system when predicting one of the systems output variables. Furthermore, although the *linearARX* model outperformed all the no-embedding models tested, similar results were not obtained for the models where the data was embedded prior to modelling. Embedding of the multi-dimensional time series data prior to modelling also improved the prediction accuracy of the developed models.

For LS-SVM model development it was found that the technique took much longer to produce a useable model compared to when using either MLP or RBF neural networks. However, for the system under investigation, LS-SVMs was successfully used to “quickly” construct a model of the mill to predict one of the mills measured outputs without explicitly selecting the input variables for the model and by fixing the embedding delay at unity and the embedding dimension at the arbitrarily chosen value of 20.

The one-dimensional RBF neural network models were found to be better suited to free-run prediction than either the one-dimensional MLP neural network or LS-SVM models. Using the RBF neural network model based on only the FAG mill power ( $x_1$ ) time series, *RFSSA\_0*, preliminary results indicated that the model could successfully predict in excess of 70% of the time whether or not the FAG mill power ( $x_1$ ) was increasing or decreasing on average over the prediction interval. RBF neural networks therefore seemed to be better at capturing the underlying dynamics in the one-dimensional FAG mill power ( $x_1$ ) time series, for the system under investigation, than either MLP neural networks or LS-SVMs.

From the multi-dimensional time series model free-run predictions it was found that the *linearARX* model performed the worst with the *MLPSSA\_3* model performing the best. For the LS-SVM models it was found that by increasing the input space of the models resulted in better free-run prediction results, this being a direct result from the fact that LS-SVMs scale very well to high dimensional input spaces. The free-run prediction results obtained clearly show the promise of embedding the data prior to non-linear modelling and could possibly be improved upon by using more training data, covering a larger region of the state space, which also more homogeneously populate the state space.

It was also concluded that the data used to model the FAG mill product size distribution (% -75 $\mu$ m;  $x_{14}$ ) and the FAG mill feed size distribution (% -31.5mm;  $x_{13}$ ) was not suited for accurate predictive modelling. Possible solutions to ensure better modelling were suggested, but these have to be qualified though in terms of the ultimate control system to be used.

## 6 On-line model validation

---

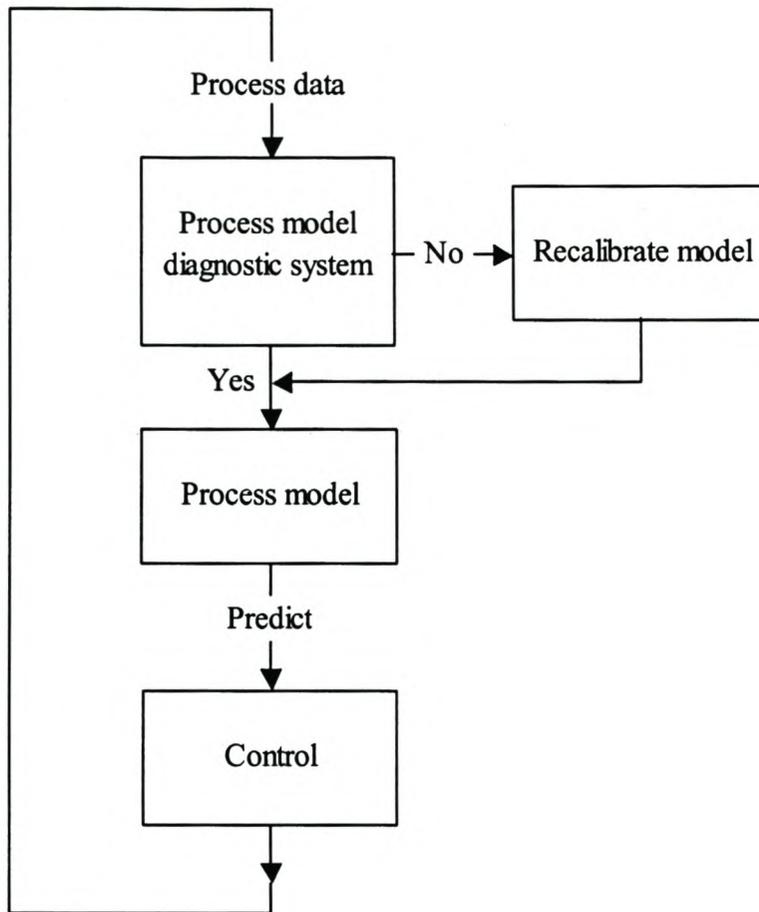
Since any empirical model is only as good as the model training data and only valid for data similar to the model training data, there is a need to validate all new process data to ensure that the new data, when fed to the model, will result in sensible predictions. Additionally, no matter how good the model, it will have to be recalibrated/reconstructed periodically owing to “process drift” or other changes. To do so intelligently, a change in the process has to be detected first. The question is how.

One way of accomplishing this is based on the methodology followed for multivariate statistical process monitoring. Multivariate statistical process monitoring techniques could be used to detect changes in the process based on a statistical model built using historical process data from when the process was normally operated. Once new process data is collected that’s different from the model training data, or the process moves away from its normal operating region, new information becomes available that could be used to increase our knowledge of the process. When used to detect changes in the process with regard to model diagnosis, this new data could be included into the model training dataset and the model retrained (figure 6.1).

### 6.1 Process model diagnostic system

Various statistical methods exist for the analysis, monitoring and diagnosis of process operating performance over time. These methods all depend on a mathematical representation of the process and can take one of two forms: a first-principles model, or an empirical model based on historical process data. Since the development of a first-principles model can be very difficult, the use of empirical models is advantageous when encountering high process non-linearity, high dimensionality or high process complexity. Based on historical process data, these methods are normally used to verify whether or not a process is within a “state of statistical control” while still allowing “common-cause” variation (variation that affects the process all the time and is essentially unavoidable within the current process). Typically a statistical model is built using historical process data when the process was operated normally, with any periods containing variations arising from special

events that one would like to detect in the future are omitted, and new process data is then compared to this model in order to detect a change in the system.



**Figure 6.1: On-line model validation methodology**

In order to detect changes in the system, conventional univariate control charts usually include both warning limits (target value  $\pm 2\sigma$ ) and control limits (target value  $\pm 3\sigma$ ). Using these limits, an out-of-control situation is then determined based on one of the following rules (Nijhuis et al., 1997):

- Any observations outside of the control limits.
- Two or more consecutive observations outside the warning limits.
- Four or more consecutive observations outside the  $1\sigma$  limits.
- Eight or more consecutive observations moving in either an upward or downward direction.
- Any unusual or non-random pattern in the data.

However, since metallurgical processes are often non-linear, essential process information may not necessarily lie within any individual process variable (univariate), but in how the variables change with respect to one another (multivariate).

Conventional, well-established statistical process monitoring (SPM) charts, such as Shewhart, CUSUM (cumulative sum) and EWMA (exponentially weighted moving average) charts, are of an univariate nature. These charts commonly only permit investigation into the magnitude of deviation of any one variable, independently of all others variables, at a given time, often resulting in inaccurate, delayed conclusions being drawn (Kourti and MacGregor, 1995). However, this deficiency is easily remedied using multivariate statistical projection methods such as principal component analysis (PCA).

### **6.1.1 Use of PCA for a process model diagnostic system**

When using PCA all the process variables are treated simultaneously, extracting information on how all the process variables are behaving relative to one another, while concurrently compressing the information in a manner which retains essential information and facilitates the graphical interpretation of the results. PCA is therefore also ideal for handling high dimensional and correlated variables with the added advantage of having a built-in noise-filter.

#### **6.1.1.1 Principal component analysis**

Conventional PCA is used to find linear combinations of the variables in a dataset  $\mathbf{X}$ , consisting of  $n$  rows (the observations) and  $p$  columns (the variables), through eigenvector decomposition of the correlation matrix of the process variables (Wise and Gallagher, 1996). Prior to PCA all the variables in  $\mathbf{X}$  must first be autoscaled relative to one another, so as to avoid having important variables whose magnitudes are small from being overshadowed by less important, but larger magnitude variables. Autoscaling therefore entails the adjustment of the data to zero mean and unit variance by subtracting off the original mean of each column and then by dividing each column by its standard deviation. Once autoscaled, the correlation matrix of  $\mathbf{X}$  can be defined as:

$$\text{corr}(\mathbf{X}) = \frac{\mathbf{X}^T \mathbf{X}}{n-1} \quad (6.1)$$

The dataset  $\mathbf{X}$  is decomposed as the sum of the outer product vectors  $\mathbf{t}_i$  and  $\mathbf{l}_i$  plus a residual matrix  $\mathbf{E}$  :

$$\mathbf{X} = \mathbf{t}_1 \mathbf{l}_1^T + \mathbf{t}_2 \mathbf{l}_2^T + \dots + \mathbf{t}_a \mathbf{l}_a^T + \mathbf{E} \quad (6.2)$$

where  $a$  is less than or equal to the smaller dimension of  $\mathbf{X}$ . The  $\mathbf{t}_i$  vectors are known as scores, containing information on how the observations relate to one another, whereas the  $\mathbf{l}_i$  vectors are eigenvectors of the correlation matrix known as loadings, containing information on how the variables relate to one another:

$$\text{corr}(\mathbf{X}) \mathbf{l}_i = \lambda_i \mathbf{l}_i \quad (6.3)$$

where  $\lambda_i$  is the eigenvalue associated with the eigenvector  $\mathbf{l}_i$ . For any  $\mathbf{t}_i, \mathbf{l}_i$  pair,  $\lambda_i$  is a measure of the amount of variance (information) described by the  $\mathbf{t}_i, \mathbf{l}_i$  pair. The  $\mathbf{t}_i, \mathbf{l}_i$  pairs are also arranged in descending order according to its associated  $\lambda_i$  with the first pair capturing the largest amount of variance of any pair in the decomposition and each subsequent pair capturing the largest possible amount of variance remaining at that step. It should also be noted that the score vector  $\mathbf{t}_i$  is a linear combination of the dataset  $\mathbf{X}$  defined by  $\mathbf{l}_i$  :

$$\mathbf{X} \mathbf{l}_i = \mathbf{t}_i \quad (6.4)$$

Once PCA has been applied to a dataset, it is generally found that the data can be adequately described using far less factors than the original variables and that the score variables produced are more normally distributed than the original variables themselves. It should also be noted that in some instances, if the score values are associated with a controlled property of the system, the scores would not be expected

to be normally distributed. The PCA model can be validated using the Hotellings  $T^2$  statistic, giving an indication as to the variation within the model, and a lack of model fit statistic,  $Q$ .

Since the dynamic behaviour of a process is not being considered when performing conventional PCA on each individual observation separately, its application to continuous process data is inadequate. This is a direct result from the time dependency of a dynamic system as can be seen in the variable histories. In dynamic processes, one is not only concerned with the correlation among variables, but also with the autocorrelation of each variable, therefore requiring dynamic models for process monitoring. Using time lagged variable matrices, PCA can be used to detect changes in the input/output relationship of dynamic systems. When applied to a lagged variable matrix, PCA has the effect of removing the major dynamics from the system, resulting in residuals that are much better behaved (uncorrelated) than would otherwise be the case (Wise and Gallagher, 1996). Once estimated, the PCA model can be validated through calculation of the  $T^2$  and  $Q$  statistics.

#### 6.1.1.2 $T^2$ -chart

The  $T^2$ -chart is a multivariate chart based on Hotellings  $T^2$  statistic which, as mentioned previously, gives an indication as to the variation within the PCA model. This chart is plotted based on the first  $a$  principal components (Kourti and MacGregor, 1995), where

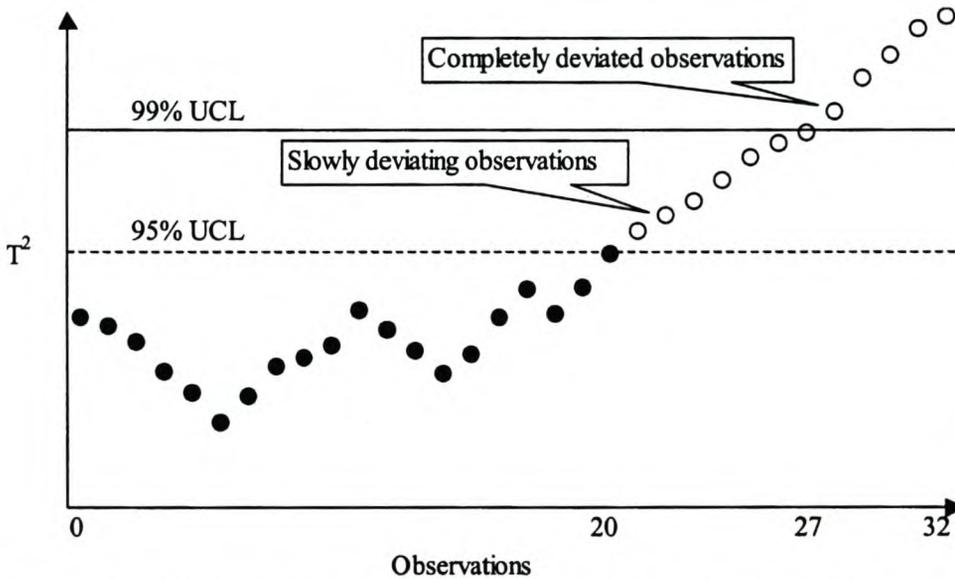
$$T_a^2 = \sum_{i=1}^a \frac{\mathbf{t}_i^2}{s_{t_i}^2} \quad (6.5)$$

and  $s_{t_i}^2$  is the estimated variance of  $\mathbf{t}_i$  according to the principal component model based on historical data. By scaling each  $\mathbf{t}_i^2$  by the reciprocal of its associated variance, each principal component term plays an equal role in the computation of  $T^2$ , irrespective of the amount of variance it explains in the dataset  $\mathbf{X}$ .

For the  $T^2$ -chart the upper confidence limit (UCL), based on historical data, is given by (Kourti and MacGregor, 1995):

$$T_{UCL}^2 = \frac{(n-1)(n+1)a}{a(n-a)} F_{\alpha}(a, n-a) \tag{6.6}$$

where  $n$  is the number of observations in the historical dataset,  $a$  is the number of principal components retained in the PCA model and  $F_{\alpha}(a, n-a)$  is the upper  $100\alpha\%$  critical point of the  $F$ -distribution with  $a$  and  $n-a$  degrees of freedom. For this investigation the 95% UCL is calculated using  $\alpha = 0.05$  and the 99% UCL is calculated using  $\alpha = 0.01$ .



**Figure 6.2:  $T^2$ -chart indicating abnormal process behaviour (Kharva, 2001)**

The  $T^2$ -chart (a typical example of which is shown in figure 6.2) provides a test for deviations in the variables that are of greatest importance to the variance of the dataset  $X$ . However, using Hotellings  $T^2$  statistic will only detect whether or not the variation in the variables in the plane of the first  $a$  PCs are greater than can be explained by “common-cause”. Should the (internal) relationships between the variables change, but the variables are still within the same min-max limits as the reference dataset used to develop the PCA model, new PCs will appear and the new observations will move away from the hyperplane defined by the reference PCA model. This problem can be overcome using a SPE-chart.

### 6.1.1.3 SPE-chart

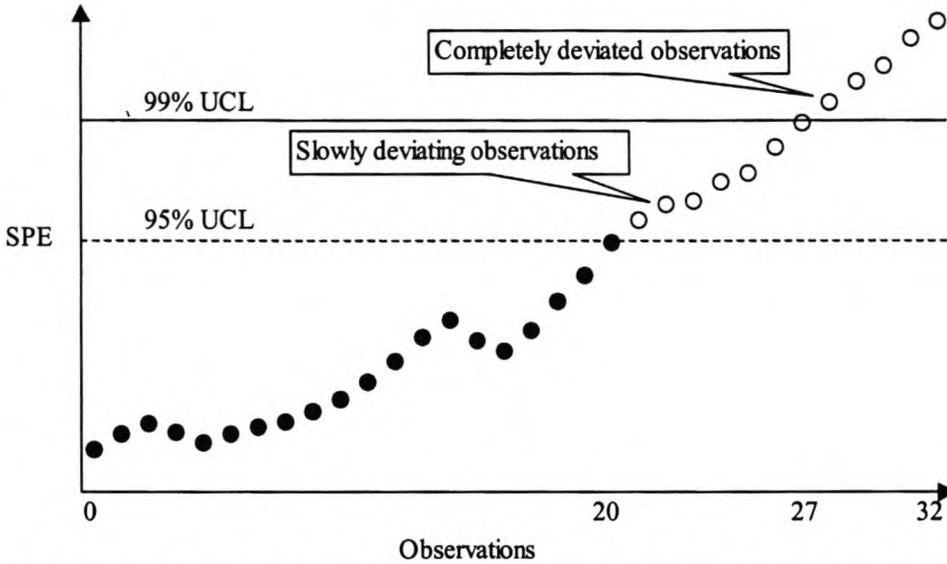
Based on the  $Q$  statistic, indicating a lack of model fit, the Squared Prediction Error ( $SPE$ ) chart is one of two multivariate charts being investigated for the validation of the process models. The SPE-chart is used as a measure of the amount of variation in each observation not captured by the  $a$  principal components retained in the PCA model by calculating the SPE of the residuals of new observations (Kourti and MacGregor, 1995):

$$SPE_x = \sum_{i=1}^q (x_{new,i} - \hat{x}_{new,i})^2 \quad (6.7)$$

where  $x_{new,i}$  is a PCA input, and  $\hat{x}_{new,i}$  is the prediction of  $x_{new,i}$  from the PCA model. For the SPE-chart the UCL, based on historical data, is given by (Nomikos and MacGregor, 1995):

$$SPE_{UCL} = g\chi_{h,\alpha}^2 \quad (6.8)$$

where  $g = \nu/2m$ , the weight of the weighted chi-squared distribution, and  $h = 2m^2/\nu$ , the degrees of freedom of the weighted chi-squared distribution, with  $m$  the mean and  $\nu$  the variance of the SPE historical dataset, at significance level  $\alpha$ . However, unlike the application by Nomikos and MacGregor (1995) where a smoothing window was used and a different UCL was calculated for each time interval, the historical dataset  $X$  was used as reference, and a fixed UCL was determined. SPE confidence limits can be established for the residuals of the dataset  $X$  or for the residuals of the individual variables. Analogous to the  $T^2$ -control chart the 95% UCL is calculated using  $\alpha = 0.05$  and the 99% UCL is calculated using  $\alpha = 0.01$ .



**Figure 6.3: SPE-chart indicating abnormal process behaviour (Kharva, 2001)**

The SPE-chart (a typical example of which is shown in figure 6.3) indicates abnormalities resulting from when the relationship among variables or their dynamic patterns change. When this happens, the SPE will increase since the PCA model can no longer explain the new correlation among the measurements.

#### 6.1.1.4 Score-chart

Similar to the SPE-chart, the score-control chart is a multivariate chart that could be used for the validation of process models. The score-chart is constructed by plotting the scores,  $t_i$ , obtained from applying PCA to the dataset  $X$  in either two or three dimensions. If more than three principal components were retained in the PCA model, plotting various combinations of  $t_i$  could result in valuable process information being extracted.

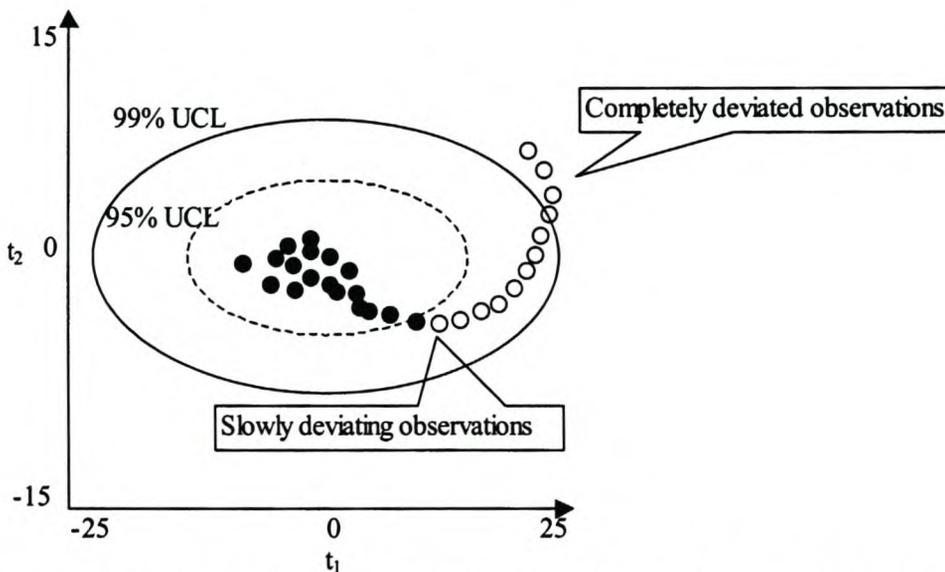
For score-charts, as with the other charts, confidence limits can be placed on the process scores. However, since the scores are related to the states of the process, they are typically quite correlated in time and therefore cannot generally be assumed to be random and normally distributed. Wise and Gallagher (1996) suggested that the confidence limits on the process scores be established based either on judgement concerning the desired process operating limits or using more sophisticated time series modelling techniques. In this investigation, time lagged variables are used,

taking the dynamics of the system into account, and allowing the use of the  $T$ -distribution for UCL determination (assuming the data is normally distributed).

For the score-chart the UCL, based on historical data, is given by:

$$T_{UCL} = \sigma_i \hat{t}_{\alpha/2, n-1} \quad (6.9)$$

where  $\sigma_i$  is the standard deviation of the score  $t_i$  and  $\hat{t}_{\alpha/2}$  is the upper  $\alpha/2$  point of the  $T$ -distribution with  $n-1$  degrees of freedom. As before the 95% UCL is calculated using  $\alpha = 0.05$  and the 99% UCL is calculated using  $\alpha = 0.01$ .



**Figure 6.4: Score-control chart indicating abnormal process behaviour**

The score-chart (a typical example of which is shown in figure 6.4) indicates abnormalities resulting from when the basic relationship among the variables and their dynamic patterns do not change but when several variables have a larger than normal change. When this happens, the SPE will remain small while the principal scores will move outside the region over which the PCA model was developed. A combination of the score-chart and the SPE-chart is also possible.

#### 6.1.1.5 PCA model development methodology

The methodology to be followed for the validation of new process data to be used on an existing empirical model is similar to that used by Chen and McAvoy (1998):

1. Data collection. Since the PCA model developed will only be applicable to the conditions under which the data are gathered, the dataset upon which the PCA model will be based should be chosen to satisfy the needs of the intended application. In the case of data validation, the collection of the model training data, and in the case of process monitoring, the collection of normal operating data encompassing low and high throughputs of process variables yielding desired product quality outputs. The normal operating data should be cleared from abnormal situations, taking into account that statistically 5% of the data exceeds the  $2\sigma$  warning limits as determined for univariate Shewhart charts. Should more than 5% of the data exceed these warning limits, they should be treated as abnormal situations and removed from the dataset (Nijhuis et al., 1997).
2. Data pre-treatment. All the data is autoscaled and possible outliers eliminated. Should it be deemed necessary, dead times between variables can be accounted for by time shifting, whereas the dynamics of the process and the disturbances can be accounted for using time-lagged values.
3. The development of the PCA model. Linear PCA or non-linear PCA is applied to the autoscaled data matrix. For monitoring purposes, a linear PCA model is usually sufficient to describe process fluctuations around an operating point whereas for inferential purposes, covering a wide range of operating regions, a non-linear PCA model may be required (MacGregor and Kourti, 1995).
4. The development of  $T^2$ -, SPE- and score-charts and their associated confidence limits. These plots are complementary in detecting special events, each having its unique advantages.

For large datasets it is important to first find out where in the data useful information can be found (Kourti and MacGregor, 1995) e.g. should clusters form when constructing score-control charts, the important information is usually found in the transitions between the various clusters. Once this is known the reasons for these shifts in process operation can be found by interrogating the underlying multivariate

model through contribution plots which reveal the groups of process variables making the greatest contributions to the deviations in the  $SPE_y$  and the scores.

Another important consideration is the number of principal components to retain. In order to ensure ease of graphical interpretation this means restricting this number to three principal components that can be represented on a three-dimensional score-chart and a separate SPE-chart. Should 3 principal components be insufficient to describe the bulk of the variation in the data, Kresta et al. (1991) suggests the following: dividing the system into logical modular sections to be monitored separately, increasing the number of charts to cope with additional principal components, or sticking with only three latent vectors where the variation not captured by them will show up in the SPE-chart.

## **6.2 Validating new process data to be used on an existing empirical model**

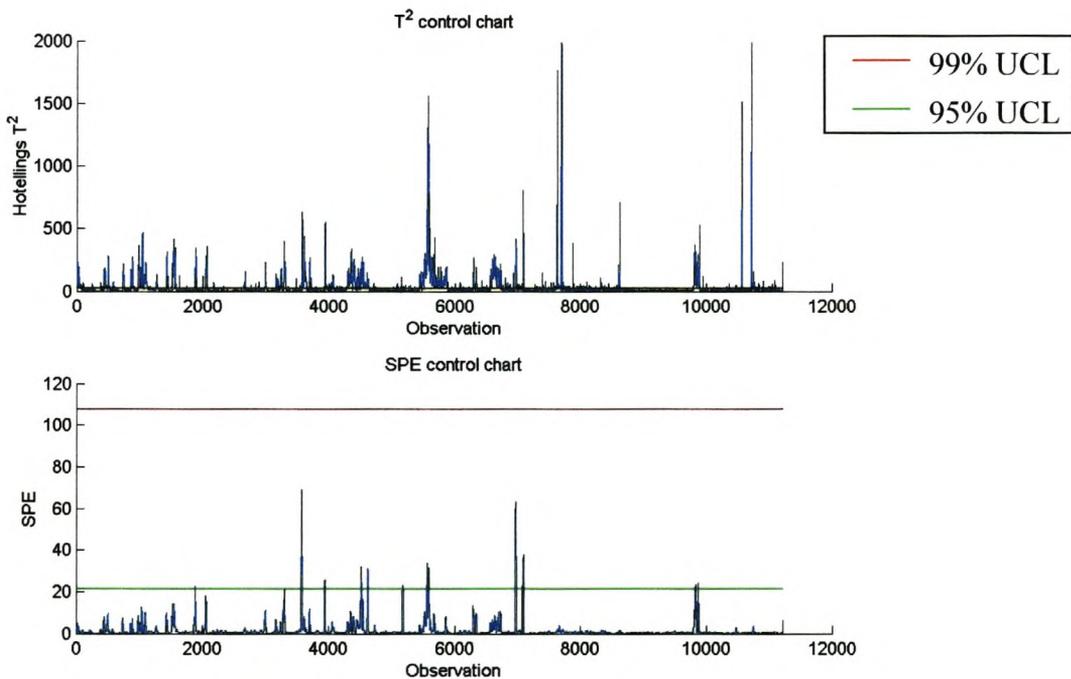
As has already been mentioned, any empirical model is only valid for data similar to that on which it was trained. The techniques employed for process model diagnosis through PCA (as discussed above) are analogous to those used for state space reconstruction (as discussed in chapter 3), making it ideal for the validation of new process data to be used on the models developed in chapter 5. The main objective of this type of data validation is to ensure that any new process data falls within the same state space as that of the data used for model training, thus indirectly also validating the model. Although the technique would be generally applicable, the *MLPSSA\_3* model developed in chapter 5 will be used as a case study.

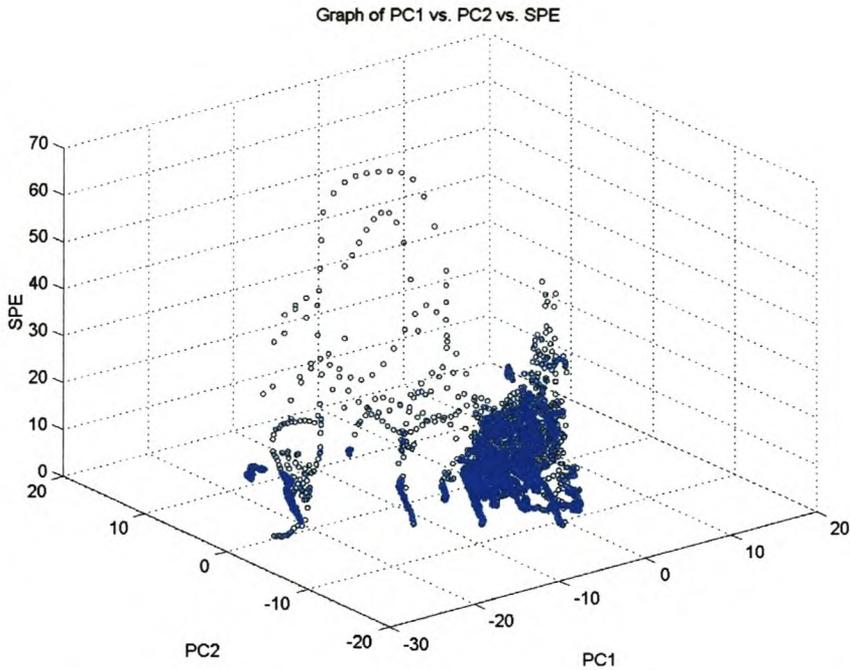
The PCA model to represent the system was constructed using both the training and validation data used for the development of the *MLPSSA\_3* model. The data was autoscaled, embedded and PCA applied to calculate the PCA weight matrix, all done equivalent to that during model development. All the confidence limits for the respective charts (table 6.1) were determined using only the model training data, excluding the validation data, this again being equivalent to that done during model development.

**Table 6.1: Upper confidence limits for respective charts**

Chart	95% UCL	99% UCL
T <sup>2</sup> -chart	25.04	30.64
SPE-chart	21.45	107.79

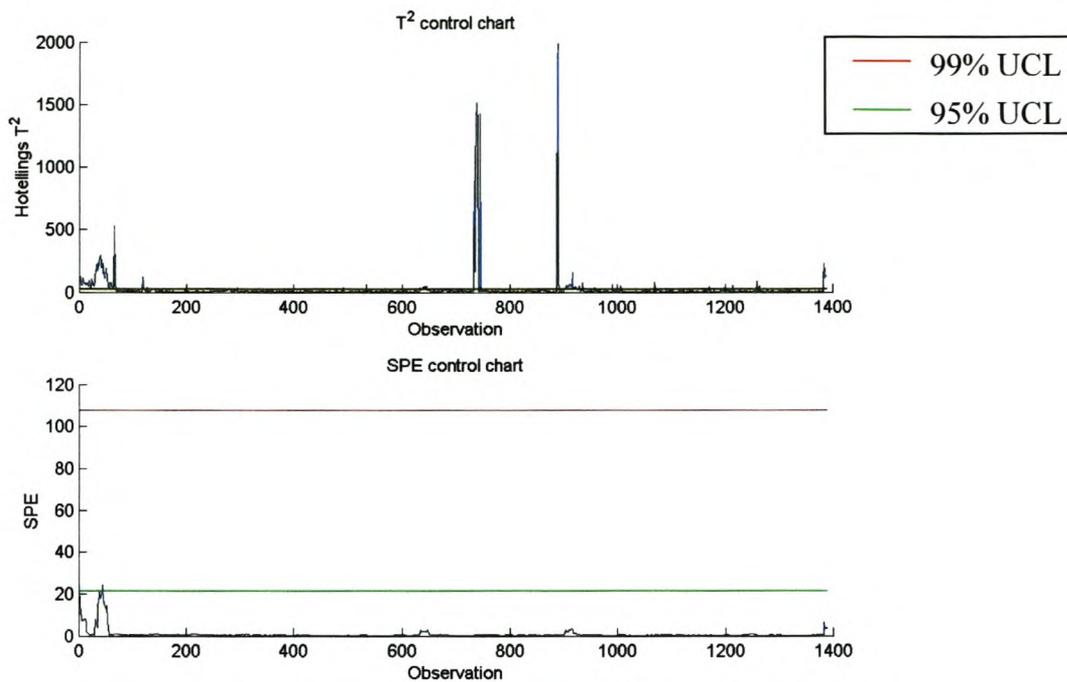
It should be noted at this stage that since 15 principal components were selected for the model development, the graphical interpretation of the score-charts are made extremely difficult and only the plot of the first 2 principal components will be considered.

**Figure 6.5: T<sup>2</sup>- and SPE-charts for model training data**

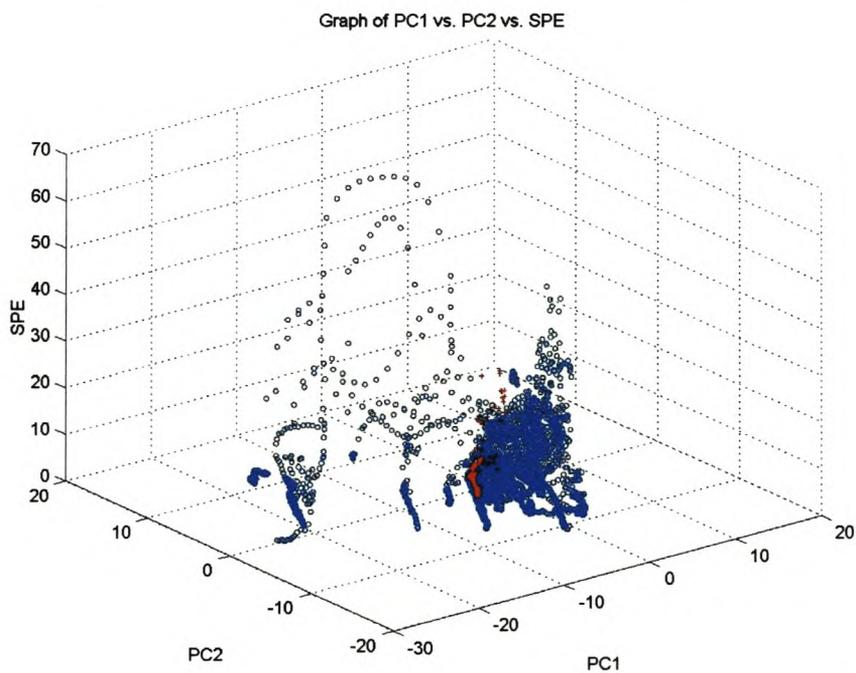


**Figure 6.6: Combined score- and SPE-chart for model training data**

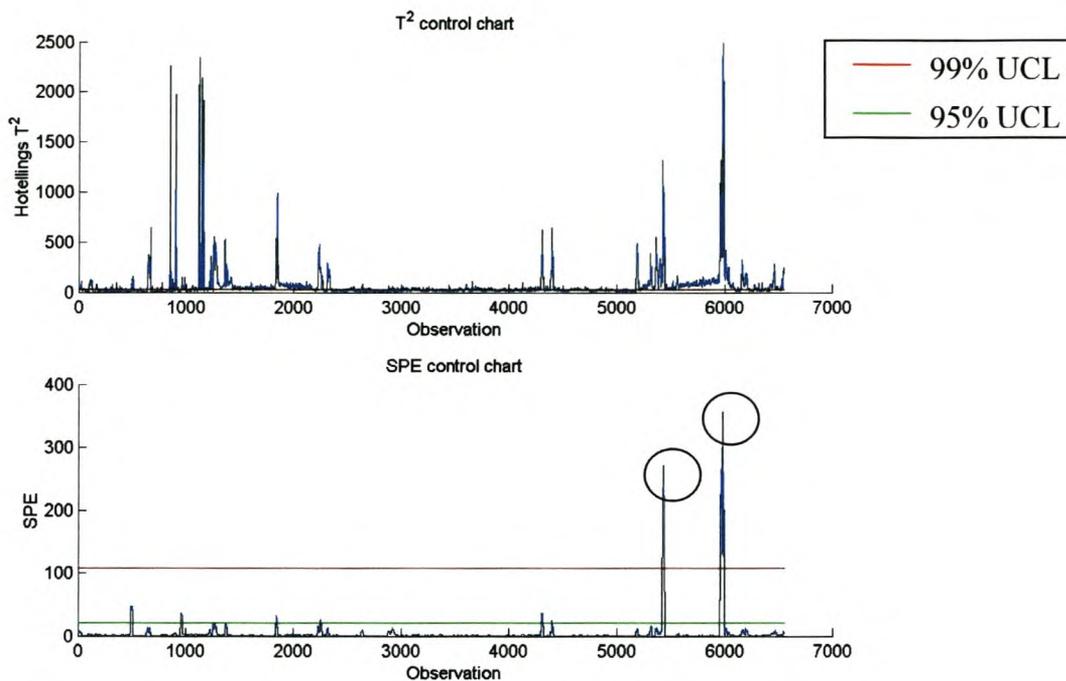
When considering the  $T^2$ -chart (figure 6.5) it would seem as if for the model training data, which was used to estimate the UCL for this chart, a considerable amount of the observations lie outside the UCL, indicating that the data is invalid. This is, however, not the case but a direct result from way in which  $T_a^2$  is calculated (equation 6.5). As soon as a principal component is retained with an estimated variance of less than 1, the value of  $T_a^2$  is increased dramatically. This results from the fact that equation 6.5 relies on the reciprocal of the squared estimated variance and prohibits any reasonable conclusions to be drawn from the chart when these principal components are included in the PCA model. Compared to this, reasonable charts were obtained from both the SPE and the scores (figure 6.6).



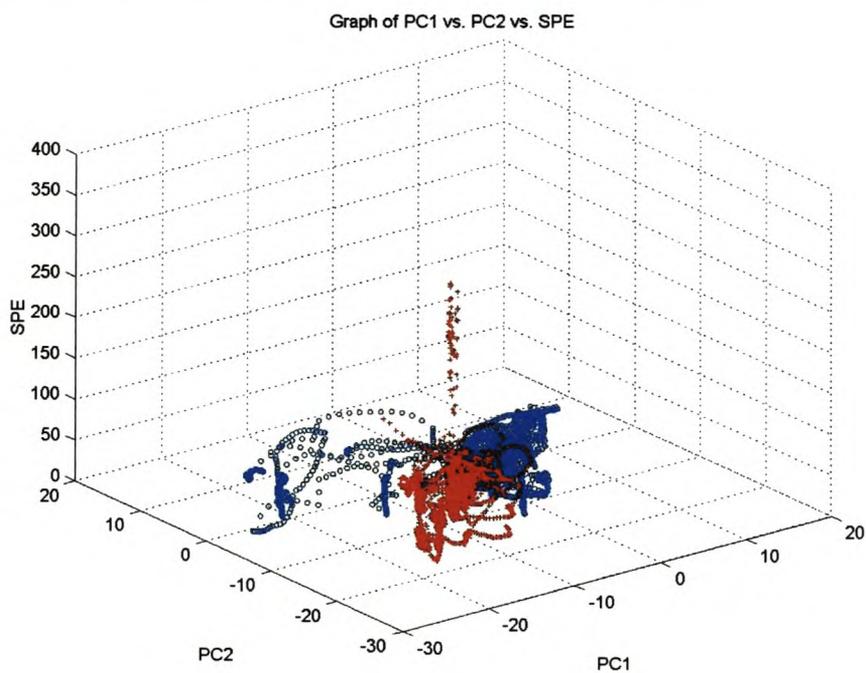
**Figure 6.7: T<sup>2</sup>- and SPE-charts for model validation data**



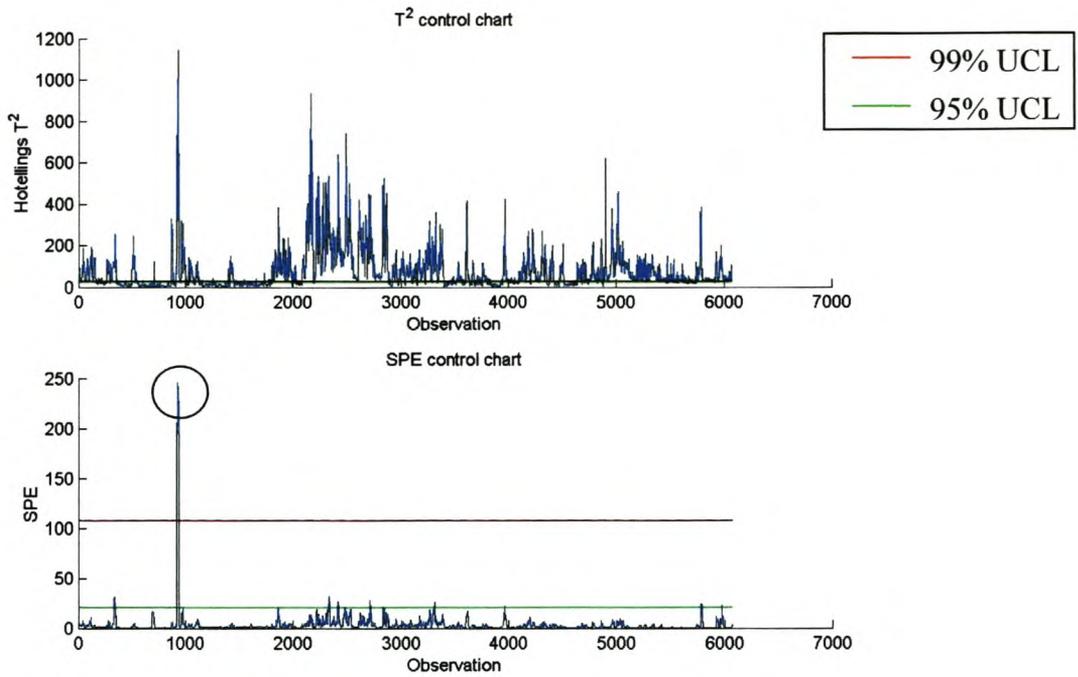
**Figure 6.8: Combined score- and SPE-chart for model validation data (red crosses) in comparison to the model training data (blue circles)**



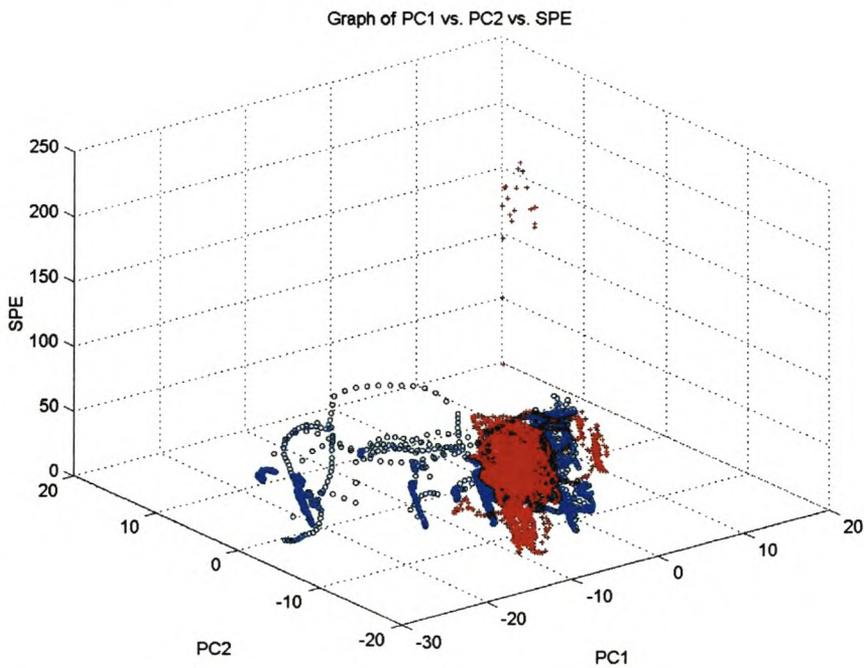
**Figure 6.9:  $T^2$ - and SPE-charts for model test dataset 1**



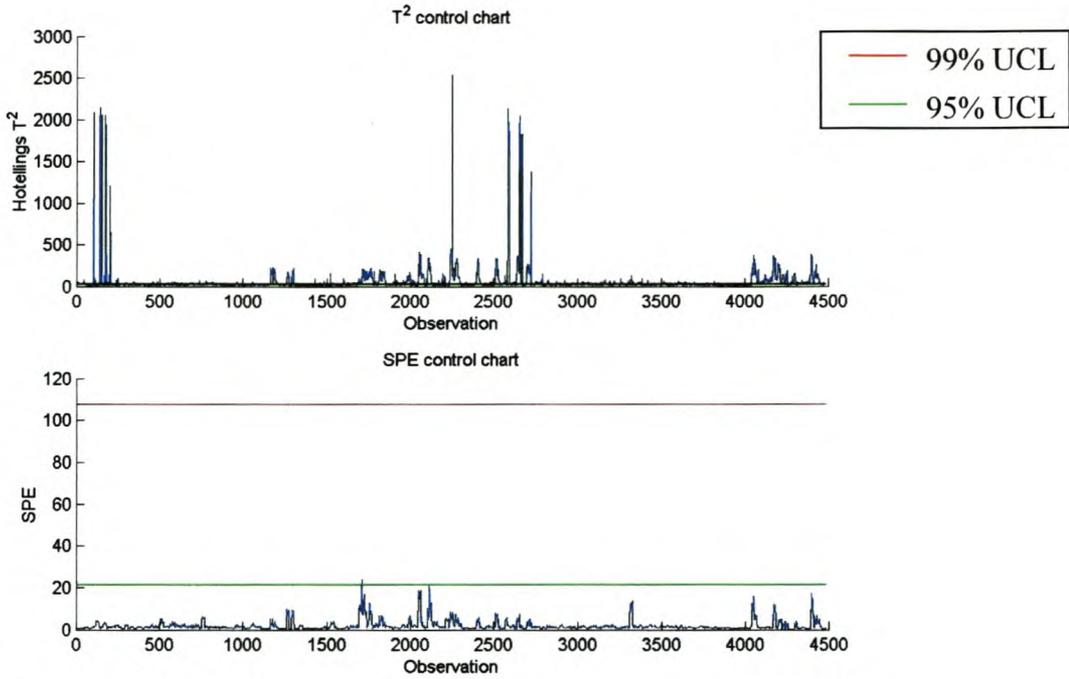
**Figure 6.10: Combined score- and SPE-chart for model test dataset 1 (red crosses) in comparison to the model training data (blue circles)**



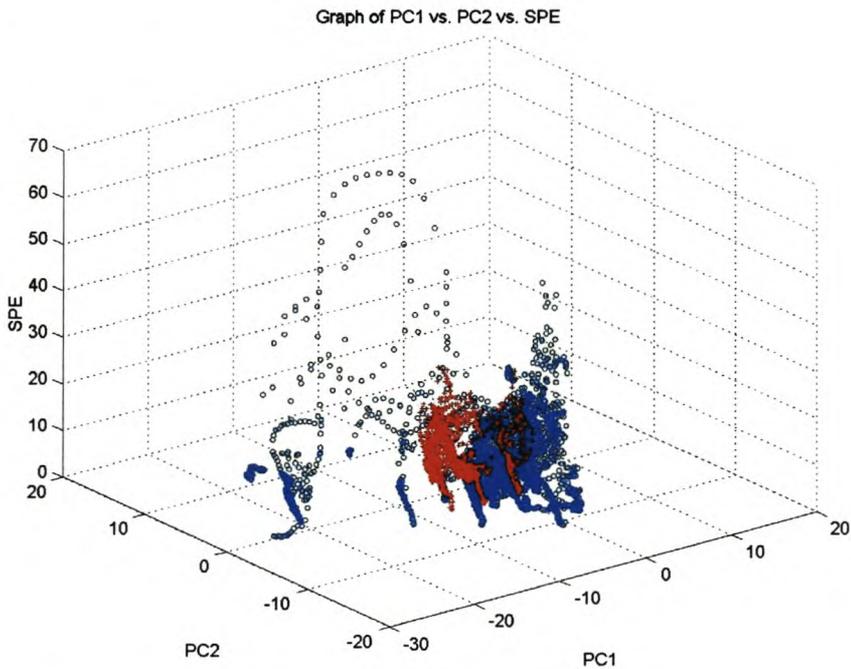
**Figure 6.11:  $T^2$ - and SPE-charts for model test dataset 2**



**Figure 6.12: Combined score- and SPE-chart for model test dataset 2 (red crosses) in comparison to the model training data (blue circles)**



**Figure 6.13: T<sup>2</sup>- and SPE-charts for model test dataset 3**



**Figure 6.14: Combined score- and SPE-chart for model test dataset 3 (red crosses) in comparison to the model training data (blue circles)**

From the SPE-charts for the validation and test data (figures 6.7, 6.9, 6.11 and 6.13) it is evident that the reconstructions are well within the estimated upper confidence limits, exceeding the 99% UCL in only 3 instances and staying well within the 95%

UCL for the majority of the observations. This indicates that very few changes occurred in the relationships among the variables or their dynamic patterns relative to the model training dataset. However, the 3 instances where the 99% UCL was exceeded could be considered special events not present in the model training dataset.

Tracing back, all three special events could be ascribed to a disturbance in the FAG mill inlet water flow rate ( $x_7$ ), causing a change in the FAG mill power ( $x_1$ ), the FAG mill load ( $x_2$ ), the FAG mill discharge 1 flow rate ( $x_9$ ), and the FAG mill discharge 2 flow rate ( $x_{10}$ ). Whenever a special event occurs the model relying on the data containing the special event become invalid and any results obtained from it, doubtful. When found, these special events should be included in the model training dataset, and the model retrained.

When considering the combined score- and SPE-charts for the validation and test data (figures 6.8, 6.10, 6.12 and 6.14) it can be seen that the greater part of the observations fall within the same range as that of the model training data. This indicates that very few instances arose where several variables had a larger than normal change relative to the model training dataset. Since the data being analysed is not normally distributed, the confidence limits as calculated using equation 6.9 is invalid and not displayed on any of the score-charts. The usefulness of these charts will be shown in more detail when applied to process monitoring.

### **6.3 Application to process monitoring**

When applying PCA for process monitoring the most important consideration is the selection of the reference dataset containing the normal operating process data. The reference dataset, containing data for the period 16/03/2001 – 17/03/2001, was selected from the main Process dataset. Three different test datasets, each containing data deviating from the reference dataset, were also selected from the main Process dataset; the details pertaining these test datasets being given in the appendices.

Due to the difficulty experienced with obtaining good normal operating process data, the reference dataset was selected to contain observations only for the following variables: FAG mill power ( $x_1$ ), FAG mill load ( $x_2$ ), FAG mill fine feed rate ( $x_4$ ) and FAG mill inlet water flow rate ( $x_7$ ). During the specified period this dataset looked to

best represent a stable, “in-control” process, with the exception that the process was not within the desired operating parameters as specified in chapter 2. It should also be noted that initially the FAG mill course feed rate ( $x_3$ ) was also included in the dataset. However, since the inclusion of this variable in the reference dataset disrupted all process monitoring results, resulting from the fact that there was no deviation in this variable for the data collection period, it was excluded from the final reference dataset.

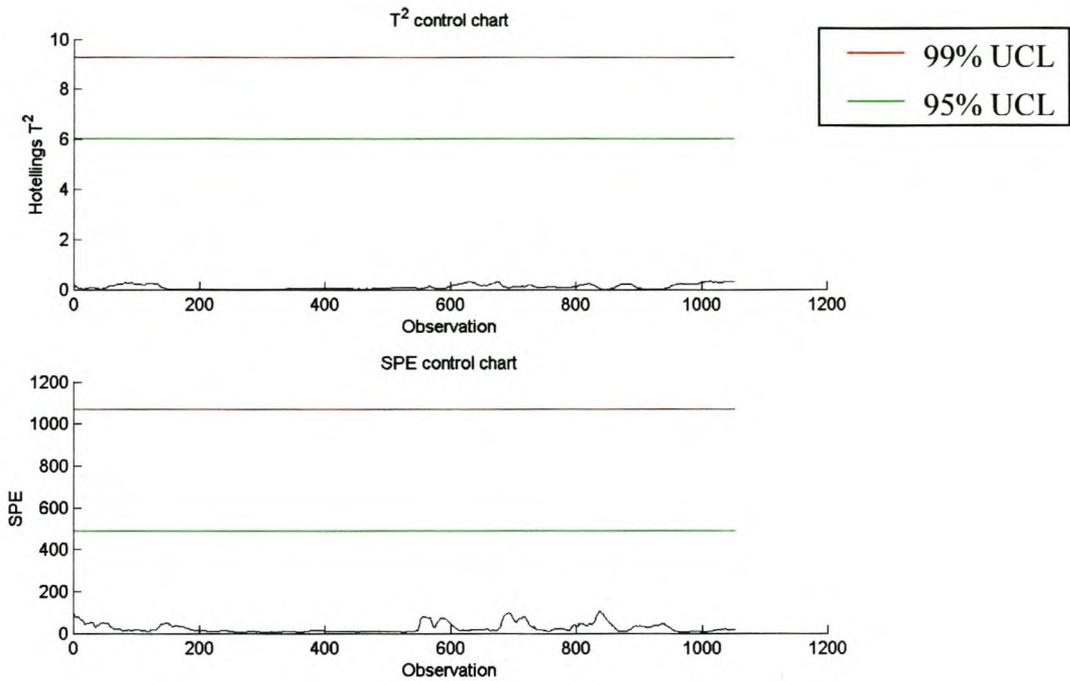
The four variables contained in the reference dataset were also checked to ensure that they conform to the specification of 5% of the observations within the  $2\sigma$  warning limits for each variable as specified by Nijhuis et al. (1997). The respective percentages of observations outside these warning limits for the variables were 4.95%, 0%, 2.71% and 4.76%. Once confirmed that this specification was met, all the variables were autoscaled relative to one another, avoiding having important variables whose magnitudes are small from being overshadowed by less important, but larger magnitude variables.

Each variable in the reference dataset was embedded with an arbitrarily chosen embedding dimension ( $m$ ) of 20 observations and a embedding delay ( $k$ ) of 1. Linear PCA applied to this new matrix. Only the first two principal components were selected to represent the PCA model. This was done so as to ensure the reliable graphical interpretation of results. The first principal component was found to explain 52.26% of the variance in the data with the second principal component explaining 15.94% of the variance in the data. Using this PCA model, the upper confidence limits for the respective charts (table 6.2) were estimated.

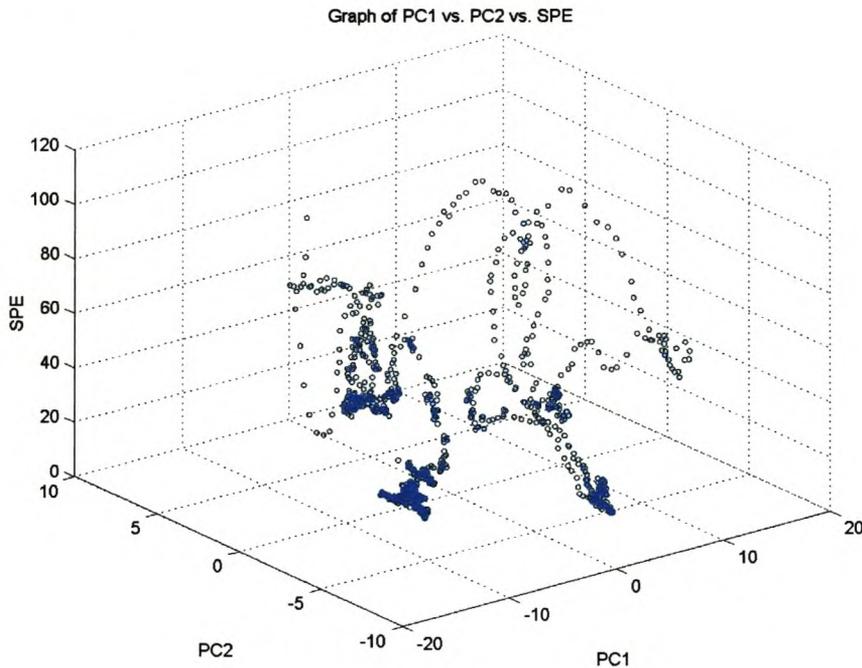
**Table 6.2: Upper confidence limits for respective charts**

Chart	95% UCL	99% UCL
T <sup>2</sup> -chart	6.02	9.27
SPE-chart	490.85	1071.10

Upper confidence limits were also calculated for the individual variables and are shown on the respective charts.



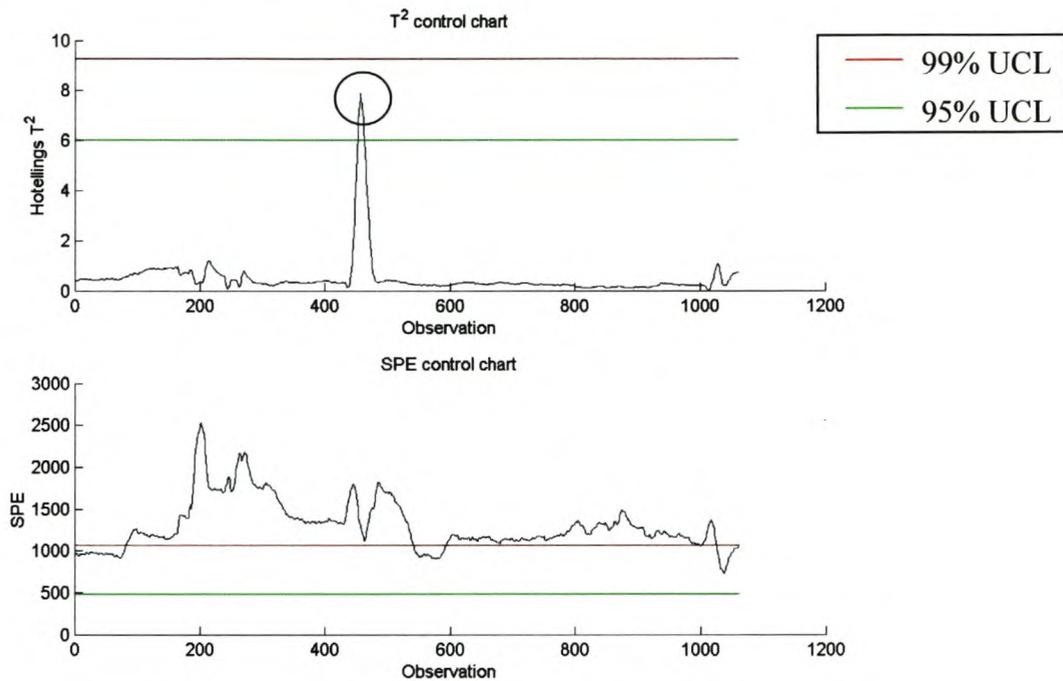
**Figure 6.15:  $T^2$ - and SPE-charts for the reference dataset**



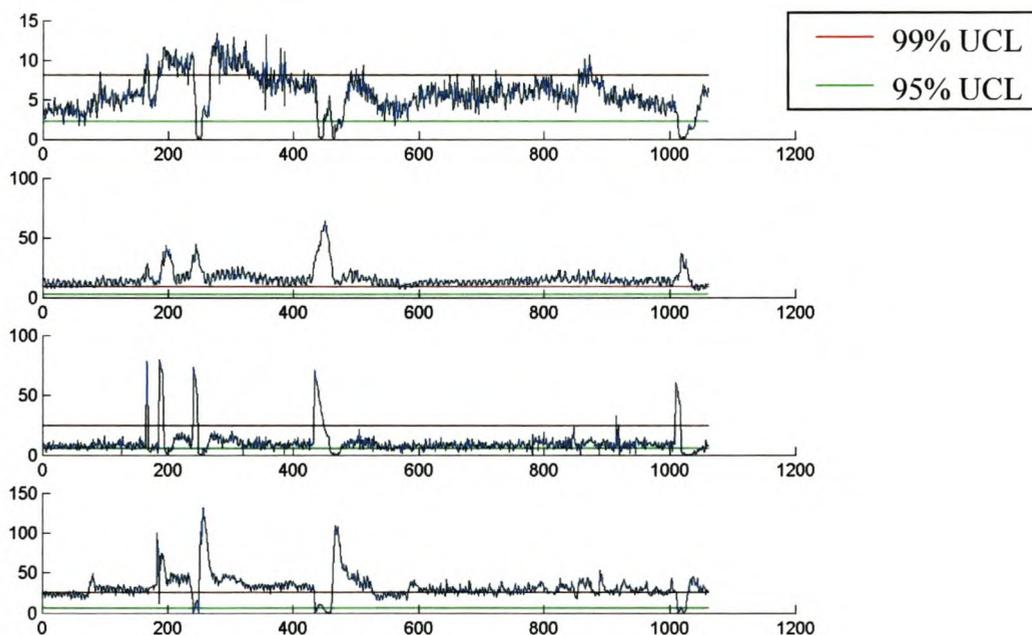
**Figure 6.16: Combined score- and SPE-chart for the reference dataset**

Figures 6.15 and 6.16 shows the charts for the reference dataset. It can clearly be seen that the problems experienced with the  $T^2$ -chart in the previous section have been

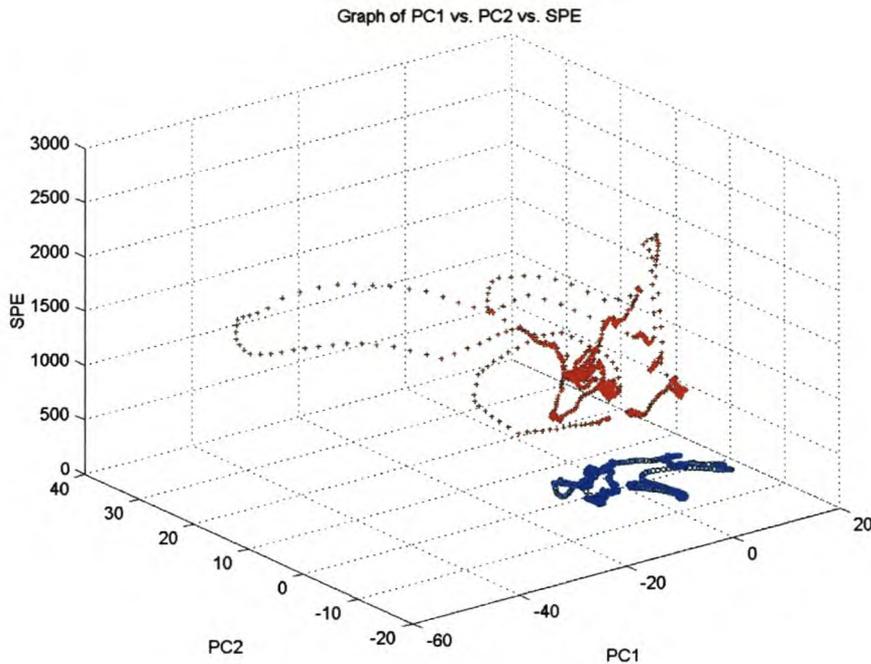
solved through the selection of less principal components to represent the PCA model. From both the  $T^2$ - and SPE-charts it is also evident that all the observations lie well within the estimated 95% UCL.



**Figure 6.17:  $T^2$ - and SPE-charts for test dataset 1**



**Figure 6.18: Individual variable SPE-charts for test dataset 1; FAG mill power ( $x_1$ ), FAG mill load ( $x_2$ ), FAG mill fine feed rate ( $x_4$ ) and FAG mill inlet water flow rate ( $x_7$ )**

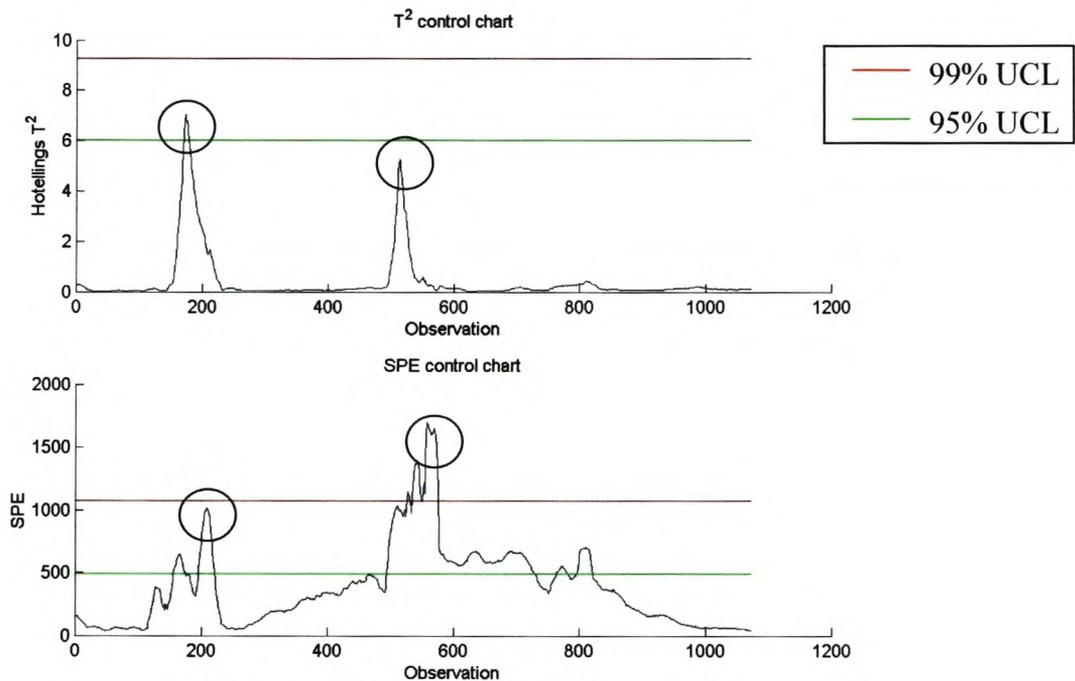


**Figure 6.19: Combined score- and SPE-chart for test dataset 1 (red crosses) in comparison to the reference dataset (blue circles)**

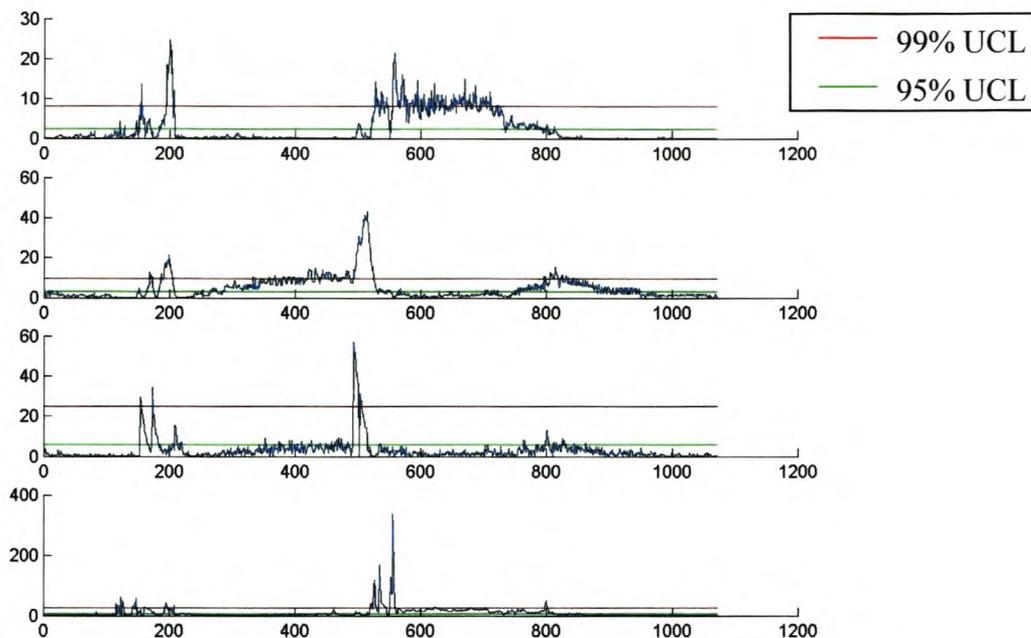
From the  $T^2$ -chart for test dataset 1 (figure 6.17) it can be seen that only one instance existed where the 95% UCL was exceeded, with no instances exceeding the 99% UCL. When traced back, it was found that the instance was caused by the FAG mill fine feed rate ( $x_4$ ) sharply dropping to zero, resulting in a sharp drop in FAG mill inlet water flow rate ( $x_7$ ), in turn causing the FAG mill load ( $x_2$ ) and FAG mill power ( $x_1$ ) to decrease.

From the SPE-chart for test dataset 1 (figure 6.17) it can be seen that the SPE is for the most part above the 99% UCL. Tracing this back to the individual variables (figure 6.18) indicated that the SPE for the majority of the FAG mill load ( $x_2$ ) observations were above its 99% UCL, the majority of the FAG mill inlet water ( $x_7$ ) observations were around its 99% UCL, the FAG mill power ( $x_1$ ) observations were between its 95% and 99% UCL, and the FAG mill fine feed rate ( $x_4$ ) were around its 95% UCL. These findings are a direct results from the fact that test dataset 1 differs immensely from the reference dataset; only for the FAG mill fine feed rate ( $x_4$ ) does the region contained by  $\pm 1\sigma$  around the mean of this time series overlap with the similar region around the time series of the same variable in the reference dataset.

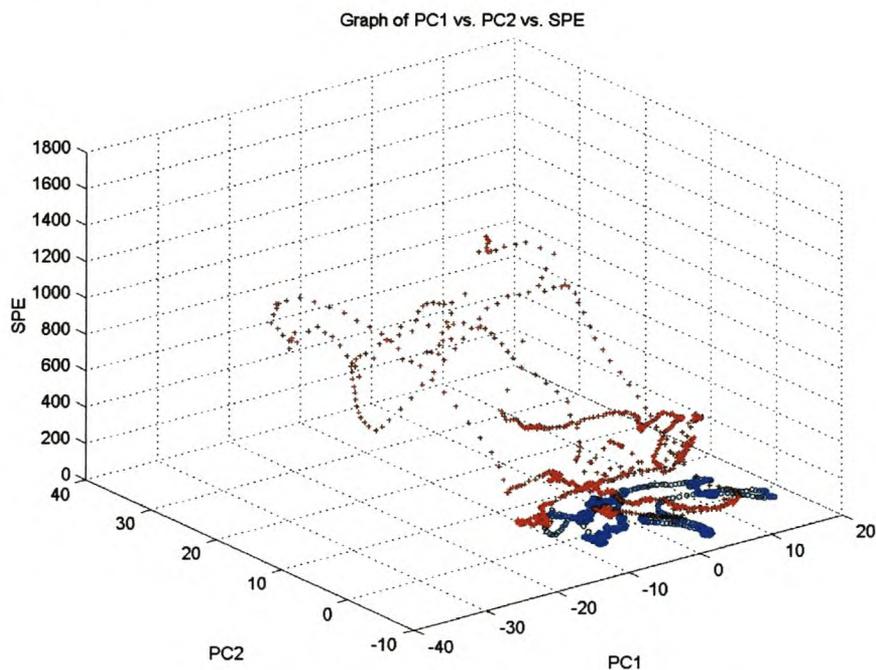
The above results are confirmed when looking at the combined score- and SPE-chart for test dataset 1 (figure 6.19). Here most of the scores for the test dataset 1 fall outside the range of the scores for the reference dataset. This plot indicates that all the variables have a larger than normal change relative to the reference dataset, causing the principal scores to move outside the region over which the PCA model was developed. It should also be noted that the instance where the scores deviated considerably from its own normal operating region coincided with the FAG mill fine feed rate ( $x_4$ ) sharply dropping to zero.



**Figure 6.20: T<sup>2</sup>- and SPE-charts for test dataset 2**



**Figure 6.21: Individual variable SPE-charts for test dataset 2; FAG mill power ( $x_1$ ), FAG mill load ( $x_2$ ), FAG mill fine feed rate ( $x_4$ ) and FAG mill inlet water flow rate ( $x_7$ )**

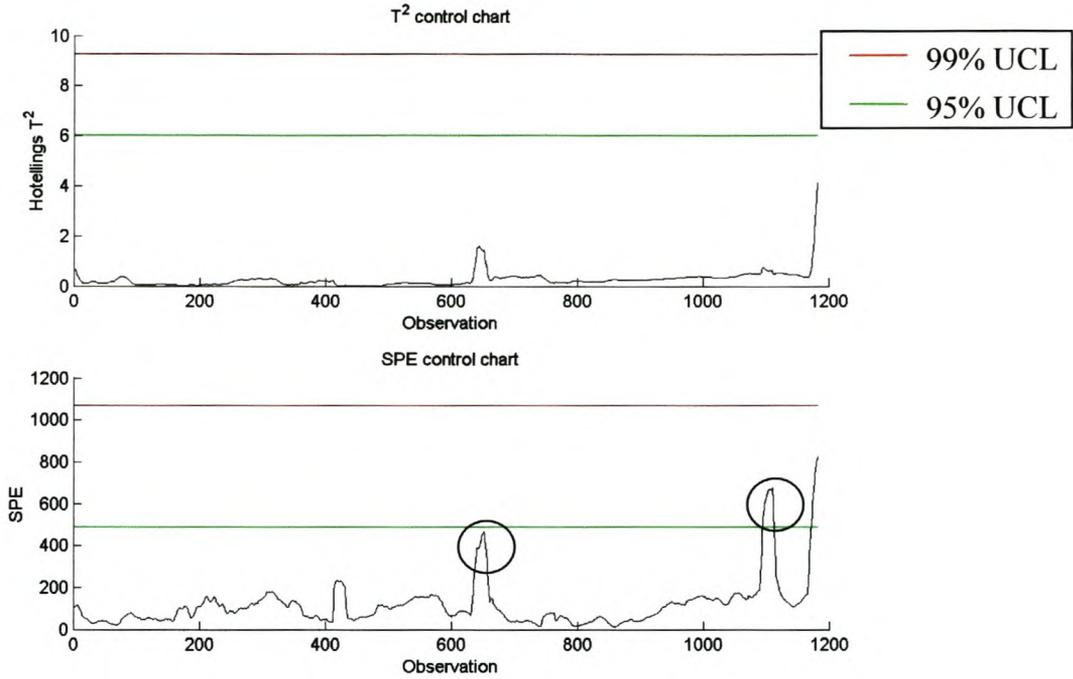


**Figure 6.22: Combined score- and SPE-chart for test dataset 2 (red crosses) in comparison to the reference dataset (blue circles)**

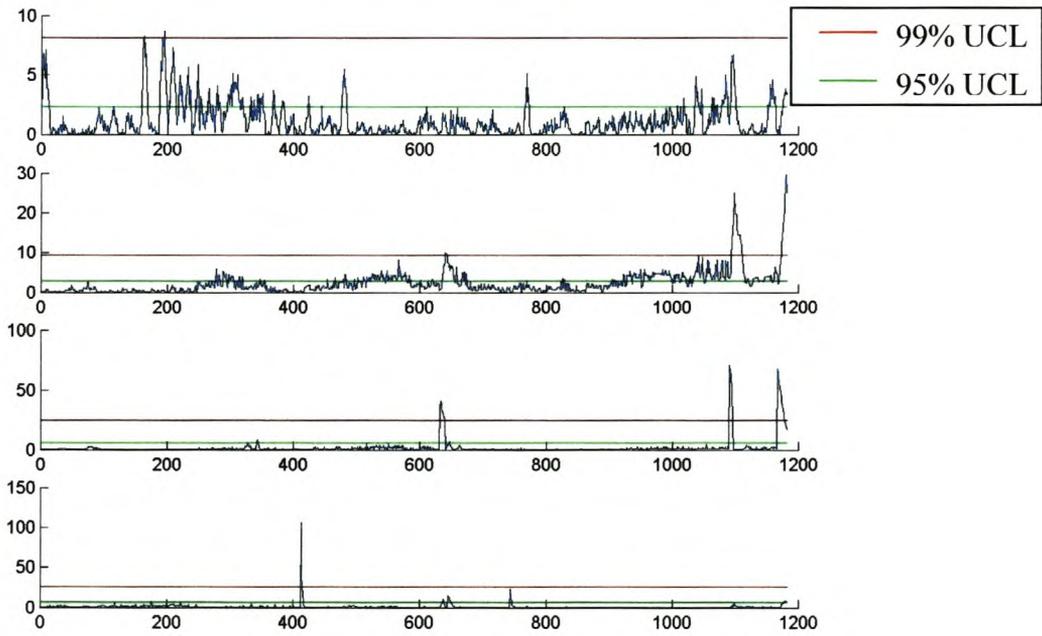
From the  $T^2$ -chart for test dataset 2 (figure 6.20) it can be seen that, similar to test dataset 1, only one instance existed where the 95% UCL was exceeded, with one other instance occurring where the  $T^2$  value came close to the 95% UCL. It was found that both instances were caused by the FAG mill fine feed rate ( $x_4$ ) sharply dropping to zero, resulting in a sharp drop in FAG mill inlet water flow rate ( $x_7$ ), in turn causing the FAG mill load ( $x_2$ ) and FAG mill power ( $x_1$ ) to decrease. The second disruption in the FAG mill fine feed rate ( $x_4$ ) was not long enough to cause the  $T^2$  value to exceed the 95% UCL.

For test dataset 2, both peaks in the  $T^2$ -chart are complimented with peaks in the SPE-chart (figure 6.20). On closer inspection (figure 6.21) it was noticed that after the first peak in the SPE-chart, the FAG mill power ( $x_1$ ) and FAG mill inlet water ( $x_7$ ) return to within the 95% UCL, with the FAG mill load ( $x_2$ ) and FAG mill fine feed rate ( $x_4$ ) starting to drift out of control. Contrary to this, after the second peak in the SPE-chart, the FAG mill power ( $x_1$ ) and FAG mill inlet water ( $x_7$ ) operated near the 99% UCL of the respective variables, with the FAG mill load ( $x_2$ ) and FAG mill fine feed rate ( $x_4$ ) returning to within the 95% UCL. In this case similar events caused the plant to react differently and the SPE-chart was ideal to indicate the abnormalities resulting from when the relationship among the variables or their dynamic patterns changed.

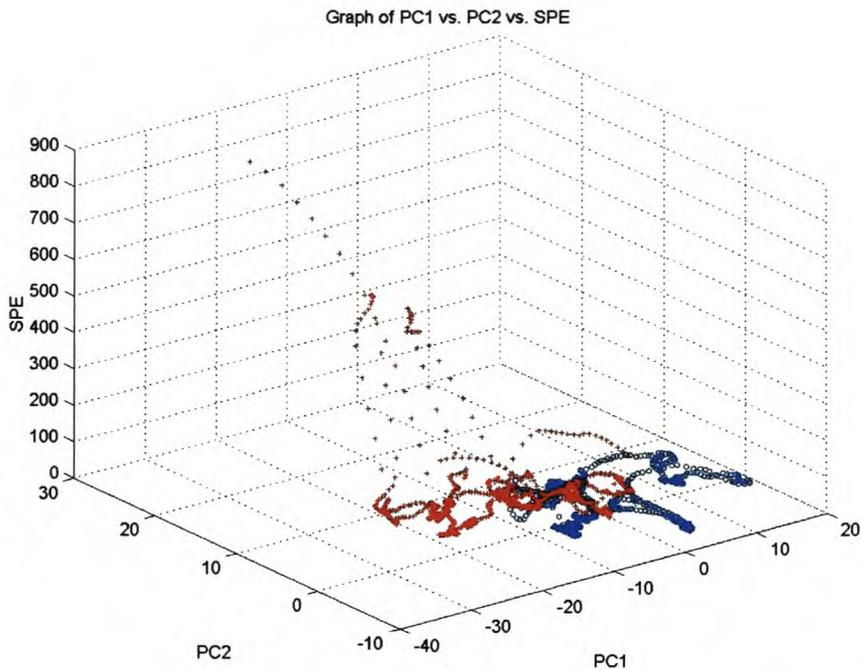
When considering the combined score- and SPE-chart for test dataset 2 (figure 6.22) it was noted that the scores moving out of the plane of normal operation coincided with the increase in the SPE. This also was indicative of the sudden shift that occurred in all the variables at those intervals.



**Figure 6.23: T<sup>2</sup>- and SPE-charts for test dataset 3**



**Figure 6.24: Individual variable SPE-charts for test dataset 3; FAG mill power ( $x_1$ ), FAG mill load ( $x_2$ ), FAG mill fine feed rate ( $x_4$ ) and FAG mill inlet water flow rate ( $x_7$ )**



**Figure 6.25: Combined score- and SPE-chart for test dataset 3 (red crosses) in comparison to the reference dataset (blue circles)**

The  $T^2$ -chart for test dataset 3 (figure 6.23) shows no instances where either the 95% UCL or 99% UCL was exceeded. This leads one to believe that the process was operated with only “common-cause” variations in the variables for the duration of the test dataset.

For test dataset 3, it was noticed that although both the FAG mill power ( $x_1$ ) and the FAG mill load ( $x_2$ ) often exceeded their respective 95% UCL (figure 6.24), noticeable peaks only occurred in the SPE-chart (figure 6.23) from deviations in the FAG mill fine feed rate ( $x_4$ ) resulting in deviations in either the FAG mill power ( $x_1$ ) or the FAG mill load ( $x_2$ ) or both. This has also been noticed in test dataset 1 and 2 and it would seem as if only major deviations in any of the process input variables, causing deviations in the process output variables, have a marked effect on the SPE-chart for the combined variables. This effect could be ascribed to the fact that the input variables account for most of the variance captured in  $t_1$ , making this technique very sensitive to which principal components are representative of which variables.

As before, the scores moving out of the plane of normal operation on the combined score- and SPE-chart for test dataset 3 (figure 6.25) coincided with the increase in the SPE.

#### 6.4 Summary

From the preceding application of PCA as a process model diagnostic system it became evident that the  $T^2$ -chart is not a useful tool when principal components are being retained with an estimated variance of less than 1, thereby prohibiting any reasonable conclusions to be drawn from this chart, as became evident through its application to data validation. However, SPE- and score-charts were successfully applied to data validation, effectively detecting special events. When found, these special events should be included in the model training dataset, and the model retrained.

PCA was also successfully applied to process monitoring but it was found that only major deviations in the process input variables, causing deviations in the process output variables, had a marked effect on the SPE- and score-charts. This effect was ascribed to the fact that the input variables accounted for most of the variance captured in the first principal component, the first principal component also capturing more than 50% of the variance in the data, making the technique very sensitive to which principal components are representative of which variables.

Although great success was achieved with the way in which linear PCA was applied for process model diagnosis, and its associated charts, some recommendations for improvement include:

- Because of the non-linearity in real processes, non-linear PCA for process model diagnosis could be used to improve results (Dong and McAvoy, 1996; Zhang et al., 1997; Jia et al., 1998; Fourie and de Vaal, 2000). Its application could also result in the first two principal components explaining more of the variance in the data than possible through linear PCA, assisting with the graphical interpretation of the results.

- PCA could also be used in conjunction with a neural network model of the process, where the monitoring charts are based on the multivariable residuals derived from the difference between the process measurements and the neural network predictions (Chen and Liao, 2002). Another possibility would be to feed the neural network predictions directly to the PCA model in order to determine when the process is moving away from its normal operating conditions prior to it actually moving away from its normal operating conditions.
- The confidence limits used to determine when the process is “in-control” and when it is “out-of-control” could also be refined. Kano et al. (2000) specified confidence limits such that the number of observations outside these limits was 1% of all the observations while the process was operated under normal conditions. Operator experience could also be used to refine the confidence limits.
- Since the only requirement for applying PCA for process model diagnosis is the existence of a good database on past operations, it is very important to ensure reliable, good quality data for development of the PCA model.

It was also found that the data being analysed was not always normally distributed. This resulted in the confidence limits calculated using equation 6.9 being invalid in certain circumstances, and not being shown on any of the score-charts.

## 7 Automated model-based mill control

---

Development of a controller using a model which does not cover the entire state space of the process will most likely result in the controller not being able to control the process at all times. Although a process model diagnostic system (as discussed in chapter 6) could be used to validate new process data, a neurocontroller, based on genetic algorithms and neural networks, for process control could also prove invaluable. Neurocontrollers are normally very robust and generalise very well when compared to the more traditional model predictive controllers. A neurocontroller would also be able to compensate for shortcomings in the process model that is used during the controller development.

### 7.1 Non-linear neurocontrol

For any system to operate more independently from human interference an increasing degree of autonomy in the system is required. For metallurgical systems this autonomy comes in the form of control and more recently intelligent control using neural networks (Duarte et al., 2001). For control, neural networks can be used either to adjust the parameters of a conventional controller or as a direct controller, termed a neurocontroller (Li and Häußler, 1996). Traditional neurocontrollers utilise a single error as network input to form a proportional type control action. However, this type of controller may not necessarily be able to cope with overshoots and oscillations in the controlled behaviour making it less than ideal for the control of complex metallurgical processes.

#### 7.1.1 Neurocontroller fundamentals

Gupta and Rao (1994) defined the functioning of a neurocontroller, in general, as performing a specific form of adaptive control where the controller takes the form of a multi-layer network and the adaptable parameters are defined as the adjustable weights. For the multi-layer neural network the most popular training method used is the backpropagation algorithm (discussed in chapter 3). However, from a control perspective, this training method suffers the following drawbacks (Li and Häußler, 1996):

- Gradient guidance passively adjusts parameters from the performance index, which must be differentiable or well behaved and thus may not allow modified error terms that suit real engineering problems.
- Training a direct feedback neurocontroller that meets constraint conditions in practical applications is difficult.
- The trained parameters may be local optima.
- Different network training control parameters (e.g. the learning rate and momentum rate) may result in different minimum RMS errors for the same number of epochs.
- The network architecture usually needs to be fixed prior to parameter training, preventing optimal architectures or topologies from being revealed for different types of data or applications.
- Incorporating knowledge and expertise that the designer may already have on the network design is usually difficult.

A possible solution to these problems is the use of reinforcement learning.

#### 7.1.1.1 Reinforcement learning

Reinforcement learning is learning through not specifying specific actions, but allowing the discovery of actions which will yield a maximum reward by actually trying them. This trial-and-error search characteristic, combined with a delayed reward characteristic, is probably the most important distinguishing feature of reinforcement learning (Sutton and Barto, 1998). During learning, the learning agent (e.g. the controller) needs to be able to observe the state of its environment (e.g. the process) and take actions that will affect this current environmental state. In contrast with many other approaches, reinforcement learning explicitly considers the whole problem of a goal-directed agent interacting with an uncertain environment, requiring the learning agent to therefore have an explicit goal.

For standard reinforcement learning it could therefore be said that the learning agent and its environment are connected through observations and actions. Whenever the agent observes an input, either a full or partial state representation of the environment, an action, based on this input, is chosen by the agent to generate an output (Conradie, 2000). The action leads to a changed environment causing the learning process to

repeat. The actions to be taken are based on a policy, a mapping from observed states of the environment to actions to be taken when in those states, that alone is sufficient to determine the behaviour of the agent. It is the task of the agent to find a policy that maximises the total reward it receives in the long run. In order for the agent to perform this task, reward and value functions are required.

The reward function maps each observed state of the environment to a single number, a reward, indicating the immediate intrinsic desirability of that state. In doing so events that are good and bad for the agent are defined. Whereas a reward function indicates whether events are good or bad in an immediate sense, a value function is indicative of good or bad events in a long-term sense. The value of a state can thus be defined as the total amount of reward an agent can expect to accumulate over the future, starting from that state. Although a state might always yield a low immediate reward, it may still have a high value based on it being regularly followed by other states yielding high rewards, or vice versa. Actions are therefore sought out to bring about states of highest value, not highest reward. Unfortunately, values are much harder to determine compared to rewards. An alternative to value functions is the use of evolutionary algorithms where the space of policies is directly searched without ever appealing to value functions (Sutton and Barto, 1998).

When comparing reinforcement learning to the more commonly used supervised learning, the former is able to learn from its own experience and not only from examples. In practice it is often impossible to obtain examples of desired behaviour that are both correct and representative of all possible situations to be explored necessitating the need for learning from own experience, which in turn is achieved through exploration and exploitation. Possible actions that have been discovered need to be exploited to obtain a better reward. Simultaneously new actions need to be explored in order to make better action selections in the future. Neither exploration nor exploitation can be pursued exclusively and a trade-off is required to ensure optimal learning. Evolutionary methods may be a solution to this problem.

#### 7.1.1.2 Evolutionary algorithms

Evolutionary algorithms (EAs) are the collective term used to describe problem-solving systems using known mechanisms of evolution in their design and

implementation. Whereas standard reinforcement learning focuses on individual structures, EAs maintain a population of structures that evolve according to rules of selection and evolutionary operators. The evolutionary operators used by EAs to simulate the evolution of individual structures are selection, mutation, recombination and reproduction. These search operators are responsible for exploitation and exploration as found in standard reinforcement learning. EAs, though simplistic, are sufficiently complex to provide robust and powerful adaptive search mechanisms (Heitkötter and Beasley, 2001).

In view of the biological processes on which EAs are based, the following is deemed important for better understanding (Heitkötter and Beasley, 2001):

- Evolution is not a directed process but could be considered a haphazard generation of biologically diverse organisms.
- When two or more individuals are competing for the same resources in an environment those that are better are more likely to survive and propagate their genetic material, termed natural selection.
- In nature, encoding of genetic information is done in such a way that allows reproduction where the offspring is genetically identical to the parent.
- Reproduction, where genetic information is exchanged between chromosomes producing offspring that contain information from each parent is termed recombination, often also referred to as crossover. Selection of which parents get to mate is largely a function of the individual's fitness (how good the individual is at competing in its environment) but a certain degree of luck (randomness) is also involved. When individuals are selected probabilistically it is termed fitness-proportionate selection whereas when individuals are randomly selected in a subgroup to compete and the fittest is selected it is termed tournament selection.
- In evolution, diversity is ensured through mutation. In EAs, a large amount of diversity is usually introduced at the start of the algorithm by randomising the genes in the population.
- Although EAs use stochastic processes, they are not random searches for a solution to a problem, their results being clearly non-random.

Combining evolutionary algorithms and reinforcement learning results in a learning technique, evolutionary reinforcement learning, that has several advantages over standard reinforcement learning. One of the more important advantages of evolutionary reinforcement learning is the fact that it can effectively be used to balance exploration and exploitation by maintaining a population of different strategies. Using evolutionary reinforcement learning, exploitation is accomplished by assigning a greater number of evaluations to individuals that display more effective behaviour than an average individual as well as by transferring parents with high fitness values unchanged to the next generation for evaluation. Exploration, using evolutionary reinforcement learning, is represented by the evolutionary (search) operators producing offspring, which, unlike other random exploration strategies, is directed towards solutions having a greater probability of producing greater rewards (Conradie, 2000).

Evolutionary reinforcement learning also has added advantages when considering problems in which the learning agent cannot accurately sense the state of its environment (Sutton and Barto, 1998). Real problems often require non-Markov decision processes where the agent can not only be uncertain about the state of the external task but also requires past and current state information in order to take appropriate action. This makes evolutionary methods especially useful for application to non-Markov decision problems where the straightforward application of traditional reinforcement learning methods often yields sub-optimal results or severely degraded performance (Whitehead and Lin, 1995).

When considering all of the above it was decided to use the SANE (Symbiotic, Adaptive Neuro-Evolution) evolutionary reinforcement learning algorithm, developed by Moriarty and Miikkulainen (1996), for developing a neurocontroller with which to control the Amandelbult FAG mill.

#### 7.1.1.3 The SANE evolutionary reinforcement learning algorithm

Whereas most current advanced control products are based on algorithmic approaches (i.e., model predictive control (MPC) software), the SANE algorithm is biologically motivated, using ideas from both neuro-science (neural networks) and Darwinian

evolution (genetic algorithms). In essence the SANE algorithm uses symbiotic evolution to evolve a population of hidden neurons for a given type of neural network architecture. Moriarty and Miikkulainen (1998) defined the basic steps in one generation of SANE as follows:

1. Fitness values and blueprints for each neuron are cleared.
2. Neurons of size  $\zeta$  are randomly selected from the population using a blueprint.
3. Use the selected neurons to create a neural network.
4. The network is evaluated in the given task and assigned a score.
5. The blueprint is assigned the evaluation of the network as its fitness.
6. Repeat steps 2-4 until each neuron has participated in a sufficient number of networks.
7. Assign each neuron the evaluation of the best 5 neural networks in which it participated.
8. Based on the average fitness value of each neuron, crossover operations are performed on the top 20% of the population. Selection by rank is employed instead of the standard fitness-proportionate selection to ensure a bias towards the best performing neurons. The offspring replaces the parents as well as the worst performing neurons in the population. As an insurance policy against missing genetic material, and not as a mechanism to create diversity, mutation, at the rate of approximately 0.1%, is performed on the new offspring.

Using the above methodology, a neurocontroller can be designed having numerous advantages over more traditional controller design techniques. Some of these advantages include (Weeks and Burgess, 1997; Li and Häußler, 1996; Moriarty and Miikkulainen, 1996; Conradie and Aldrich, 2001a):

- No previous knowledge about the system to be controlled is required. This includes the dimensionality of the system and the location of the unstable fixed points.
- The controller structure is not fixed and the neural network allowed to adopt non-linear forms.

- The neural network driving the controller is able to ignore useless inputs and the size of its hidden layer does not need to be precisely predetermined. The specialization of neurons can result in neurons that are redundant.
- Rather than by providing correct signals for various input data, the neural network weights can be found by examining the performance of the network as a controller.
- The algorithm can simultaneously evaluate performance at multiple points while intelligently searching through the solution space. This allows the global optima for almost any type of objectives to be approached without the need for differentiations.
- No need exists for the plant model to be linearised first. It has been found that a neurocontroller designed using a linearised model resulted in a performance inferior to that obtained by directly using the original non-linear model.
- The SANE algorithm is specifically designed to maintain a diverse population of neurons in prolonged evolution, aiding the parallel searching power of genetic algorithms.
- Process-manipulated variable pairings and the elimination of control interactions are implicitly determined whereas MPC requires explicit definition of these control considerations.
- The controller is more robust than MPC in the presence of plant/model mismatch or large process disturbances.
- A large degree of controller autonomy is demonstrated owing to the generalisation afforded to the neurocontroller by the SANE algorithm in dealing with considerable uncertainty in its operating environment.

It immediately becomes evident that the use of the SANE algorithm for non-linear multivariable controller design is far superior to PID control and has proven to be more effective than various other controller designs, including model predictive control. Neurocontrollers, developed for process control using the SANE algorithm, have successfully been applied to simulation studies on a bioreactor (Conradie et al., 2000), the Tennessee Eastman Control Challenge (Conradie and Aldrich, 2001a) and a ball mill grinding circuit (Conradie and Aldrich, 2001b). More recently the SMNE algorithm (Conradie et al., 2002a), a derivative of the SANE algorithm, was applied to

the neurocontroller development for a pilot scale multi-effect batch distillation (MEBAD) column. The neurocontroller outperformed a pure feedback PI controller by ensuring near-optimal temperature and hold-up profiles in the columns and vessels, significantly reducing the batch time. In view of the aforementioned success obtained, the decision to use the SANE algorithm for developing a neurocontroller with which to control the Amandelbult FAG mill was supported.

### 7.1.2 Amandelbult FAG mill neurocontroller

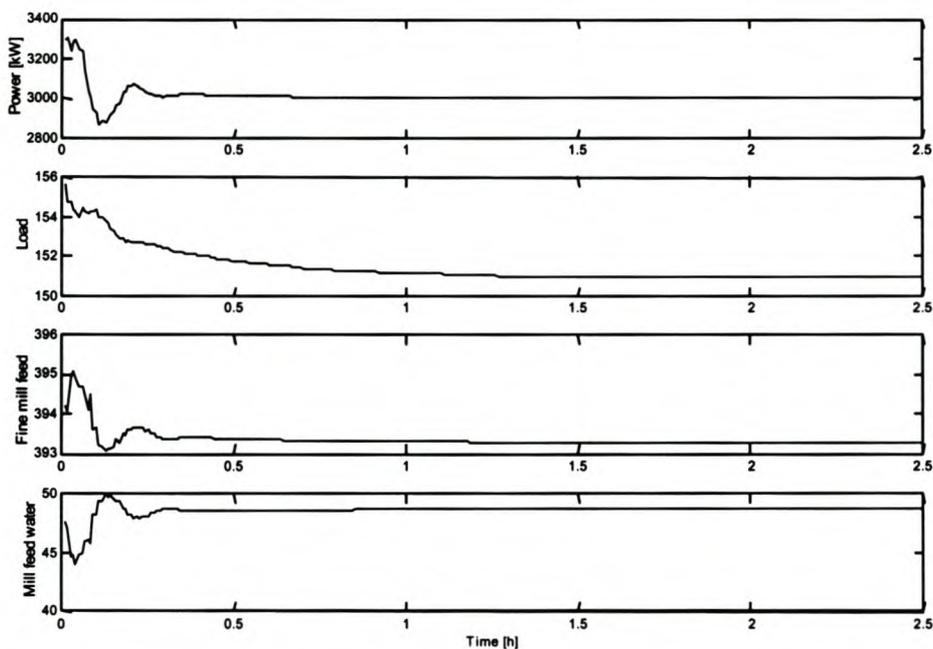
For the development of a neurocontroller a model with which to train the controller is required. Based on the work reported in this thesis and the requirements of the software implementing the SANE algorithm for neurocontroller development (developed at the University of Stellenbosch), a neural network model very similar to *MLPSSA\_1* was constructed.

As with the *MLPSSA\_1* model, all the variables were embedded using embedding parameters estimated using SSA; these include the input variables that are also the principal variables that can affect the control of an autogenous mill. The dataset used for training this model was also the same one used to train the *MLPSSA\_1* model, with the exception that the number of variables were reduced from 7 to 4; the variables to be used being only the FAG mill power ( $x_1$ ), FAG mill load ( $x_2$ ), FAG mill fine feed rate ( $x_4$ ) and FAG mill inlet water flow rate ( $x_7$ ). The resulting model was a MLP neural network containing 7 nodes in its single hidden layer. The output of this model predicts both the FAG mill power ( $x_1$ ) and the FAG mill load ( $x_2$ ) one-step ahead in time (30 seconds sampling interval).

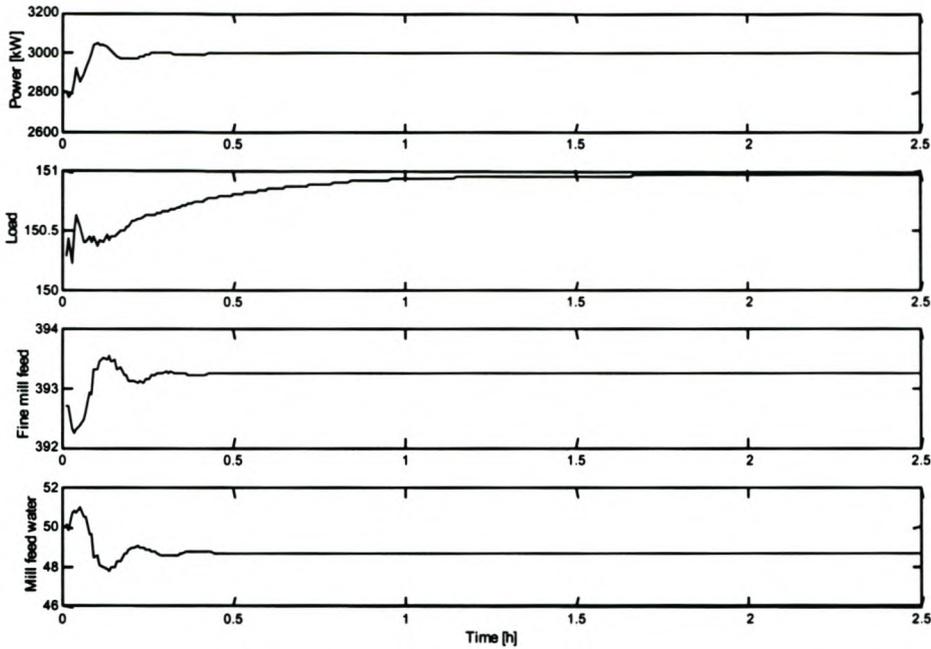
Using the aforementioned model with the neurocontroller software utilising the SANE algorithm, a neurocontroller for the UG2 FAG mill at Amandelbult was developed. Simulation tests were conducted using the developed neurocontroller whereby the desired setpoint for the FAG mill power ( $x_1$ ) was set at 3000 [kW], although only sparse process data was available within this operating region, with the neurocontroller determining the FAG mill fine feed rate ( $x_4$ ) and the FAG mill inlet water flow rate ( $x_7$ ) to be used in order to achieve the desired setpoint.

During simulation, the desired setpoints were achieved within approximately one hour after starting conditions (figures 7.1 and 7.2), which is as expected when analysing the original process data. The neurocontroller also selected the FAG mill inlet water flow rate ( $x_7$ ) as the primary manipulated variable, in that the control actions have a larger magnitude than for the FAG mill fine feed rate ( $x_4$ ). This finding is consistent with results obtained from both a sensitivity analysis and decision tree development done for the system under investigation.

It should be noted that a drawback of using artificial neural networks for control, as with modelling, is that the network “learns” a region for the input-output response which does not mean it is easy to extrapolate, scale up, or transfer the “learning” to other regions of operation as you can with a model composed of differential equations (Hoskins and Himmelblau, 1992). It is therefore important that sufficient process data be collected in the region of the desired process setpoint (currently the FAG mill power ( $x_1$ ) at 3000 [kW]), which should result in improved process model performance in this region of the state space.



**Figure 7.1: Process responses for FAG mill power ( $x_1$ ) initially above 3000 [kW]**



**Figure 7.2: Process responses for FAG mill power ( $x_1$ ) initially below 3000 [kW]**

## 7.2 Summary

It was shown that a neural network model built using the modelling methodology from chapter 5 could effectively be used, in conjunction with the SANE algorithm, to develop a neurocontroller. During simulation studies the neurocontroller proved capable of successfully achieving desired setpoints for the FAG mill power ( $x_1$ ). These are, however, only simulation results and the real ability of the developed neurocontroller can only be determined through on-line application of the controller. A plant test, using the developed neurocontroller to control the Amandelbult FAG mill, will in effect determine the accuracy of both the process model and the neurocontroller.

## 8 Conclusions

---

Based on the results obtained from this research and the discussion thereof, the following conclusions were drawn:

- Since the time series analysis and modelling techniques used were all data driven, a comprehensive understanding of the system under investigation and the fundamentals of the process to be modelled was deemed necessary. On occasion the techniques used did not give exact results. This was ascribed to the nature of the data under investigation and was found when considering the sensibility of some of the results obtained. Consequently, the fundamental knowledge of the system was found to greatly assist in the interpretation of the results.
- Familiarisation with the data to be used for modelling gave insight into the nature of the data and emphasized problem areas in the data. It was found from both the LA function and the FNN statistic that the data were very noisy. Furthermore, comparable results were obtained between the LA function and the AMI statistic for embedding delay estimation. It was also found that there was no significant distinction between the embedding dimensions calculated using FNN based on the embedding delay calculated by means of either the LA function or the AMI statistic. This in turn indicated that, for the data under investigation, the choice of embedding delay did not carry a significant weight when using the FNN statistic to calculate the embedding dimension.
- The method of surrogate data was used to determine the suitability of the data for predictive modelling through the successful classification of all the time series as either exhibiting deterministic or stochastic character. The FAG mill power ( $x_1$ ) time series, the focus of the predictive modelling in this research, was found to be highly deterministic, having a significant displacement between the correlation dimension curves of the surrogate data and that of the observed data, and was thus considered suitable for predictive modelling.

- A direct connection was found to exist between the embedding strategy used, the quality of the reconstructed attractor, and the quality of the resulting model. It was concluded that for the system under investigation the better the quality of the attractor reconstruction, based on the embedding strategy, and the more of the state space it covered, the more the attractor was unfolded and the better the quality of the resulting model. From the various embedding strategies tested it was further found that the attractor reconstructions based on the SSA embeddings were superior to that of the reconstructed attractors based on the MOD embeddings. This difference can be ascribed to the fact that SSA is more robust in the presence of noise, owing to its built-in noise filter, when compared to the MOD when used for attractor reconstruction.
- In all instances the models based on the SSA embedding strategies outperformed those based on the MOD embedding strategies. This was a direct result from the superior attractor reconstructions obtained based on the SSA embedding strategies and found to be the case for both the one-dimensional and multi-dimensional time series models. When comparing the one-step ahead prediction results for the one-dimensional and multi-dimensional time series models it was found that by adding more variables as model inputs did not significantly improved the modelling results; the additional variables not presenting any new, useful information for the modelling of the system. However, from a control perspective, including both the input and the output variables of the system in a multi-dimensional embedding prior to modelling is important and did improve the modelling results. Furthermore, embedding of the multi-dimensional time series data prior to modelling, compared to only using the data at time  $t-1$  to make predictions at time  $t$ , also improved the prediction accuracy of the developed models.

- When comparing the different modelling techniques applied, it was found that the MLP neural network models generally performed the best, with comparable results from the LS-SVM, RBF neural network and linear ARX models. The best MLP neural network model had as input all the process inputs, embedded using the embedding parameters estimated by SSA, and all the process outputs, embedded with an embedding delay and embedding dimension of unity. In contrast, modelling using LS-SVMs, having the advantage of scaling very well to high dimensional input spaces (Suykens, 2001), allowed for the FAG mill power ( $x_1$ ) to be accurately modelled without explicitly selecting the input variables for the model and by fixing the embedding delay at unity and the embedding dimension at the arbitrarily chosen value of 20. However, this technique is computationally much more time-consuming and it takes more than three times as long to determine an optimal LS-SVM model when compared to a MLP neural network model.
- The strength of the RBF neural network models on the other hand are that their one-dimensional time series models appeared to be better suited to free-run prediction when compared to the other one-dimensional models. The RBF neural networks seemed to be better at capturing the underlying dynamics in the one-dimensional FAG mill power ( $x_1$ ) time series than the other modelling techniques applied. Contrary to the free-run prediction success obtained from the one-dimensional RBF neural networks models, a multi-dimensional MLP neural network model with appropriately embedded model inputs outperformed all other models with regard to free-run prediction success. This is a clear indication of the promise shown by embedding the data prior to non-linear modelling. Further model improvements could also possibly be gained by using more training data, covering a larger region of the state space, which also more homogeneously populate the state space.

- Owing to the fact that any empirical model is only as good as the model training data and only valid for data similar to the model training data, a process model diagnostic system, to accompany the process model, was needed. Using linear PCA combined with SPE- and score-charts, a process model diagnostic system was developed and successfully used for data validation. This system effectively detected special events that, when found, could be included in the model training dataset and the model retrained. It was also suggested that, owing to the inherent non-linearity in real processes, the use of non-linear PCA could lead to an improved process model diagnostic system. Usually the application of non-linear PCA leads to the first two principal components explaining more of the variance in the data than possible through linear PCA, assisting with the graphical interpretation of the results.
- The application of one of the developed models in an advanced control system was also in part accomplished through the development of a neurocontroller using evolutionary reinforcement learning. One of the models developed during the research was successfully used, in conjunction with the SANE algorithm, to develop the neurocontroller. The ability of the developed neurocontroller was also shown through simulation studies; the neurocontroller successfully achieved various set points for the FAG mill power ( $x_1$ ).

In short, the following methodology, as carried out during the research, is presented for the modelling and control of an autogenous mill:

1. Thoroughly investigate the system to be modelled in order to better understand the system and to gain insight into the fundamentals of the process to be modelled.
2. Classify the process variables as either input or output variables. Whereas input variables can further be divided into manipulated and disturbance variables, output variables can be subdivided into controlled and measured variables.

3. Calculate the various statistics required for non-linear system identification and reconstruct the state space.
4. Using the method of surrogate data, determine the suitability of the data for predictive modelling, by classifying each time series as either deterministic or stochastic.
5. Develop and validate the desired process model.
6. Using linear or non-linear PCA develop and calibrate a process model diagnostic system.
7. Develop and test the controller to be used in an advanced control system.

## 9 References

---

Abarbanel, H.D.I., 1996. *Analysis of Observed Chaotic Data*.

New York: Springer-Verlag.

Abarbanel, H.D.I., Masuda, N., Rabinovich, M.I. and Turner, E., 2001. Distribution of mutual information.

*Physics Letters A*, 281, 368-373.

Adusumilli, S. and Rivera, D.E., 1997. *A Methodology for Integrated System Identification and PID Controller Design*.

*In: IFAC Symposium on Advanced Control of Chemical Processes (ADCHEM '97)*, Banff 9-11 June, Canada, 123-128.

Aldrich, C., Reuter, M.A. and van Deventer, J.S., 1993. Control and Optimisation Techniques: Artificial Intelligence in Process Engineering.

*In: H.W. Glen, ed. Measurement, Control, and Optimisation in Mineral Processing*. Johannesburg: The South African Institute of Mining and Metallurgy, 123-156.

Aleksić, Z., 1991. Estimating the embedding dimension.

*Physica D*, 52, 362-368.

Aström, K.J. and Hägglund, T., 1995. *PID Controllers: Theory, Design, and Tuning*.

Instrument Society of America, Research Triangle Park, North Carolina.

Atmanspacher, H., Scheingraber, H. and Voges, W., 1988. Global scaling properties of a chaotic attractor reconstructed from experimental data.

*Physical Review A*, 37 (4), 1314-1322.

Barnard, J.P., 1999. *Empirical State Space Modelling with Application in Online Diagnosis of Multivariate Non-Linear Dynamic Systems*.

Thesis (PhD), University of Stellenbosch.

Bellman, R.E., 1961. *Adaptive Control Process*.

Princeton: Princeton University Press.

Berardinis, L.A., 1992. Clear thinking on fuzzy logic.

*Machine Design*, 64 (8), 46-53.

Bishop, C.M., 1999. *Neural Networks for Pattern Recognition*.

Great Britain: Oxford University Press.

Bond, F.C., 1952. The third theory of comminution.

*Transactions SME/AIME*, 193, 484-494.

Brambilla, A., Scali, C. and Chen, S., 1989. Tuning of conventional controllers for robust performance.

*In: IFAC Symposium on Low Cost Automation Techniques, Components and Instruments, Applications*, Milan 8-10 November 1989. Elmsford, NY, USA: Pergamon Press Inc., 123-128.

Broomhead, D.S. and King, G.P., 1986. Extracting qualitative dynamics from experimental data.

*Physica D*, 20, 217-236.

Broomhead, D.S. and Lowe, D., 1988. Multivariate functional interpolation and adaptive networks.

*Complex Systems*, 2, 321-355.

Cattell, R.B., 1966. The scree test for the number of factors.

*Multivariate Behavioural Research*, 1, 245-276.

Chen, G. and McAvoy, T.J., 1998. Predictive on-line monitoring of continuous processes.

*Journal of Process Control*, 8 (5-6), 409-420.

Chen, J. and Liao, C.M., 2002. Dynamic process fault monitoring based on neural network and PCA.

*Journal of Process Control*, 12, 277-289.

Chen, S., Cowan, C.F.N. and Grant, P.M., 1991. Orthogonal least squares learning for radial basis function networks.

*IEEE Transactions on Neural Networks*, 2(2), 302-309.

Cohen, H.E., 1983. Energy usage in mineral processing.

*Trans Inst Min Metall*, 92, C160-C163.

Conradie, A.v.E., 2000. *Neurocontroller Development for Nonlinear Processes Utilising Evolutionary Reinforcement Learning*.

Thesis (MSc). University of Stellenbosch.

Conradie, A.v.E. and Aldrich, C., 2001a. Plant-Wide Neurocontrol of the Tennessee Eastman Challenge Process Using Evolutionary Reinforcement Learning. [online]

3<sup>rd</sup> International Conference in Intelligent Processing and Manufacturing of Materials, Vancouver, Canada, 29 July – 3 August 2001. Available from: <http://www.chemeng.sun.ac.za/Content/conradie.htm> [12 April 2002].

Conradie, A.v.E. and Aldrich, C., 2001b. Neurocontrol of a ball mill grinding circuit using evolutionary reinforcement learning.

*Minerals Engineering*, 14 (9), 1277-1294.

Conradie, A.v.E., Miikkulainen, R. and Aldrich, C., 2002a. Intelligent Process Control utilising Symbiotic Memetic Neuro-Evolution. [online]

To appear in proceedings of the 2002 Congress on Evolutionary Computation (CEC-2002). Available from: <http://www.chemeng.sun.ac.za/Content/conradie.htm> [12 April 2002].

Conradie, A.v.E., Nieuwoudt, I. and Aldrich, C., 2000. Nonlinear neurocontroller development with evolutionary reinforcement learning. [online]

9<sup>th</sup> National Conference of the South African Institution of Chemical Engineers, Secunda, South Africa. Available from: <http://www.chemeng.sun.ac.za/Content/conradie.htm> [12 April 2002].

Demuth, H. and Beale, M. 2001. *Neural Network Toolbox For Use with Matlab*. United States of America: The MathWorks, Inc.

Dong, D. and McAvoy, T.J., 1996. Nonlinear principal component analysis – based on principal curves and neural networks.

*Computers and Chemical Engineering*, 20 (1), 65-78.

Du, Y-G., de Villar, R. and Thibault, J., 1997. Neural net-based softsensor for dynamic particle size estimation in grinding circuits.

*International Journal of Minerals Processing*, 52, 121-135.

Duarte, M., Suarze, A., and Bassi, D., 2001. Control of grinding plants using predictive multivariable neural control.

*Powder Technology*, 115, 193-206.

Dubé, Y., Lanthier, R. and Hodouin, D., 1987. Computer aided dynamic analysis and control design for grinding circuits.

*CIM Bulletin*, 80 (905), 65-70.

Eckmann, J.P. and Ruelle, D., 1985. Ergodic theory of chaos and strange attractors.

*Rev. Mod. Phys.*, 57, 617-656.

Eksteen, J.J. and Reuter, M.A., 2000. Model based control of metallurgical furnaces.

*In: SAIMM Workshop on Developments in Non-Ferrous Pyrometallurgy*, Bantry Bay 23 August, Western Cape Branch of the SAIMM, South Africa.

Flament, F., Thibault, J. and Hodouin, D., 1992. The control of a grinding process with a neural network trained on differential data.

*In: S.A. Argyropoulos and G. Carayannis, eds. Artificial Intelligence in Materials Processing, Edmonton 23-27 August 1992. Canada: Canadian Institute of Mining, Metallurgy and Petroleum, 17-31.*

Fourie, S.H. and de Vaal, P., 2000. Advanced process monitoring using an on-line non-linear multiscale principal component analysis methodology.

*Computers and Chemical Engineering, 24, 755-760.*

Fraser, A.M., 1989. Reconstructing attractors from scalar time series: a comparison of singular system and redundancy criteria.

*Physica D, 34, 391-404.*

Gill, C.B., 1991. *Materials Research and Engineering: Materials Beneficiation.*

United States of America: Springer-Verlag.

Glass, L. and Mackey, M.C., 1988. *From clocks to chaos: the rhythms of life.*

Princeton: Princeton University Press.

Gradišek, J., Friedrich, R., Govekar, E. and Grabec, I., 2002. Analysis of data from periodically forced stochastic processes.

*Physics Letters A, 294, 234-238.*

Gradišek, J., Siegert, S., Friedrich, R. and Grabec, I., 2000. Analysis of time series from stochastic processes.

*Physical Review E, 62 (3), 3146-3155.*

Grassberger, P. and Procaccia, I., 1983. Characterisation of strangeness of strange attractors.

*Physica D, 9, 189-208.*

Gunn, S.R., Brown, M. and Bossley, K.M., 1997. Network performance assessment for neurofuzzy data modelling.

*Lecture Notes in Computer Science*, 1280, 313-323.

Gupta, M.M. and Dandina, H.R., 1994. Neuro-Control Systems: A Tutorial.

*In: M.M. Gupta and D.H. Rao, eds., NeuroControl Systems: Theory and Application.* IEEE Press, 1-43.

Hagan, M.T. and Menhaj, M., 1994. Training feed-forward networks with the Marquardt algorithm.

*IEEE Transactions on Neural Networks*, 5 (6), 989-993.

Hanley, J.P., 1990. How to Keep Control Loops in Service.

*Intech*, 10, 30.

Harris, C.A. and Meech, J.A., 1987. Fuzzy logic: a potential control technique for mineral processing.

*CIM Bulletin*, 80 (905), 51-59.

Hayes, P., 1993. *Process Principles in Minerals & Materials Production.*

Australia: Hayes Publishing Co.

Heiskanen, K., 1989. The modelling of autogenous grinding.

*In: V. Koppel, ed. Automation in Mining, Mineral and Metal Processing,* Buenos Aires 4-8 September 1989. Great Britain: Pergamon Press, 115-121.

Heitkötter, J. and Beasley, D., eds. (2001). *The Hitch-Hiker's Guide to Evolutionary Computation: A list of Frequently Asked Questions (FAQ)* [online]. USENET: comp.ai.genetic.

Available from: <ftp://rtfm.mit.edu/pub/usenet/news.answers/ai-faq/genetic/> [12 April 2002].

Hiemstra, S.A., 1982. The distribution of some platinum-group elements in the UG-2 Chromitite layer of the Bushveld Complex.

*Economic Geology*, 80, 944-957.

Hively, L.M., Gailey, P.C. and Protopopescu, V.A., 1999. Detecting dynamical change in nonlinear time series.

*Physics Letters A*, 258, 103-114.

Hukki, R.T., 1961. Proposal for a Solomonian settlement between the theories of von Rittinger, Rick and Bond.

*Transactions SME/AIME*, 220, 403-408.

Hulbert, D.G., Craig, I.K., Coetzee, M.L. and Tudor D., 1990. Multivariable control of a run-of-mine milling circuit.

*Journal of South African Institute of Minerals and Metallurgy*, 90 (7), 173-181.

Hussain, M.A., 1999. Review of the applications of neural networks in chemical process control – simulation and online implementation.

*Artificial Intelligence in Engineering*, 13, 55-68.

Jamsa, S-L., 1988. Experiences in the use of expert system in grinding process control in Siilinjärvi concentrator.

*In: E. Forssber, ed. XVI International Mineral Processing Congress, Stockholm 5-10 June 1988. Amsterdam: Elsevier Science Publishers, 1759-1771.*

Jia, F., Martin, E.B. and Morris, A.J., 1998. Non-linear principal components analysis for process fault detection.

*Computers and Chemical Engineering*, 22, S851-S854.

Judd, K., 1992. An improved estimator of dimension and some comments on providing confidence intervals.

*Physica D*, 56, 216-228.

Kanjilal, P.P., 1995. *Adaptive prediction and predictive control*.

London: Peter Peregrinus Ltd.

Kano, M., Nagao, K., Hasebe, S., Hashimoto, I., Ohno, H., Strauss, R. and Bakshi, B., 2000. Comparison of statistical process monitoring methods: application to the Eastman challenge problem.

*Computers and Chemical Engineering*, 24, 175-181.

Kantz, H. and Schreiber, T., 2000. *Nonlinear Time Series Analysis*.

United Kingdom: University Press, Cambridge.

Kennel, M.B. and Abarbanel, H.D.I., 1995. False neighbors and false strands: A reliable minimum embedding dimension algorithm.

*Physical Review E*, preprint.

Kennel, M.B., Brown, R. and Abarbanel, H.D.I., 1992. Determining embedding dimension for phase-space reconstruction using a geometrical construction.

*Physical Review A*, 45, 3403-3411.

Kharva, M., 2001. *Monitoring of froth systems using principal component analysis*.

Thesis (MSc). University of Stellenbosch.

Klimasauskas, C.C., 1992. Neural networks: an engineering perspective.

*IEEE Communications Magazine*, 30 (9), 50-53.

Kick, F., 1883. The law of proportional resistance and its application to sand and explosions.

*Dinglers Journal*, 247, 1-5.

Kim, I.W., Liebman, M.J. and Edgar, T.F., 1990. Robust error-in-variables estimation using nonlinear programming techniques.

*AIChE Journal*, 36 (7), 985-993.

Knoper, M., 2001. *Bushveld Complex, Rockscapes*.

Available from: <http://www.geocities.com/Yosemite/Trails/1453/geology.htm>  
[20 June 2002].

Kourti, T. and MacGregor J.F., 1995. Process analysis, monitoring and diagnosis, using multivariate projection methods.

*Chemometrics and Intelligent Laboratory Systems*, 28, 3-21.

Kravaris, C. and Arkun, Y., 1991. Geometric Linear Control – An overview.

In: Y. Arkun and H Ray, eds. *Chemical Process Control-CPCIV, 4<sup>th</sup> International Conference*, Padre Island February 1991. Austin, Texas: CACHE American Institute of Chemical Engineers, 477-515.

Kresta, J.V., MacGregor, J.F. and Marlin, T.E., 1991. Multivariate statistical monitoring of process operating performance.

*The Canadian Journal of Chemical Engineers*, 69, 35-47.

Kugiumtzis, D., 1996. State space reconstruction parameters in the analysis of chaotic time series – the role of the time window length.

*Physica D*, 95 (1), 13-28.

Kugiumtzis, D. and Christophersen, N., 1997. *State space reconstruction: method of delays vs. singular spectrum approach* [online]. Department of Informatics, University of Oslo.

Available from: <http://citeseer.nj.nec.com/kugiumtzis97state.html> [12 December 2001].

Kugiumtzis, D., Lillekjendlie, B. and Christophersen, N., 1995. Chaotic time series Part I: Estimation of some invariant properties in state space.

*Modelling, Identification & Control*, 15 (4), 205-224.

Langari, R., 1993. Synthesis of nonlinear controllers via fuzzy logic.

*In: A. Kandel and G. Langholz, eds. Fuzzy Control Systems. United States of America: CRC Press, 264-273.*

Lee, W. and Weekman, V.W., 1976. Advanced Control Practice in the Chemical Industry: A View from Industry.

*American Institute of Chemical Engineers, 22, 27-38.*

Leon, R.X., 1999. *A data-driven approach to the development of a supervisory control system for an industrial semi-autogenous grinding circuit.*

Thesis (MSc). University of Stellenbosch.

Leung, K., Morrison, R.D. and Whiten, W.J., 1987. An energy based ore specific model for autogenous and semi-autogenous grinding.

*In: A. Varschavsky and P. Kittl, eds. Copper '87: Physical Metallurgy of Copper, Santiago, 72.85.*

Levenberg, K., 1944. A method for the solution of certain nonlinear problems in least squares.

*Quart. Applied Mathematics, 2, 164-168.*

Li, Y. and Häußler, A., 1996. Artificial evolution of neural networks and its application to feedback control.

*Artificial Intelligence in Engineering, 10, 143-152.*

Lillekjendlie, B., Kugiumtzis, D. and Christophersen, N., 1994. Chaotic time series Part II: System identification and prediction.

*Modelling, Identification and control, 15 (4), 225-243.*

Loveday, B.K., 1978. Prediction of autogenous milling from pilot-plant tests.

*In: M.J. Jones, ed. Proceedings of the 11<sup>th</sup> Commonwealth Mining and Metallurgical Congress, Hong Kong 1978. England: The Institution of Mining and Metallurgy, 557-564.*

- Lowrison, G.C., 1974. *Crushing and Grinding: The size reduction of solid materials*.  
London: Butterworth's.
- Lynch, A.J., 1977. *Mineral Crushing and Grinding Circuits: Their Simulation, Optimisation, Design and Control*.  
Great Britain: Elsevier Scientific Publishing Company.
- MacGregor, J.F. and Kourti, T., 1995. Statistical process control of multivariate processes.  
*Control Engineering Practice*, 3 (3), 403-414.
- Maciejowski, J.M., 1990. *Multivariable Feedback Design*.  
Great Britain: Addison-Wesley.
- Maerz, N.H., Palangio, T.C. and Franklin, J.A., 1996. WipFrag image based granulometry system.  
*In: J. Franklin and T. Katsabanis, eds. Proceedings of the FRAGBLAST 5 Workshop on Measurement of Blast Fragmentation, Montreal 23-24 August 1996.*  
Rotterdam: A.A. Balkema, 91-99.
- Marquardt, D.W., 1963. An algorithm for least squares estimation of nonlinear parameters.  
*Journal of SAIM*, 11, 431-441.
- McLaren, C.H. and De Villiers, J.P.R., 1982. The platinum-group chemistry and mineralogy of the UG-2 Chromitite layer of the Bushveld Complex.  
*Economic Geology*, 77, 1348-1366.
- Meech, J.A. and Jordon, L.A., 1993. Development of a self-tuning fuzzy logic controller.  
*Minerals Engineering*, 6 (2), 119-131.
- Miller, A.J., 1990. *Subset Selection in Regression*.  
London: Chapman and Hall.

Moriarty, D.E. and Miikkulainen, R., 1996. Efficient Reinforcement Learning through Symbiotic Evolution.

*Machine Learning*, 22, 11-33.

Moriarty, D.E. and Miikkulainen, R., 1998. Forming Neural Networks through Efficient and Adaptive Coevolution.

*Evolutionary Computation*, 5 (4), 373-399.

Moshgbar, M., Parkin, R. and Bearman, R.A., 1995. Application of fuzzy logic and neural network technologies in cone crusher control.

*Minerals Engineering*, 8 (1/2), 41-50.

Mukherjee, S., Osuna, E. and Girosi, F., 1997. Nonlinear prediction of chaotic time series using support vector machines.

*In: Proceedings of the 1997 7<sup>th</sup> IEEE Workshop on Neural Networks for Signal Processing*, Amelia Island 24-26 September 1997. Piscataway, NJ: Institute of Electrical and Electronics Engineers, 511-519.

Muratet, G. and Bourseau, P., 1993. Artificial intelligence for process engineering – state of the art.

*Supplement to Computers and Chemical Engineering*, 17, S381-S388.

Mylopoulos, J., Kramer, B.M., Wang, H., Benjamin, M.E., Chow, Q.B. and Mensah, S., 1992. Applications of expert systems to process control.

*In: S.A. Argyropoulos and G. Carayannis, eds. Artificial Intelligence in Materials Processing*, Edmonton 23-27 August 1992. Canada: Canadian Institute of Mining, Metallurgy and Petroleum, 3-15.

Napier-Munn, T.J., 1995. Detecting performance improvements in trials with time-varying mineral processes – three case studies.

*Minerals Engineering*, 8 (8), 843-858.

Napier-Munn, T.J., Morrell, S., Morrison, R.D. and Kojovic, T., 1999. *Mineral Comminution Circuits: Their Operation and Optimisation*.

Australia: Julius Kruttschnitt Mineral Research Centre.

Narendra, K.S. and Parthasarathy, K., 1990. Identification and control of dynamical systems using neural networks.

*IEEE Transactions on Neural Networks*, 1 (1), 4-27.

Nichols, J.M. and Nichols, J.D., 2001. Attractor reconstruction for non-linear systems: a methodological note.

*Mathematical Biosciences*, 171, 21-32.

Nijhuis, A., de Jong, S. and Vandeginste, B.G.M., 1997. Multivariate statistical process control in chromatography.

*Chemometrics and Intelligent Laboratory Systems*, 38, 51-62.

Nomikos, P. and MacGregor, J.F., 1995. Multivariate SPC charts for monitoring batch processes.

*Technometrics*, 37 (1), 41-59.

Orr, M.J.L., 1995. Regularisation in the selection of radial basis function centres.

*Neural Computation*, 7(3), 606-623.

Orr, M.J.L. (1996). *Introduction to radial basis function networks* [online]. Technical report, Institute for Adaptive and Neural Computation, Division of Informatics, Edinburgh University.

Available from: [www.anc.ed.ac.uk/~mjo/papers/intro.ps](http://www.anc.ed.ac.uk/~mjo/papers/intro.ps) [4 December 2001].

Orr, M.J.L. (1999a). *Recent advances in radial basis function networks* [online]. Technical report, Institute for Adaptive and Neural Computation, Division of Informatics, Edinburgh University.

Available from: [www.anc.ed.ac.uk/~mjo/papers/recad.ps](http://www.anc.ed.ac.uk/~mjo/papers/recad.ps) [4 December 2001].

Orr, M.J.L. (1999b). *Matlab functions for radial basis function networks* [online]. Technical report, Institute for Adaptive and Neural Computation, Division of Informatics, Edinburgh University.

Available from: [www.anc.ed.ac.uk/~mjo/software/rbf2.zip](http://www.anc.ed.ac.uk/~mjo/software/rbf2.zip) [4 December 2001].

Orr, M.J.L. (1999c). *Assessing RBF networks using DELVE* [online]. Technical report, Institute for Adaptive and Neural Computation, Division of Informatics, Edinburgh University.

Available from: [www.anc.ed.ac.uk/~mjo/papers/delve99.ps](http://www.anc.ed.ac.uk/~mjo/papers/delve99.ps) [4 December 2001].

Packard, N.H., Crutchfield, J.P., Farmer, J.D. and Shaw, R.S., 1980. Geometry from a time series.

*Physical Review Letters*, 45, 712-716.

Plaut, D., Nowlan, S. and Hinton, G.E., 1986. *Experiments on learning by back propagation*.

Technical Report CMU-CS-86-126, Department of Computer Science, Carnegie Mellon University, Pittsburgh, P.A.

Pomerleau, A., Hodouin, D., Desbiens, A. and Gagnon, E., 2000. A survey of grinding circuit control methods: from decentralized PID controllers to multivariable predictive controllers.

*Powder Technology*, 108, 103-115.

Popivanov, D. and Mineva, A., 1999. Testing procedures for non-stationarity and non-linearity in physiological signals.

*Mathematical Biosciences*, 157, 303-320.

Pouskouleli, G. and Wheat, T.A., 1992. Artificial intelligence and data acquisition for expert systems.

*In:* S.A. Argyropoulos and G. Carayannis, eds. *Artificial Intelligence in Materials Processing*, Edmonton 23-27 August 1992. Canada: Canadian Institute of Mining, Metallurgy and Petroleum, 165-174.

Reuter, M.A. and Yang, Y., 2001. *Modelling and Control of Resource Cycle*.

The Netherlands: Department of Applied Earth Sciences, Delft University of Technology.

Rhodes, C. and Morari, M., 1997. The false nearest neighbours algorithm: an overview.

*Computers and Chemical Engineering*, 21, S1149-S1154.

Sano, M. and Sawada, Y. 1985. Measurement of the Lyapunov spectrum from a chaotic time series.

*Physical Review Letters*., 55, 1082-1085.

Sauer, T., Yorke, J.A. and Casdagli, M., 1991. Embedology.

*Journal of Statistical Physics*, 65, 579-616.

Schreiber, T. and Schmitz, A., 2000. Surrogate time series.

*Physica D*, 142, 346-382.

Schwartz, G., 1978. Estimating the dimension of a model.

*Annals of Statistics*, 6, 461-464.

Siegert, S., Friedrich, R. and Peinke, J., 1998. Analysis of data sets of stochastic systems.

*Physics Letters A*, 243, 275-280.

Small, M. and Judd, K., 1998a. Correlation dimension: A Pivotal statistic for non-constrained realizations of composite hypotheses in surrogate data analysis.

*Physica D*, 129, 386-400.

Small, M. and Judd, K., 1998b. Comparison of new non-linear techniques with application to infant respiration.

*Physica D*, 117, 283-298.

Stange, W.W., 1993. Using artificial neural networks for the control of grinding circuits.

*Minerals Engineering*, 6 (5), 479-489.

Sutton, R.S. and Barto, A.G., 1998. *Reinforcement Learning: An Introduction*.

London: MIT Press.

Suykens, J.A.K., 2001. Nonlinear modelling and support vector machines.

*In: IEEE Instrumentation and Measurement Technology Conference*, Budapest 21-23 May 2001. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, 287-294.

Suykens, J.A.K., Lukas, L. and Vandewalle, J., 2000. Sparse approximation using least squares support vector machines.

*In: Proceedings of the IEEE International Symposium on Circuits and Systems*, Geneva 28-31 May 2000. Piscataway, NJ: Institute of Electrical and Electronics Engineers, II757-II760.

Taggart, A.F., 1951. *Elements of Ore Dressing*.

New York: Wiley.

Takens, F., 1981. Detecting strange attractors in turbulence.

*In: D.A. Rand and L.S. Young, eds., Dynamical Systems and Turbulence*, Warwick 1980, Lecture Notes in Mathematics 898, pages 366-381. Springer, Berlin, 1981.

Theiler, J., Eubank, S., Longtin, A., Galdikaran, B. and Farmer, J.D., 1992. Testing for nonlinearity in time series: the method of surrogate data.

*Physica D*, 58, 77-94.

Valery Jnr, W. and Morrell, S., 1995. The development of a dynamic model for autogenous and semi-autogenous grinding.

*Minerals Engineering*, 8 (11), 1285-1297.

Van Der Walt, T.J., Van Deventer, J.S.J. and Barnard, E., 1993. Neural nets for the simulation of mineral processing operations: Part I. Theoretical principles.

*Minerals Engineering*, 6 (11), 1127-1134.

Vapnik, V., 1995. *The Nature of Statistical Learning Theory*.

New York: Springer-Verlag.

Walters, B.M., 2001. UG-2 Chromitite – Mineralogical appraisal.

*Anglo Platinum*.

Weeks, E.R. and Burgess, J.M., 1997. Evolving artificial neural networks to control chaotic systems.

*Physical Review E*, 56 (2), 1531-1540.

Weiss, N.L., 1985. *SME Mineral Processing Handbook*.

New York: Society of Mining Engineers of the American Institute of Mining, Metallurgical, and Petroleum Engineers

Whitehead, S.D. and Lin, L.J., 1995. Reinforcement learning of non-Markov decision processes.

*Artificial Intelligence*, 73, 271-306.

Whiten, W.J., 1974. A matrix theory of comminution machines.

*Chemical Engineering Science*, 29, 589-599.

Williams, G.P., 1999. *Chaos Theory Tamed*.

Great Britain: Joseph Henry Press.

Wills, B.A., 1979. *Mineral Processing Technology: An Introduction to the Practical Aspects of Ore Treatment and Mineral Recovery.*

London: Pergamon Press.

Wills, B.A., 1992. *Mineral Processing Technology.*

5<sup>th</sup> ed., Oxford: Pergamon Press.

Wise, B.M. and Gallagher, N.B., 1996. The process chemometrics approach to process monitoring and fault detection.

*Journal of Process Control*, 6 (6), 329-348.

Woollacott, L.C. and Eric, R.H., 1994. *Mineral and Metal Extraction: An Overview.*

Johannesburg: The South African Institute of Mining and Metallurgy.

Ynchausti, R.A. and Hales, L.B., 1992. Real-time supervisory control of grinding and flotation circuits using object oriented expert systems.

*In: S.A. Argyropoulos and G. Carayannis, eds. Artificial Intelligence in Materials Processing*, Edmonton 23-27 August 1992. Canada: Canadian Institute of Mining, Metallurgy and Petroleum, 53-61.

Zhang, J., Martin, E.B. and Morris, A.J., 1997. Process monitoring using non-linear statistical techniques.

*Chemical Engineering Journal*, 67, 181-189.

Zhu, H. and Rohwer, R., 1996. No free lunch for cross-validation.

*Neural Computation*, 8, 1421-1426.

# 10 Appendix A

---

## 10.1 Comminution

Comminution is the process by which the particle size of the ore is progressively reduced until the clean particles of mineral can be separated by such methods as are available. Milling plays a very important role in comminution and it is therefore deemed necessary to get an insight into the theory of comminution before looking at milling, and more specifically, FAG milling.

### 10.1.1 Comminution theory

Comminution theory is concerned with the relationship between the energy consumed by a grinding mill and the amount of size reduction that the consumption of this energy brought about. The greatest problem herein lies with the fact that most of the energy input to a crushing or grinding machine is absorbed by the machine itself. Only a small fraction of the total energy is therefore available for breaking the material. The amount of available energy is further reduced when the material to be crushed or grinded is plastic, causing the material to change shape without breaking and creating new surfaces. The relationship between the energy required to break the material, and the creation of new surfaces can only be made apparent if the energy consumed in creating new surfaces can be separately measured. All theories of comminution therefore assume that the material is brittle, so that no energy is absorbed in processes such as elongation or contraction that is not finally utilised in particle breakage (Lowrison, 1974).

Lynch (1977) states that it was experimentally observed that in a size reduction process, the small size change produced was proportional to the energy expended per unit weight of the particle, and that the energy required to bring about the same relative size change was inversely proportional to some function of the initial particle size. The energy-breakage relationship can thus be expressed as:

$$dE = -K \frac{dx}{x^n} \quad (10.1)$$

Many a researcher have given a different interpretation of this relationship, the oldest being that of Rittinger (1867), stating that the energy consumed in the size reduction is proportional to the area of new surface produced:

$$E = K \left[ \frac{1}{x_2} - \frac{1}{x_1} \right] \quad (10.2)$$

This theory was soon followed by that of Kick (1883) stating that the work required is proportional to the reduction in volume of the particles concerned:

$$E = K \ln \left( \frac{x_1}{x_2} \right) \quad (10.3)$$

Much later, and after extensive experimental investigation into rod and ball milling, Bond (1952) developed an equation that is based on the theory that the work input is proportional to the new crack tip length produced in particle breakage, and equals the work represented by the product minus that represented by the feed:

$$E = K \left[ \frac{1}{\sqrt{x_2}} - \frac{1}{\sqrt{x_1}} \right] \quad (10.4)$$

In the above equations,  $x_1$  is the feed size,  $x_2$  the product size (usually the 80% passing size),  $K$  some constant, and  $E$  energy.

Hukki (1961) found that Kick's law is reasonably accurate in the crushing range above approximately 1 [cm] diameter, Bond's theory applies reasonably in the range of conventional rod and ball mill grinding, and Rittinger's law applies fairly well in the fine grinding range, concluding that no single relationship is adequate to express the energy-breakage relationship. Pure energy models therefore only provide a useful gross description of total breakage. These models do not consider particle transport and energy expenditure that does not result in breakage, and the underlying assumption that the shape of the size distribution remains relatively constant, regardless of breakage history, is often erroneous. However, owing to the evolution

of the digital computer, more advanced comminution models have been developed. In order to better comprehend these models, a greater understanding of milling is required.

### **10.1.2 Tumbling mills**

The principal purpose of milling is to achieve the correct degree of mineral liberation, increasing the surface area of the valuable minerals, to assist further hydrometallurgical methods of treatment. The grinding mechanisms taking place inside a mill include impact or compression (forces applied almost normally to the particle surface), chipping (oblique forces), and abrasion (forces acting parallel to the surfaces). They are typically influenced by the size, quantity, the type of motion and the spaces between the individual pieces of the medium in the mill and result from the rotation of the mill about its axis. Due to the friction of the mill shell during rotation, the grinding medium is lifted along the rising side of the mill until reaching a position of dynamic equilibrium. From here the particles cascade and cataract down the free surface of the other particles, about a dead zone where little movement occurs, down to the toe of the crop load, resulting in grinding. If the particles are sufficiently distorted by the grinding mechanisms, they will break, leading to mineral liberation.

An important factor to consider in mill operation is therefore the speed at which the mill is run, governing the nature of the product and the amount of wear on the shell liners. During milling, three states can occur within the mill, namely, cascading, cataracting, or centrifuging. At low rotational speeds, cascading takes place. The grinding medium remain in contact with one another in the body of the load, leading to a finer grind with increased slimes production and increased liner wear. At higher speeds cataracting occurs. The grinding medium begin to be ejected from the main body of the load and are thrown into free flight before impacting on the toe of the load or on the mill liners, leading to a coarser end product and, generally, reduced liner wear. At even higher speeds centrifuging arises. When the mill load begins to centrifuge, the load, and relative movement between the medium, particles, and mill liners is insignificant, resulting in a drastic reduction in grinding. The speed at which centrifuging occurs is termed the critical speed of the mill. Generally mills are operated at fixed speeds between 60 and 92% of the mills critical speed.

In a wet-grinding circuit, another important consideration is the water-to-solids ratio. This ratio not only affects the viscosity of the slurry in the mill, affecting the extent of grinding in the mill, but also the grinding action in the mill. The water flow rate to a wet mill is therefore usually ratioed to the flow rate of solids, maintaining an optimum pulp density.

As stated in the introduction, the milling types most frequently used is rod, ball, autogenous, pebble, and semi-autogenous milling. Since the focus of this thesis is on the modelling and control of an autogenous mill, autogenous milling will be discussed in more detail.

### **10.1.3 Autogenous milling**

Autogenous mills operate on the same principles as ball mills, but with large lumps of material being in the feed and reduced material being capable of being broken further by the lumps. The wearing away and breakage of the grinding medium therefore now constitute part of the size-reduction process.

Three broad categories of autogenous mills exists: drum type, cascade type, and flighted or Aerofall type. The drum type autogenous mill is simply a ball mill without any foreign media, whereas the cascade type autogenous mill is a drum type with a very large diameter-to-length ratio of up to 3 or 4 to 1. On the other hand, the Aerofall type autogenous mill is a cascade type with flights set around the periphery or near the circumference of the sides of the drum. These mills are often run in an open circuit, however, even in this case a coarse classifier, such as a trommel attached to the mill or a vibrating screen, will be used to classify the mill product. Should a finer mill product be required, the mill can be operated in a closed circuit with a fine classifier at the cost of having the circuit operating at a lower throughput capacity.

Since grinding is accomplished through the impact of larger pieces of material on smaller pieces of equal density, the bed of material in the mill has to be kept shallow to avoid cushioning and buoyancy. The mill load is therefore usually kept below 30% of the effective mill volume. For successful autogenous milling, it is further essential that the top size range of the ore be capable of generating a grinding load able to crush the finer portion of the feed plus its own progeny but it must not be so durable that it

cannot be reduced at a rate equal to that at which feed it is entering the mill. Autogenous mills are also operated at higher speeds than is usual in conventional milling, with speeds of the order of 80-85% of the critical speed generally agreed upon as optimal (Lowrison, 1974). This increased speed promotes impact breakage, usually leading to an increase in coarse rock breakage, which in turn leads to higher throughputs.

#### 10.1.3.1 Advantages

The advantages of autogenous milling are numerous (Lowrison, 1974; Woollacott and Eric, 1994; Gill, 1991). Some of the more important advantages include:

- Feed material much coarser than that used for conventional fine grinding (e.g. 25 [cm]) can be fed to an autogenous mill, often directly from a primary crusher, offering a very cheap, one-step, size reduction process, with an evenly-graded, uncontaminated product, reducing the feed in a single stage to a finely ground product (e.g. 0.1 [mm]).
- Autogenous mills draw less power per ton of ore processed relative to conventional ball mills. This is a direct result from the fact that most minerals are less dense than steel balls, increasing milling effectiveness and reducing the energy required in lifting the material in the mill. The absence of steel grinding media in autogenous mills also lead to reduced operating costs. Although the capital cost of autogenous mills is higher than conventional ball mills, there is no need for a fine crushing circuit, saving in turn on capital and operating cost.
- Owing to increased boundary breakage achieved in autogenous milling when compared to conventional mills, a flotation feed is produced that usually permits a higher percentage of value recovery. This is a direct result from the feed having a grain size that more nearly approaches the natural material grain size found in the ore.

- Because the power demands of autogenous mills are more sensitive to the load of material than other tumbling mills, they are more amendable to automatic control of feed based upon power demand.

#### 10.1.3.2 Disadvantages

Autogenous mills have disadvantages in some areas (Lowrison, 1974; Woollacott and Eric, 1994):

- Due to the fact that the hardness, abrasive and fracture properties of an ore could vary significantly within an orebody, the properties of the grinding media are not as predictable as with steel grinding media.
- Not all material responds well to autogenous milling. Material of a friable, grainy nature is generally best, but careful trials with truly representative samples of the actual material to be processed are essential before the process is to be adopted.
- Control in autogenous mills are made difficult owing to the need to vary the throughput to reduce the effects caused by changes in ore hardness or any other size problems. The results in greater adjustment of feed tonnages being required to avoid overgrinding of soft ores and undergrinding of hard ores as grindability of the feed fluctuates. One solution to this problem is the blending of the run-of-mine material in the stockpile prior to milling.
- Autogenous mills also have more operation variables than conventional ball mills.

#### 10.1.3.3 Control

As with the control of conventional milling, the principal variables that can affect the control of an autogenous mill are ore feed rate, water addition rate, ore feed hardness, and ore feed size analysis. Ore feed rate and water addition rate are variables that may be deliberately manipulated in order to control the process, whereas ore feed hardness and size analysis are disturbance variables, responsible for random disturbances in the process feed, and have consistently be found to be the two single

largest problems encountered when attempting to run a stable mill under optimum conditions (Napier-Munn et al., 1999). Step changes performed on a rock-mill – pebble-mill circuit (Lynch, 1977) showed that both an increase in the fine ore feed rate and an increase in the mill power set point results in a coarser product to be produced. The grinding medium charge in the mill can thus be controlled by observing the power consumption of the mill, once the power consumption falls below a certain level, fresh grinding medium should be added to the mill.

It was also found that a strong relationship exists between the charge volume and the power drawn by the mill, both being affected by the mill feed rate. Both an increase in mill feed rate and an increase in ore hardness will result in a rise in charge volume leading to an increase in power drawn by the mill. Power draw therefore does not change in direct response to feed rate, ore hardness or any other external influence but in accordance with the mill charge (Napier-Munn et al., 1999).

When considering feed size and ore hardness it is important to note that since autogenous mills rely exclusively on the feed ore for their grinding media, insufficient large rock grinding media will lead to accumulation of critical sized material in the mill, seriously affecting throughput. Soft ore will result in insufficient large grinding media being present in the mill, whereas too hard an ore may possess insufficient energy to break the critical size ore.

From a control perspective, advanced model based control could improve the conditions of continuous operation when these changes in the circuit feed must be compensated for, preventing undesirable changes in the circuit product.

## **10.2 Process variable classification**

As stated in chapter 2, all process variables can be classified as either input variables, being physical variables affecting the process outputs, or output variables, being variables associated with exit streams or conditions inside the process. Whereas input variables can be subdivided into manipulated and disturbance variables, output variables can be subdivided into controlled and measure variables. The reason for some of the output variables possibly being classified as measured variables is due to

the fact that it is not always feasible to control all of the output variables (Eksteen and Reuter, 2000):

- On-line measurement of all outputs not being possible or economical.
- Insufficient manipulated variables available.
- Controlling certain output variables may result in impractical control loop resulting from slow dynamics, interactions between control loops or a low sensitivity to available manipulated variables.

Eksteen and Reuter (2000) also give the following guidelines for selecting controlled variables from available output variables and selecting manipulated variables from available input variables:

- Selecting controlled variables:
  - Controlled variables should not be self-regulating.
  - Output variables that could exceed the equipment or operating constraints should be controlled.
  - Output variables that are a direct measure of product quality, or can strongly affect it, should be controlled.
  - Controlled variables should seriously interact with each other.
  - Output variables having favourable dynamic and static characteristics should be considered as controlled variables.
- Selecting manipulated variables:
  - Manipulated variables should have a large effect on the controlled variables.
  - Input variables that are able to rapidly affect the controlled variables should be considered as manipulated variables.
  - Manipulated variables should have a large enough range, increasing the likelihood of good control.
  - Input variables that effect the controlled the controlled variables directly should be considered as manipulated variables.
  - Recycling disturbances should be avoided.

### 10.3 Lyapunov spectrum

As stated in chapter 3, the method used to calculate the Lyapunov exponents for this research is based on the work of Sano and Sawada (1985). The Lyapunov exponents are calculated as follow:

$$\lambda_i = \lim_{n \rightarrow \infty} \frac{1}{n\tau} \sum_{j=1}^n \ln \|A_j e_i^j\| \quad (10.5)$$

for  $i = 1, 2, \dots, d$ , where  $\{e_i^j\}$  ( $i = 1, 2, \dots, d$ ) is a set of basis vectors of the tangent space at  $x_j$  and

$$A_j V = C \quad (10.6)$$

where  $V$  and  $C$  are  $d \times d$  covariance matrices:

$$V_{kl} = \frac{1}{N} \sum_{i=1}^N y^{ik} y^{il} \quad (10.7)$$

$$C_{kl} = \frac{1}{N} \sum_{i=1}^N z^{ik} y^{il} \quad (10.8)$$

where  $y^{ik}$  and  $z^{ik}$  are the  $k$  components of vectors  $y_i$  and  $z_i$  as defined by Sano and Sawada (1985).

### 10.4 Correlation and correlation coefficients

The most widely used type of correlation coefficient is the Pearson R, also called the linear or product-moment correlation. The correlation coefficient determines the extent or strength to which values of two variables are proportional (linearly related) to each other and does not depend on the specific measurement units used. The correlation between two variables is therefore high if a straight line can be used to approximate it. This line, called the regression or least squares line, is determined such that the sum of the squared distances of all the data points from the line is the

lowest possible. Whereas a correlation coefficient of +1 represents a perfect positive correlation (when the one variable increases the other variable also increases), -1 represents a perfect negative correlation (when the one variable increases the other variable decreases), and 0 represent a lack of correlation.

Squaring the correlation coefficient produces the coefficient of determination,  $R^2$ , which represents the proportion of common variation in the two variables (the strength or magnitude of the relationship). This, together with the significance of the correlation is used to evaluate the correlation between two variables. However, it should be noted that depending on the size of the sample from which the correlation coefficient of a particular magnitude was calculated, its significance would change; the larger the sample size the smaller the effect that can be proven significant in the sample. It is furthermore assumed that the distribution of the residual values from the dependent variable follows a normal distribution and the variability of the residual values is the same for all values of the independent variable. Although it is generally accepted that for a sample size of over 100 samples one should not be concerned with the normality assumptions, other common threats to the validity of information that a correlation coefficient can provide can still occur. These threats include outliers, non-homogenous groups and non-linear relations between variables.

### 10.5 Fourier analysis

The Fourier transform is simply a method of expressing a function in terms of the sum of its projections onto a set of basis functions. This decomposition of a time series into a sequence of sines and cosines of different frequencies and amplitudes is accomplished similarly to the decomposition of a point in Euclidean space into the sum of its basis vector components e.g. given the vector  $\mathbf{v}$  in 3-dimensional Euclidean space:

$$\mathbf{v} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k} \quad (10.9)$$

The components of  $\mathbf{v}$  in each of the  $\mathbf{i}$ ,  $\mathbf{j}$  and  $\mathbf{k}$  directions can be calculated using the dot product of  $\mathbf{v}$  and each of the basis vectors:

$$\begin{aligned}
 x &= \mathbf{v} \cdot \mathbf{i} \\
 y &= \mathbf{v} \cdot \mathbf{j} \\
 z &= \mathbf{v} \cdot \mathbf{k}
 \end{aligned}
 \tag{10.10}$$

For a function,  $f(x)$ , the sine transform is defined by:

$$F_s(s) = 2 \int_0^{\infty} f(x) \sin(2\pi s x) dx \tag{10.11}$$

where  $s$  characterises the frequency of the sinusoid and is equal to the number of cycles per unit of  $x$ . Similarly, the cosine transform of  $f(x)$  is defined by:

$$F_c(s) = 2 \int_0^{\infty} f(x) \cos(2\pi s x) dx \tag{10.12}$$

When combining the sine and cosine transforms of  $f(x)$ , the Fourier transform of the whole of  $f(x)$  is found:

$$\begin{aligned}
 F(s) &= \frac{1}{2} F_c(s) - \frac{1}{2} i F_s(s) \\
 &= \int_{-\infty}^{\infty} f(x) \exp(-i2\pi s x) dx
 \end{aligned}
 \tag{10.13}$$

However, numerical computation of the Fourier transform of  $f(t)$  requires discrete sample values,  $f_k$ , and inversely can only provide discrete samples of the transform,  $F_r$ . Therefore, given that  $F(rs_o)$  is the  $r^{\text{th}}$  sample of  $F(s)$ , then:

$$F_r = F(r2\pi T_0^{-1}) \tag{10.14}$$

and the discrete Fourier transform (DFT) is defined as:

$$F_r = \sum_{k=0}^{N_0-1} f_k \exp(-ir2\pi N_0^{-1}k) \quad (10.15)$$

where  $N_0$  is the number of samples in the signal in one period  $T_0$ . A popular DFT algorithm is the Fast Fourier Transform (FFT). When analyzing the power spectral density plots obtained from Fourier analysis, peaks in the data are indicative of oscillatory behavior at that frequency. Furthermore, the power spectral density plot should exhibit some decay over the frequency range with clearly distinguishable peaks from the normal decay in order to be considered significant.

## 11 Appendix B – Additional tables and figures

---

### 11.1 Model validation test datasets

Below is listed the statistics that summarises the test datasets used during model validation and multivariate statistical process monitoring.

**Table 11.1: Summary of statistics calculated for test dataset 1 (6571 observations)**

Variable	Mean	Min	Max	Std Dev
x <sub>1</sub>	2826.71	1359.95	3763.02	263.03
x <sub>2</sub>	147.65	128.47	184.03	9.94
x <sub>3</sub>	56.96	0.00	80.00	5.40
x <sub>4</sub>	345.14	0.12	400.00	71.00
x <sub>7</sub>	63.30	0.17	189.32	16.72
x <sub>9</sub>	253.29	1.16	1000.00	133.11
x <sub>10</sub>	71.56	2.03	590.28	134.58
x <sub>11</sub>	207.38	0.58	582.75	184.22
x <sub>12</sub>	631.18	1.45	1000.0	364.48

**Table 11.2: Summary of statistics calculated for test dataset 2 (6091 observations)**

Variable	Mean	Min	Max	Std Dev
x <sub>1</sub>	2359.81	1047.45	3242.19	404.73
x <sub>2</sub>	133.67	108.80	156.25	8.82
x <sub>3</sub>	61.25	6.04	80.00	4.33
x <sub>4</sub>	208.61	0.12	400.00	122.82
x <sub>7</sub>	44.32	0.35	202.69	18.28
x <sub>9</sub>	59.26	0.58	504.92	122.24
x <sub>10</sub>	255.91	1.45	601.56	131.16
x <sub>11</sub>	355.76	116.03	750.87	74.05
x <sub>12</sub>	1.58	0.29	26.33	0.35

**Table 11.3: Summary of statistics calculated for test dataset 3 (4501 observations)**

Variable	Mean	Min	Max	Std Dev
x <sub>1</sub>	2764.54	1568.29	3531.54	326.41
x <sub>2</sub>	146.60	122.69	179.40	8.77
x <sub>3</sub>	53.85	50.00	70.00	4.79
x <sub>4</sub>	317.92	0.12	400.00	82.05
x <sub>7</sub>	36.98	0.17	70.92	9.09
x <sub>9</sub>	12.23	0.00	1000.00	94.64
x <sub>10</sub>	282.83	18.81	611.11	42.49
x <sub>11</sub>	263.79	0.29	660.88	168.99
x <sub>12</sub>	121.25	0.29	1000.00	211.65

**11.2 Datasets used for multivariate statistical process monitoring**

Below is listed the statistics that summarises the datasets used for multivariate statistical process monitoring.

**Table 11.4: Summary of statistics calculated for reference dataset (1071 observations)**

Variable	Mean	Min	Max	Std Dev
x <sub>1</sub>	2899.01	2358.22	3352.14	148.94
x <sub>2</sub>	156.08	149.31	164.35	4.16
x <sub>4</sub>	349.87	183.33	400.00	51.96
x <sub>7</sub>	39.85	24.74	61.28	4.73

**Table 11.5: Summary of statistics calculated for test dataset 1 (1080 observations)**

Variable	Mean	Min	Max	Std Dev
x <sub>1</sub>	2424.20	1669.56	2750.29	148.09
x <sub>2</sub>	132.34	114.58	138.89	3.77
x <sub>4</sub>	275.13	0.12	400.00	69.28
x <sub>7</sub>	62.19	6.25	88.98	11.06

**Table 11.6: Summary of statistics calculated for test dataset 2 (1091 observations)**

Variable	Mean	Min	Max	Std Dev
x <sub>1</sub>	2677.42	1587.09	3253.76	299.03
x <sub>2</sub>	146.76	123.84	178.24	8.67
x <sub>4</sub>	299.79	0.12	400.00	65.97
x <sub>7</sub>	45.09	0.35	115.36	12.61

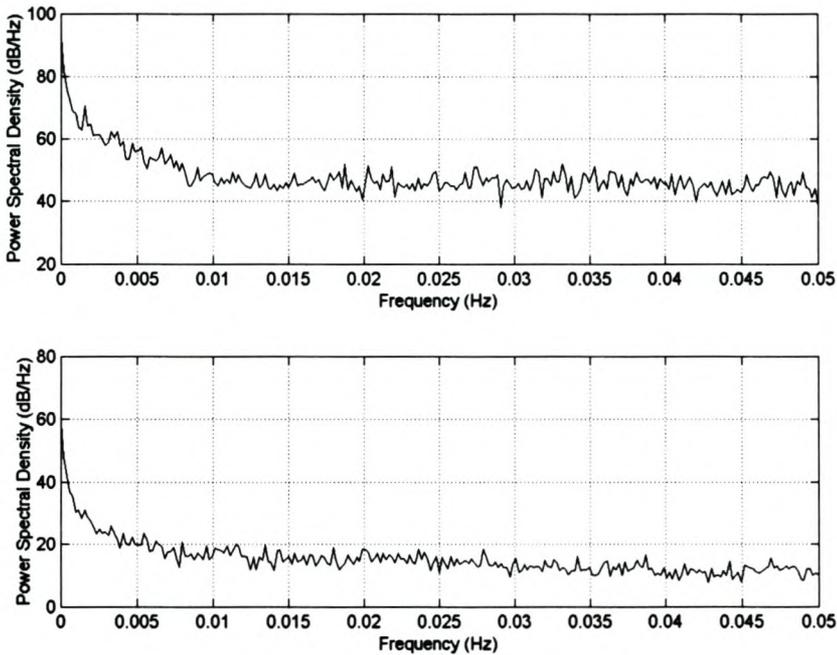
**Table 11.7: Summary of statistics calculated for test dataset 3 (1201 observations)**

Variable	Mean	Min	Max	Std Dev
$x_1$	2762.13	2223.67	3214.70	201.24
$x_2$	144.70	123.84	162.04	7.41
$x_4$	352.67	0.12	400.00	66.40
$X_7$	32.20	7.03	83.07	5.76

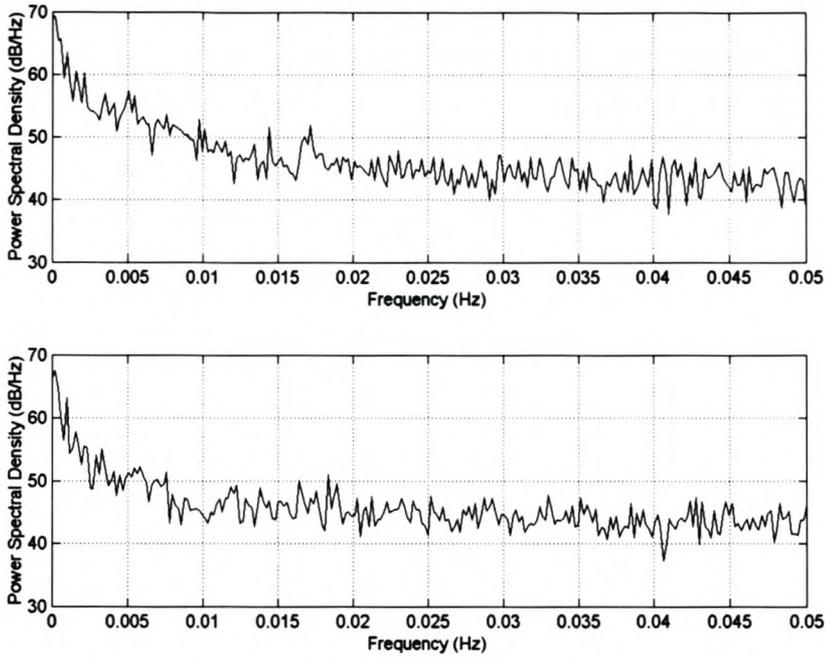
### 11.3 Additional figures

Shown below are figures that were not included in the main body of this thesis, but were used during data analysis.

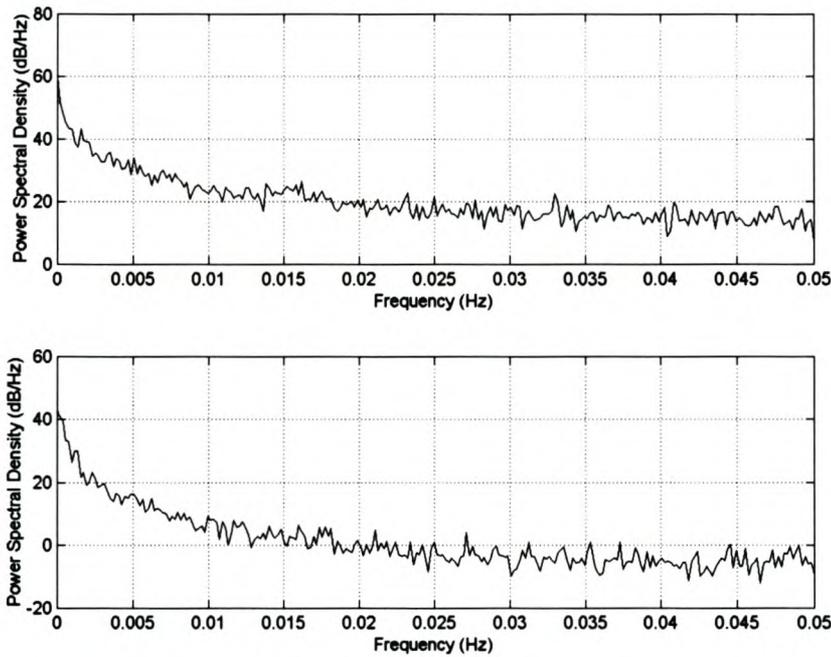
#### 11.3.1 Fourier power spectrum plots



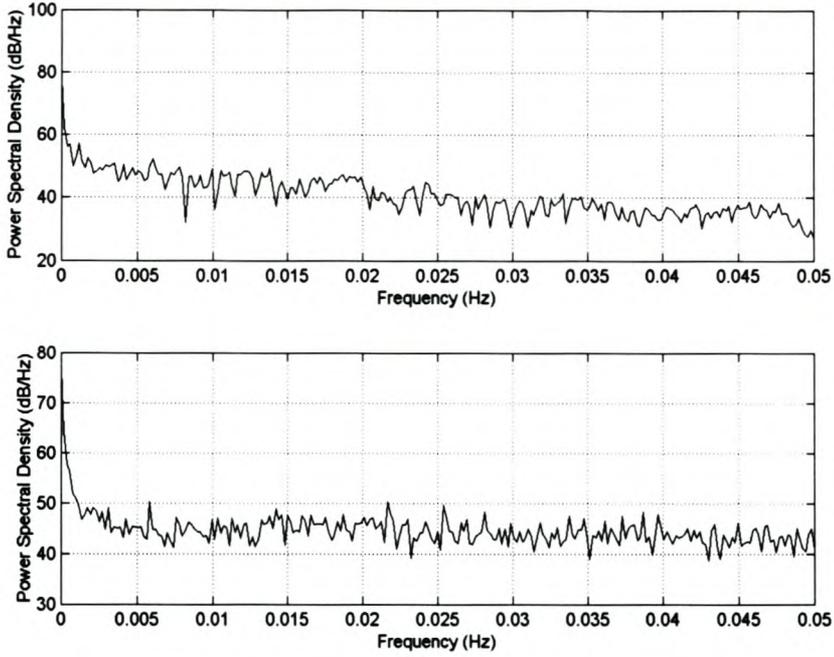
**Figure 11.1: Fourier power spectrum for FAG mill power ( $x_1$ ) and load ( $x_2$ )**



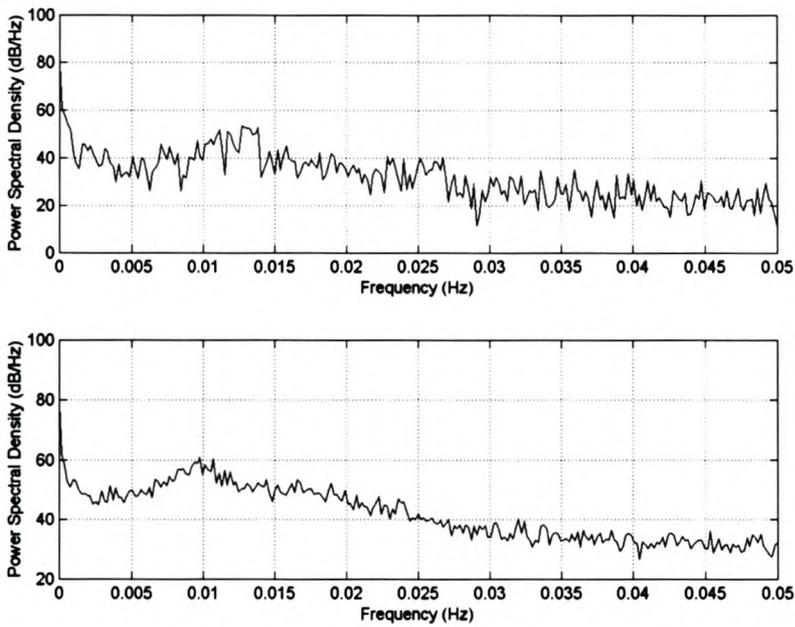
**Figure 11.2: Fourier power spectrum for FAG mill total screen feed rate ( $x_5$ ) and feed rate ( $x_6$ )**



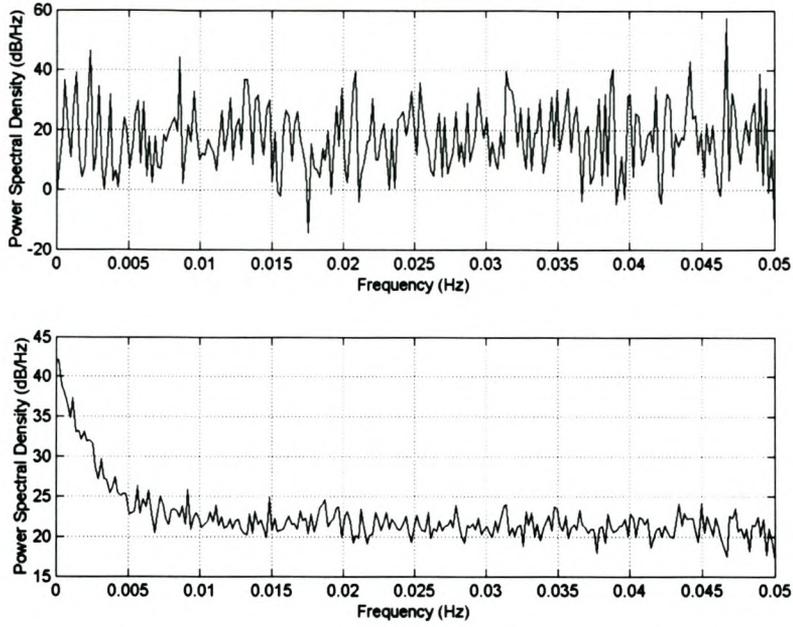
**Figure 11.3: Fourier power spectrum for FAG mill inlet water flow rate ( $x_7$ ) and motor reference speed ( $x_8$ )**



**Figure 11.4: Fourier power spectrum for FAG mill discharge 1 flow rate ( $x_9$ ) and discharge 2 flow rate ( $x_{10}$ )**

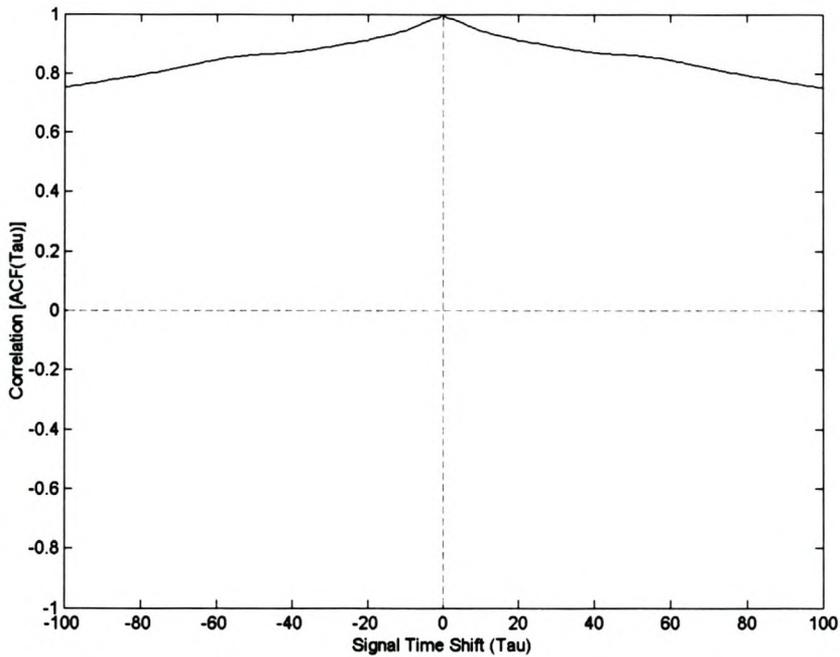


**Figure 11.5: Fourier power spectrum for FAG mill rougher feed 1 flow rate ( $x_{11}$ ) and rougher feed 2 flow rate ( $x_{12}$ )**

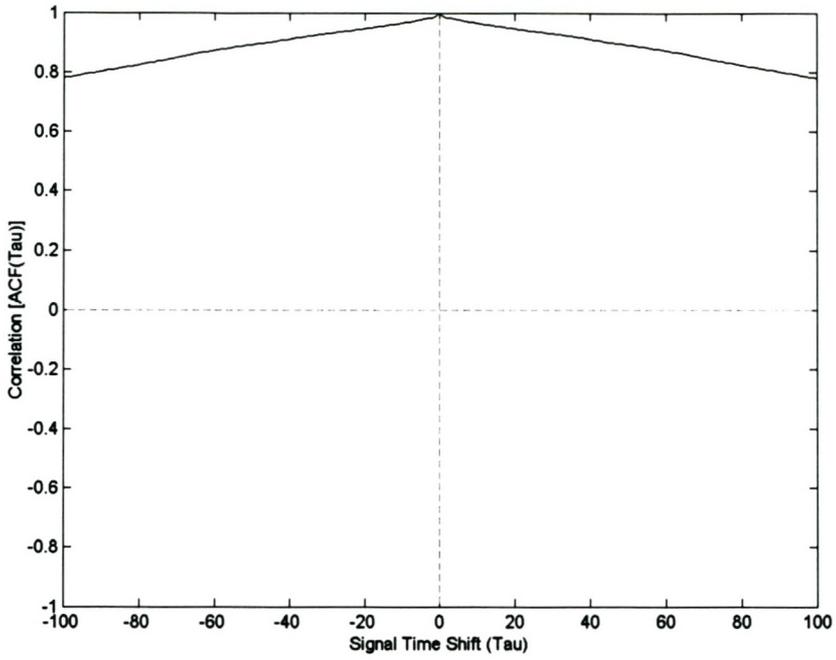


**Figure 11.6: Fourier power spectrum for FAG mill feed size distribution % – 31.5mm ( $x_{13}$ ) and product size distribution % -75 $\mu$ m ( $x_{14}$ )**

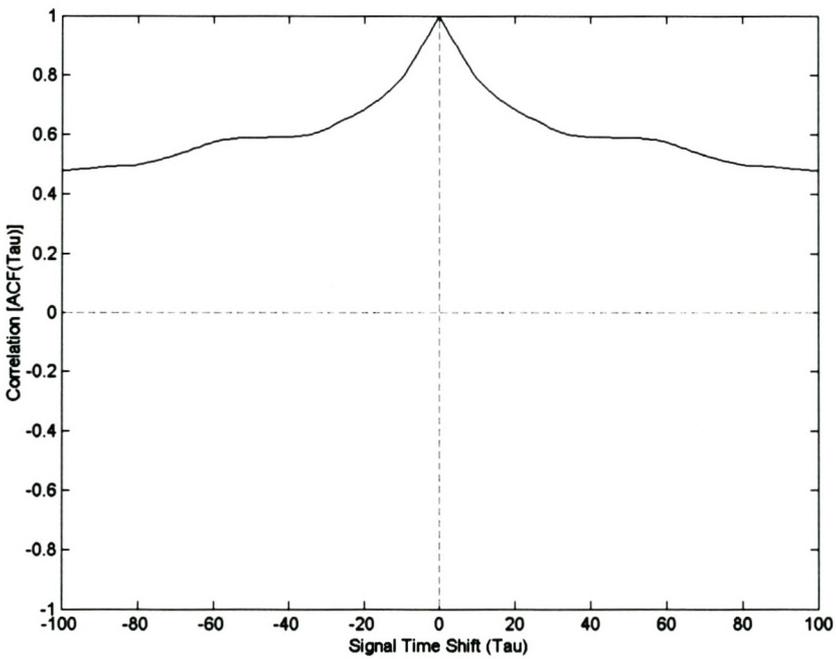
### 11.3.2 Linear autocorrelation plots



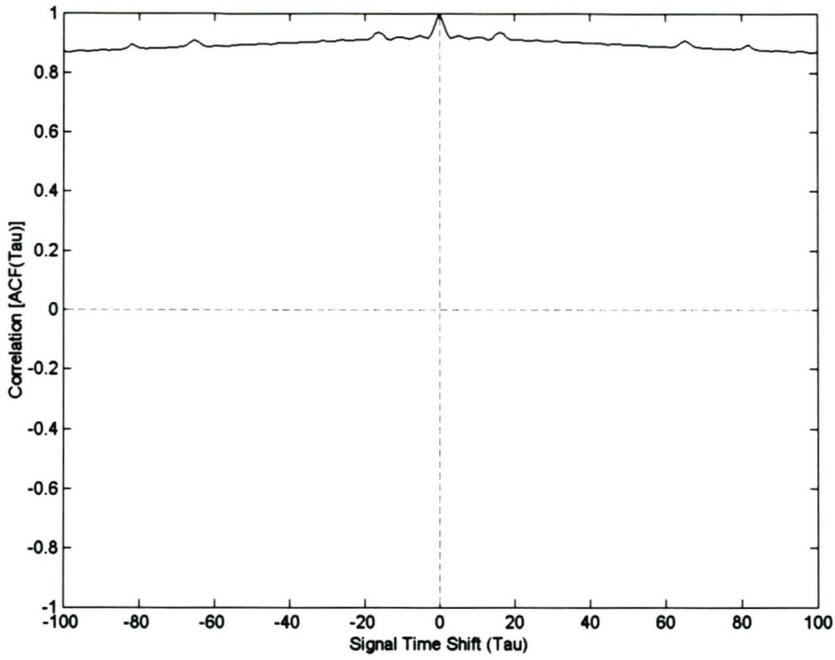
**Figure 11.7: Autocorrelation for FAG mill power ( $x_1$ )**



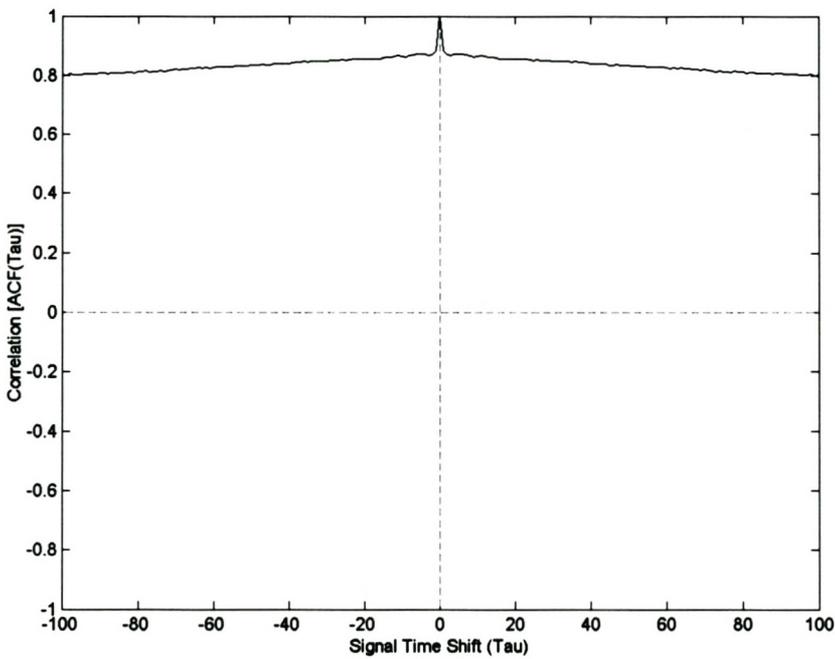
**Figure 11.8: Autocorrelation for FAG mill load ( $x_2$ )**



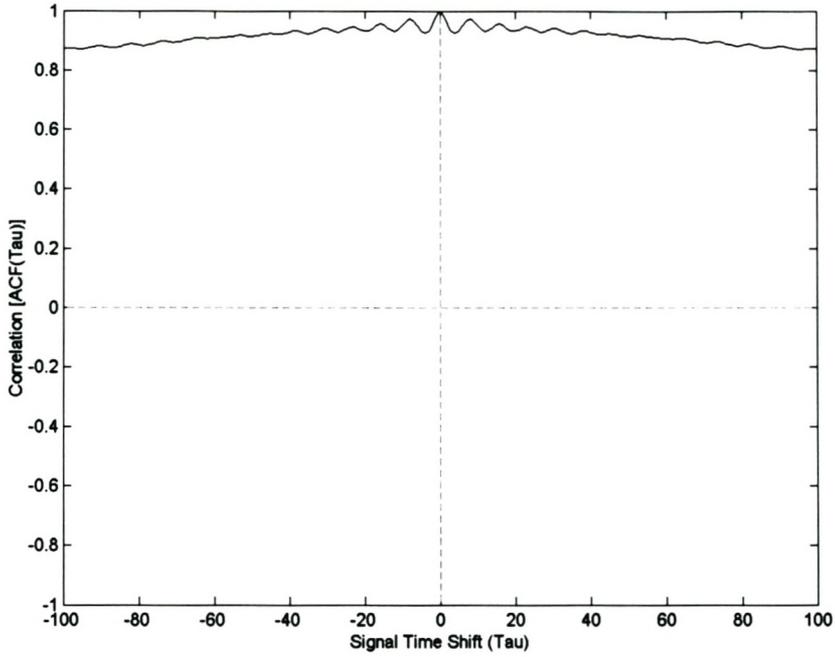
**Figure 11.9: Autocorrelation for FAG mill inlet water flow rate ( $x_7$ )**



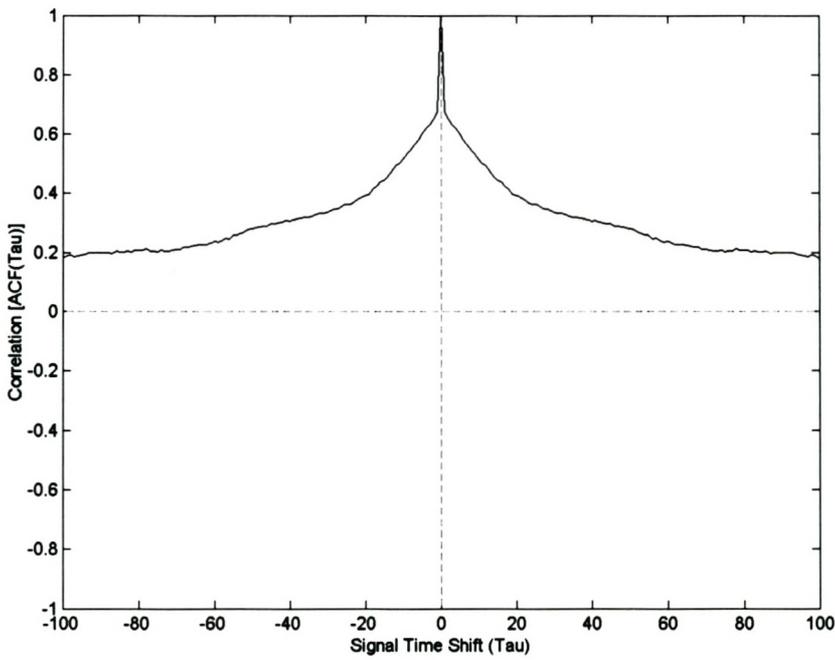
**Figure 11.10: Autocorrelation for FAG mill discharge 1 flow rate ( $x_9$ )**



**Figure 11.11: Autocorrelation for FAG mill discharge 2 flow rate ( $x_{10}$ )**



**Figure 11.12: Autocorrelation for FAG mill rougher feed 1 flow rate ( $x_{11}$ )**



**Figure 11.13: Autocorrelation for FAG mill feed size distribution %  $-31.5\text{mm}$  ( $x_{13}$ )**

### 11.3.3 Average Mutual Information plots

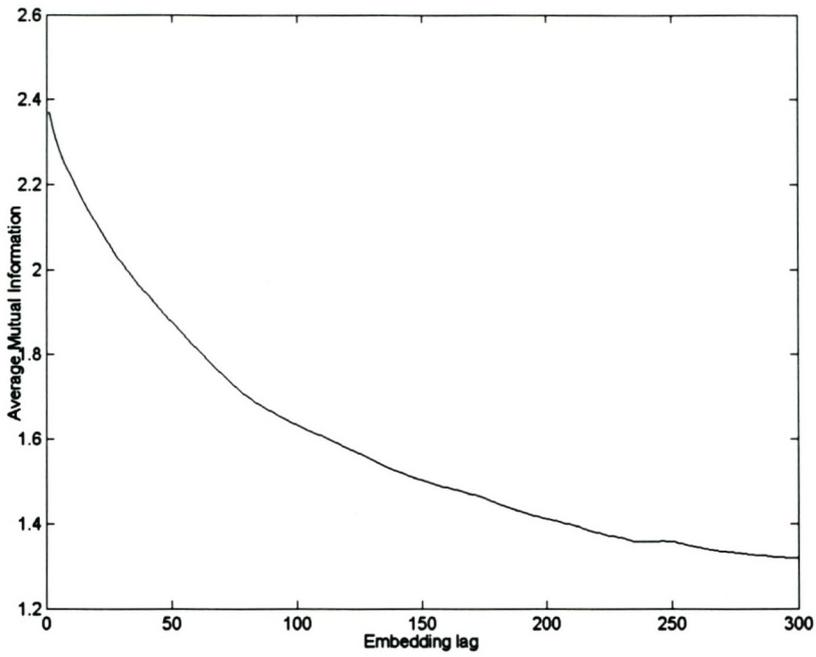


Figure 11.14: AMI plot of FAG mill coarse feed rate ( $x_3$ )

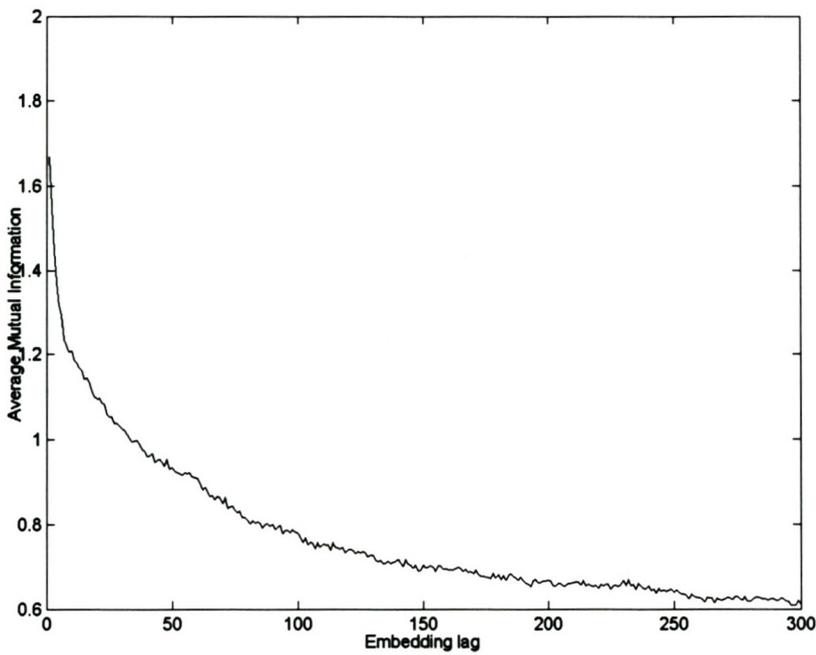
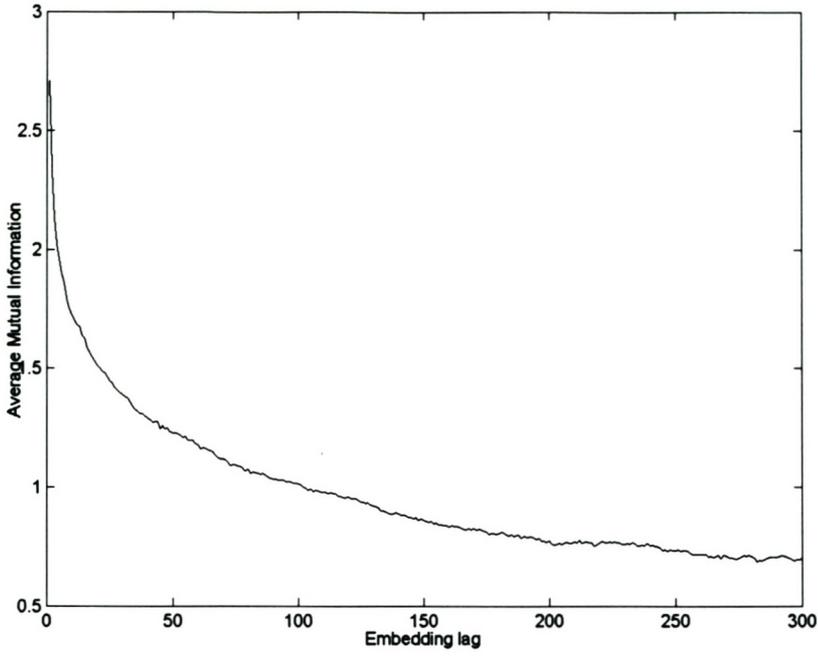
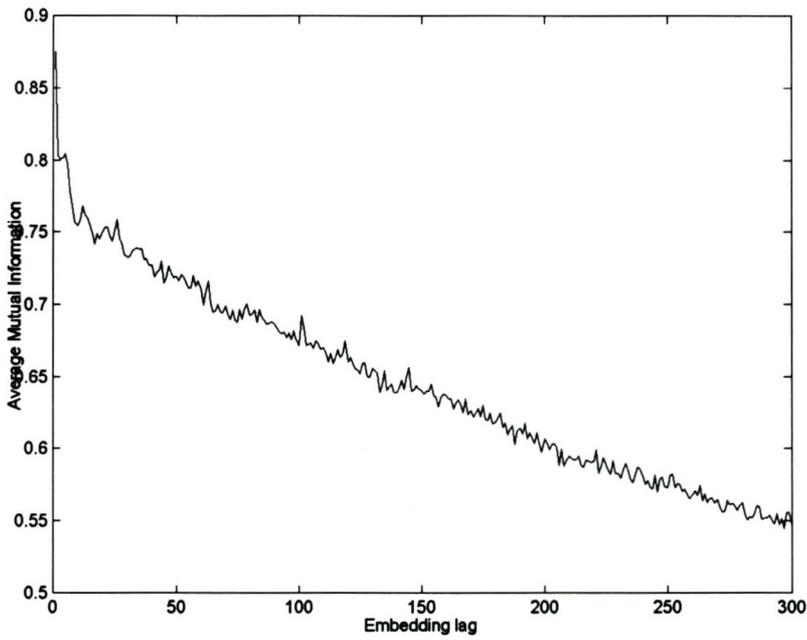


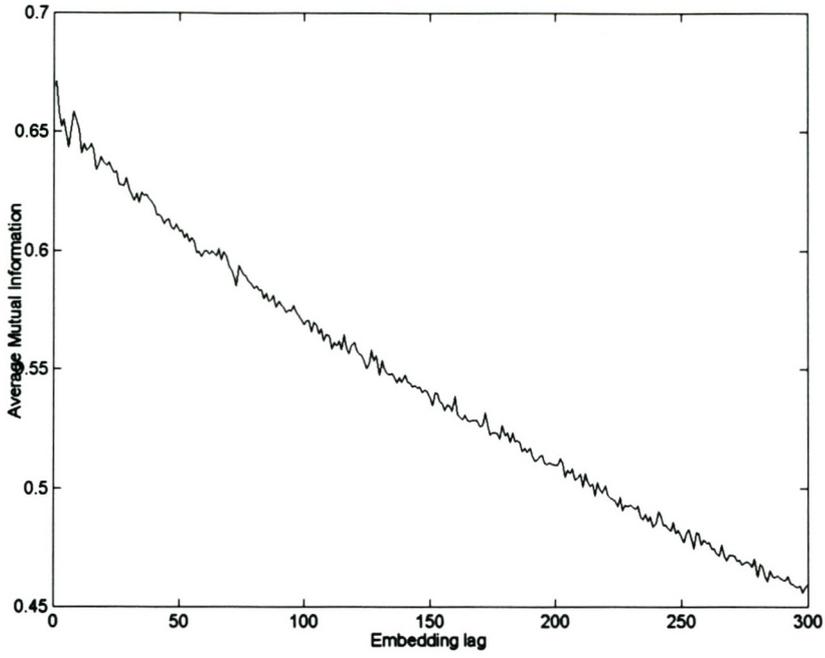
Figure 11.15: AMI plot of FAG mill fine feed rate ( $x_4$ )



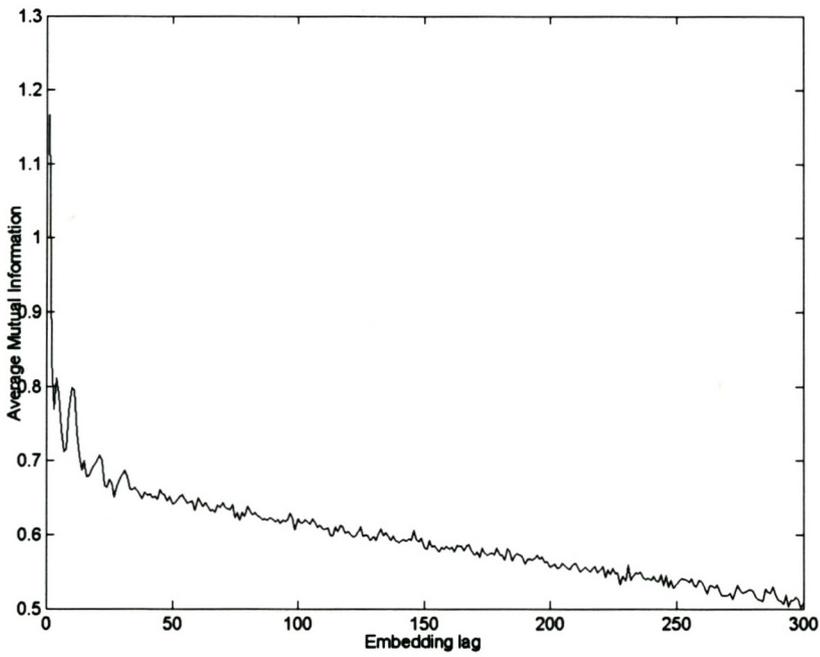
**Figure 11.16: AMI plot of FAG mill inlet water flow rate ( $x_7$ )**



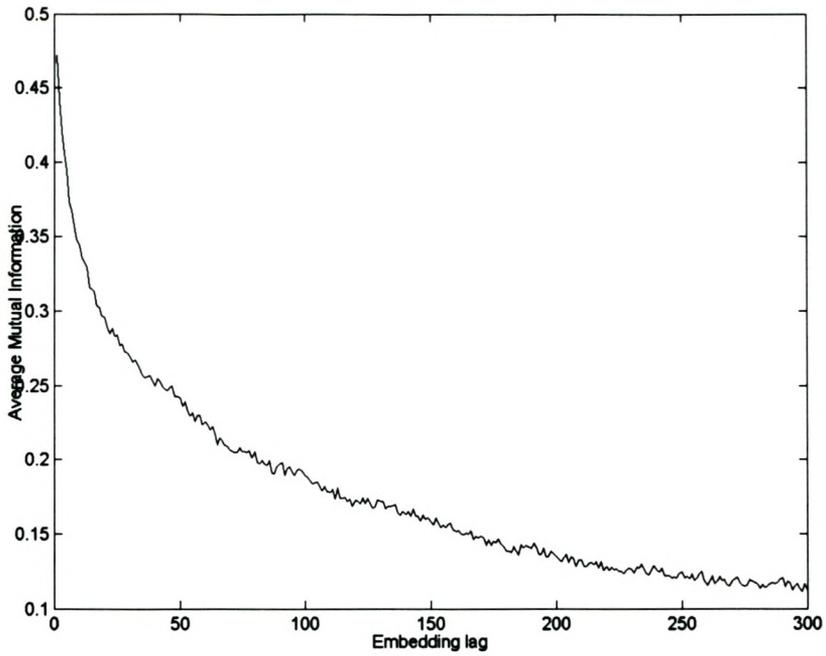
**Figure 11.17: AMI plot of FAG mill discharge 2 flow rate ( $x_{10}$ )**



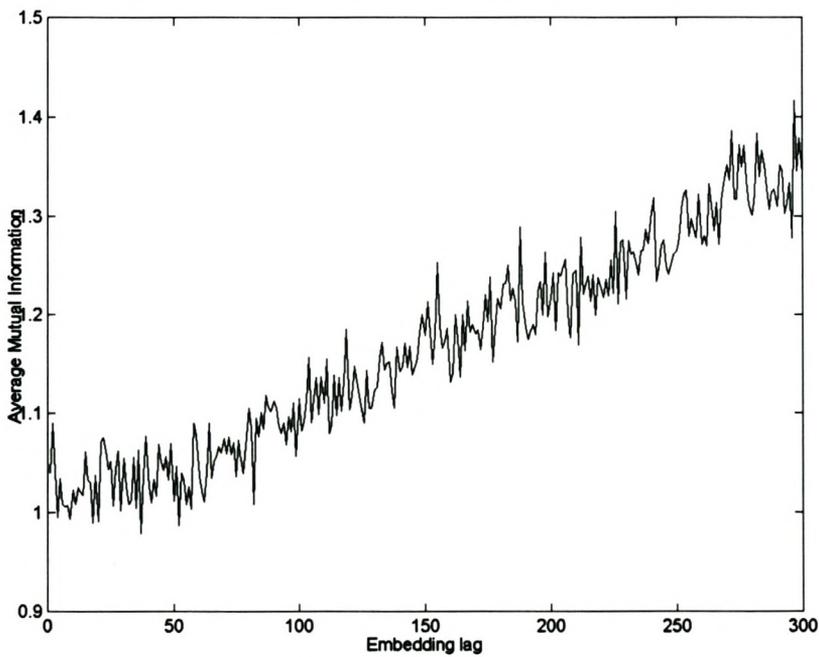
**Figure 11.18: AMI plot of FAG mill rougher feed 1 flow rate ( $x_{11}$ )**



**Figure 11.19: AMI plot of FAG mill rougher feed 2 flow rate ( $x_{12}$ )**



**Figure 11.20: AMI plot of FAG mill feed size distribution % -31.5mm ( $x_{13}$ )**



**Figure 11.21: AMI plot of FAG mill product size distribution % -75 $\mu$ m ( $x_{14}$ )**

### 11.3.4 False Nearest Neighbours plots

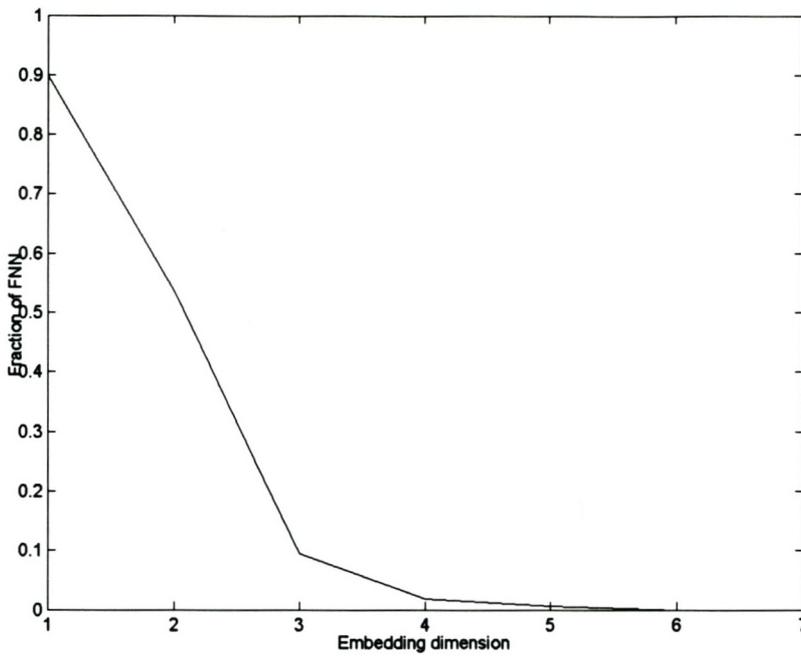


Figure 11.22: FNN plot of FAG mill power ( $x_1$ ) with  $k = 100$

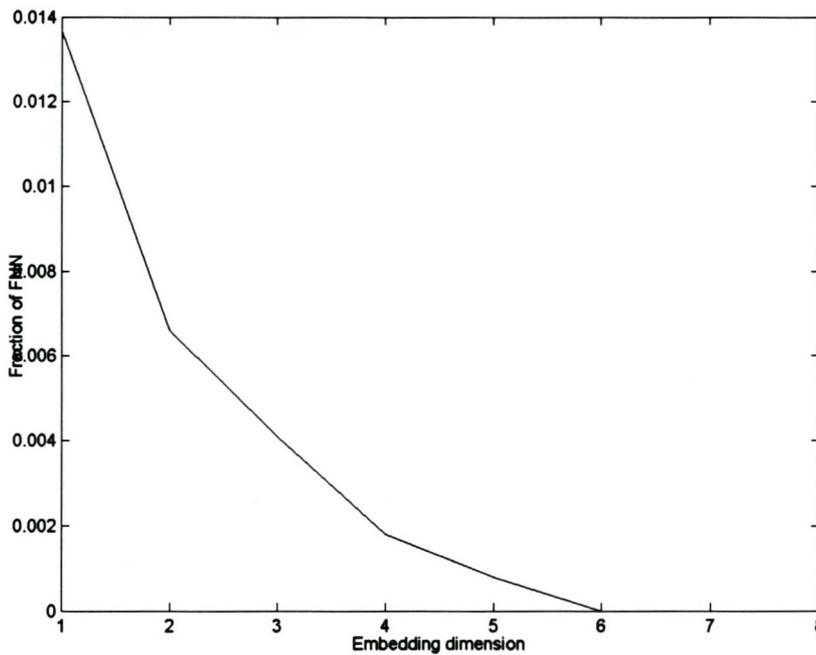
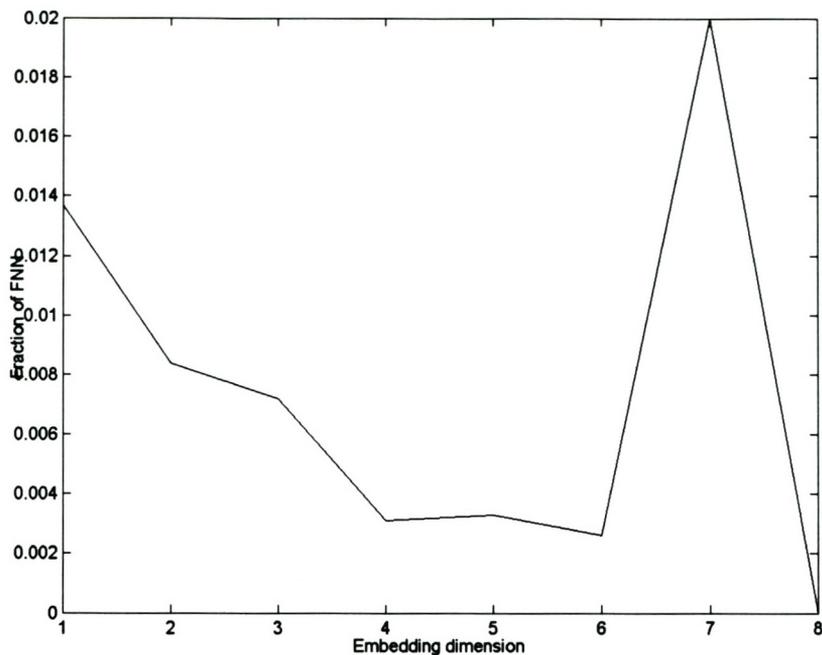
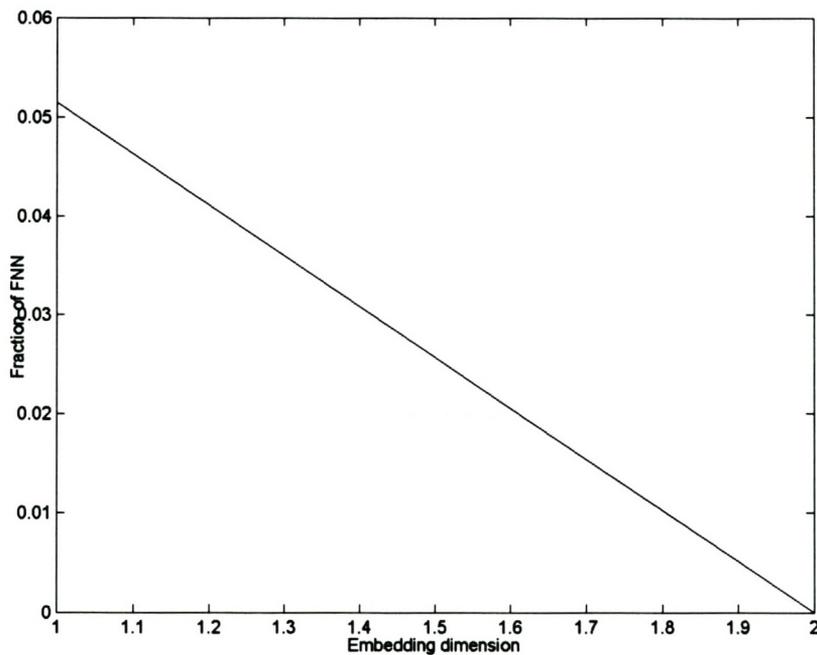


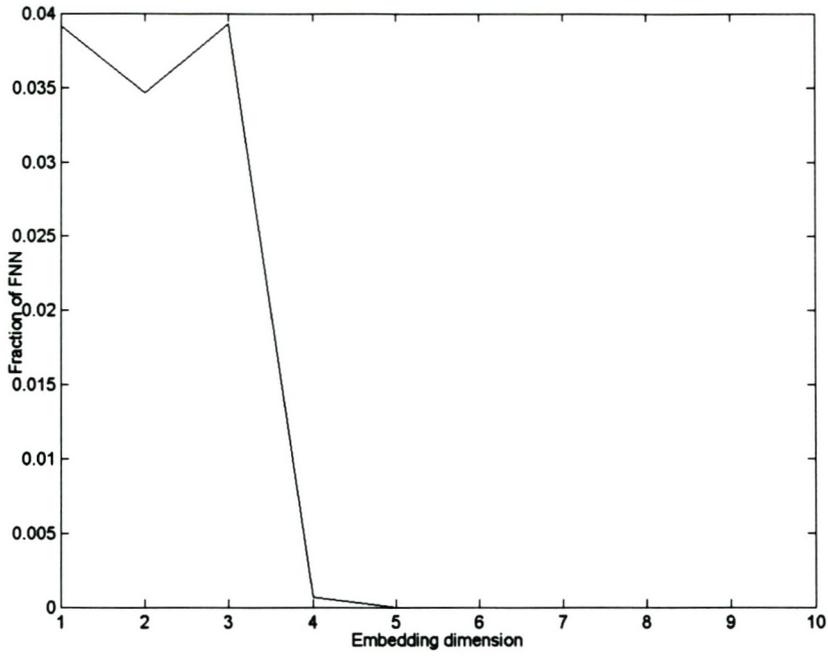
Figure 11.23: FNN plot of FAG mill load ( $x_2$ ) with  $k = 106$



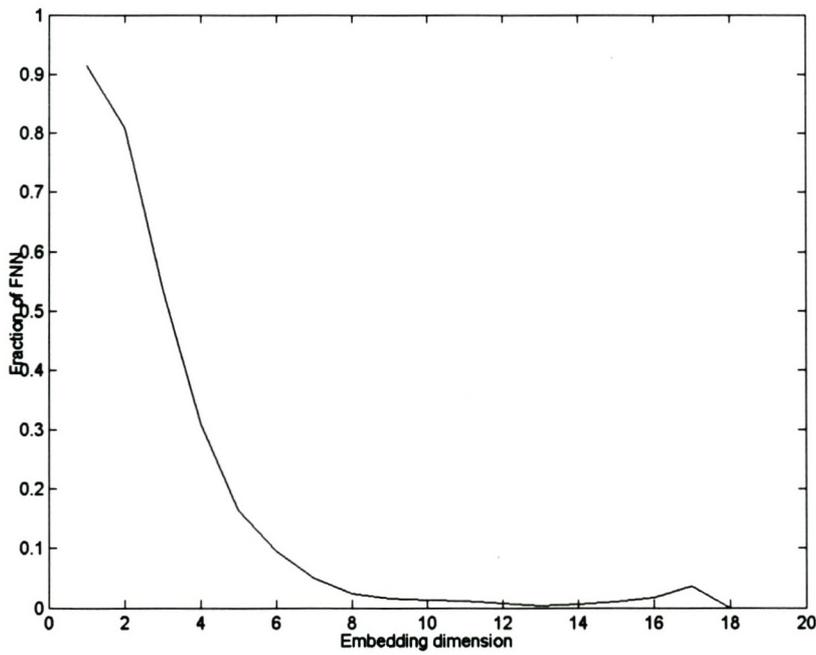
**Figure 11.24: FNN plot of FAG mill load ( $x_2$ ) with  $k = 100$**



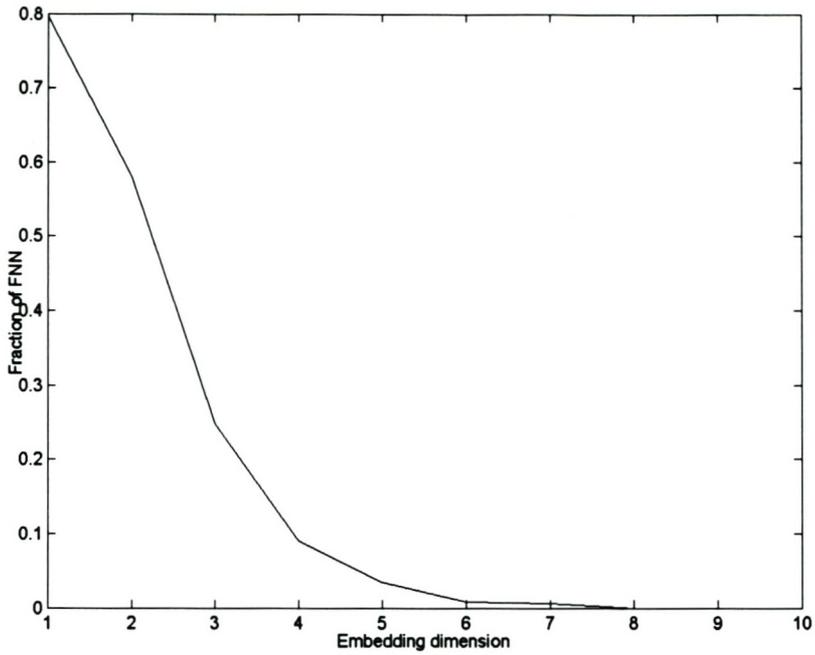
**Figure 11.25: FNN plot of FAG mill coarse feed rate ( $x_3$ ) with  $k = 240$**



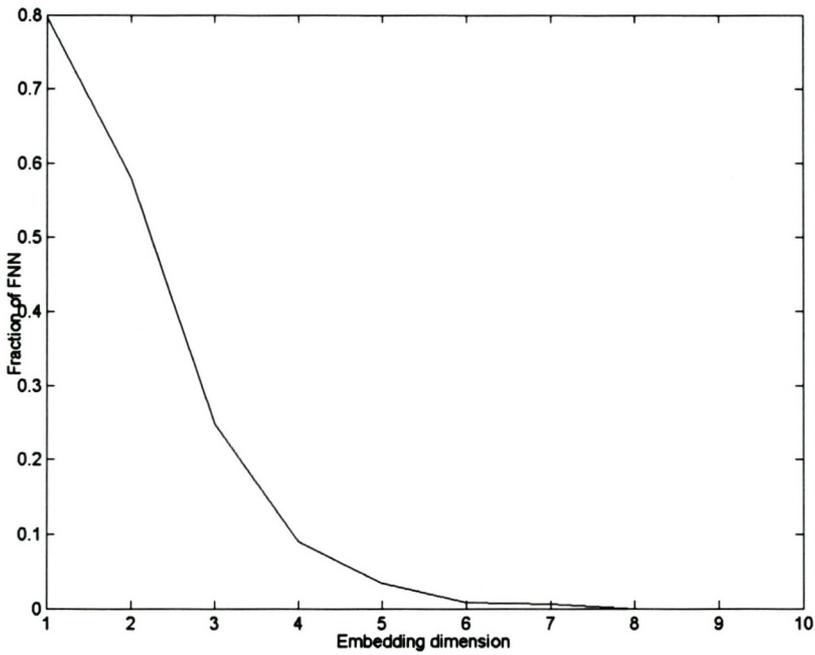
**Figure 11.26: FNN plot of FAG mill coarse feed rate ( $x_3$ ) with  $k = 100$**



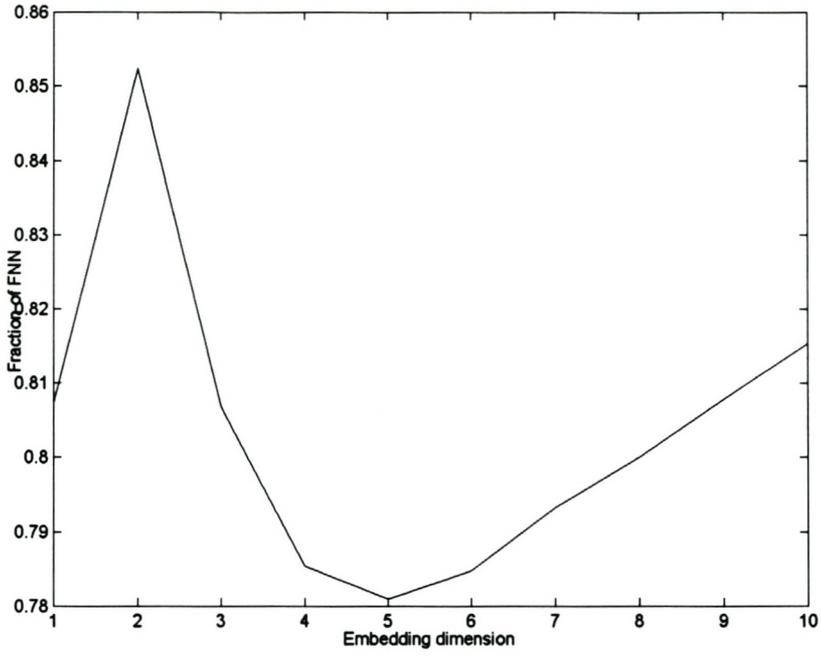
**Figure 11.27: FNN plot of FAG mill fine feed rate ( $x_4$ ) with  $k = 16$**



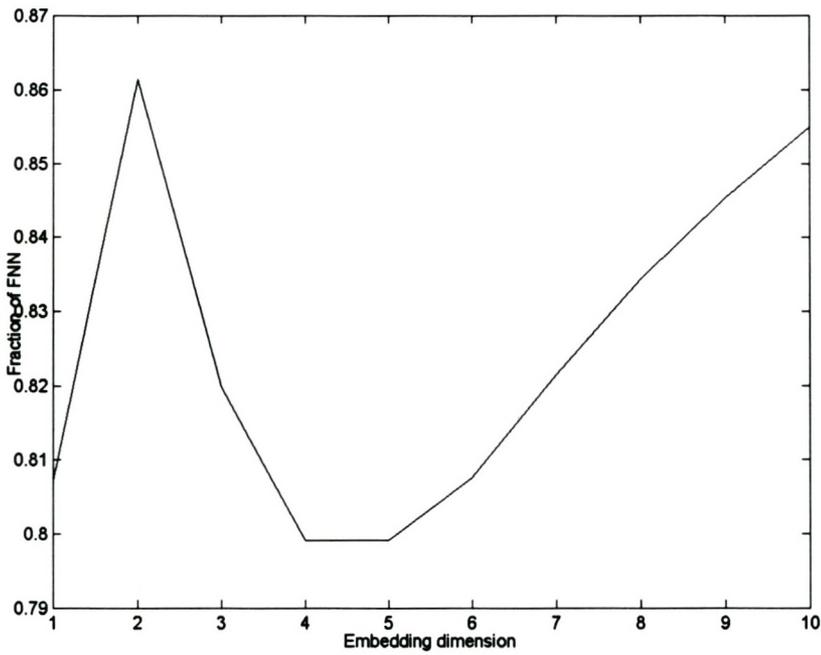
**Figure 11.28: FNN plot of FAG mill inlet water flow rate ( $x_7$ ) with  $k = 42$**



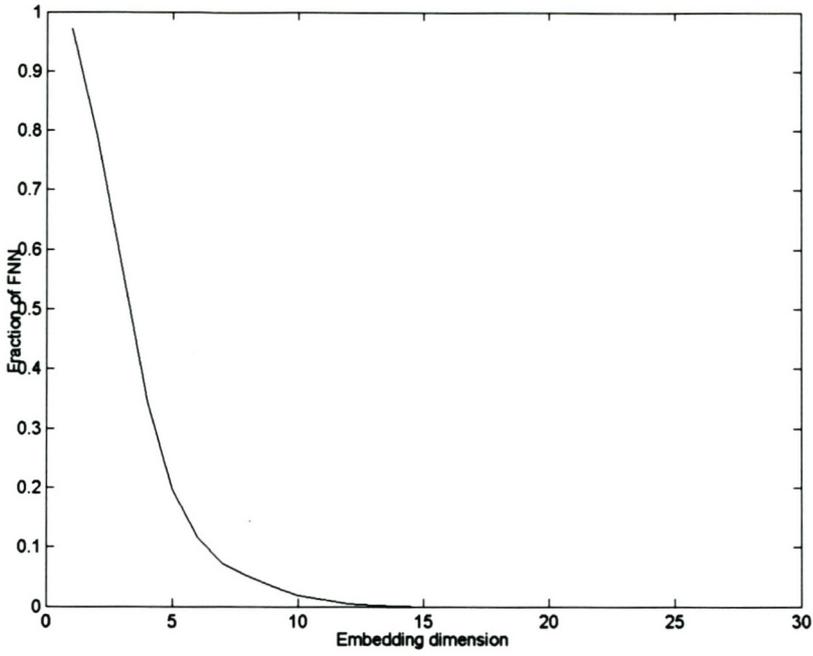
**Figure 11.29: FNN plot of FAG mill inlet water flow rate ( $x_7$ ) with  $k = 42$**



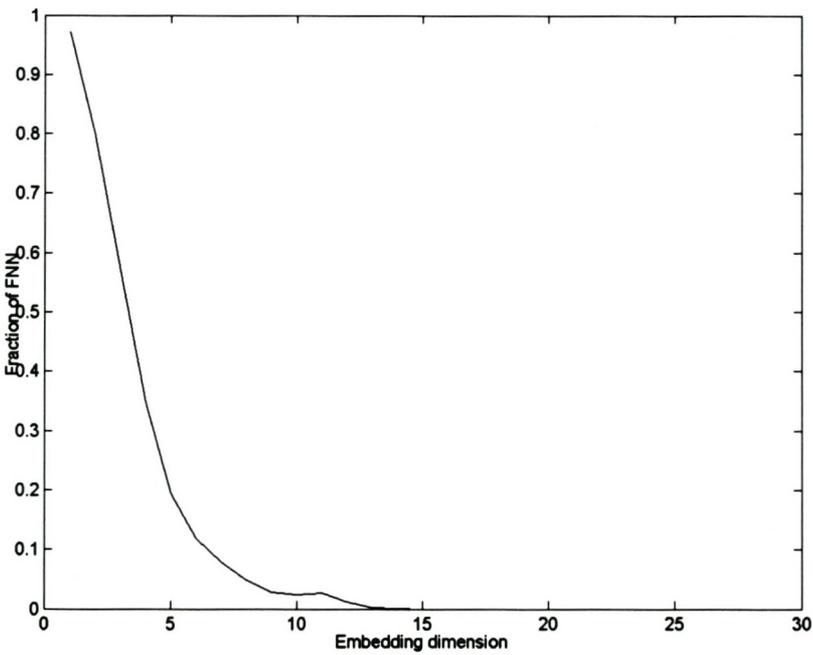
**Figure 11.30: FNN plot of FAG mill discharge 1 flow rate ( $x_9$ ) with  $k = 1$**



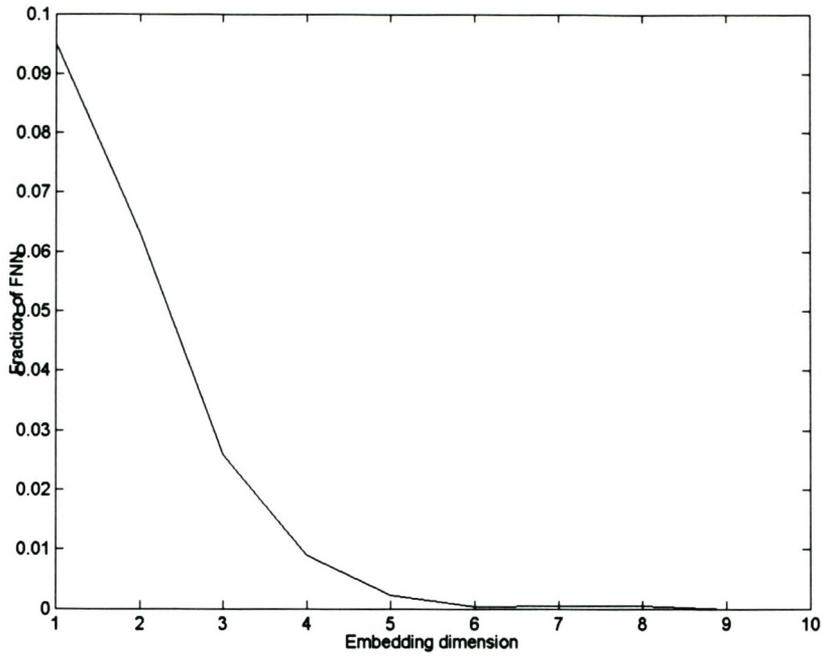
**Figure 11.31: FNN plot of FAG mill discharge 1 flow rate ( $x_9$ ) with  $k = 4$**



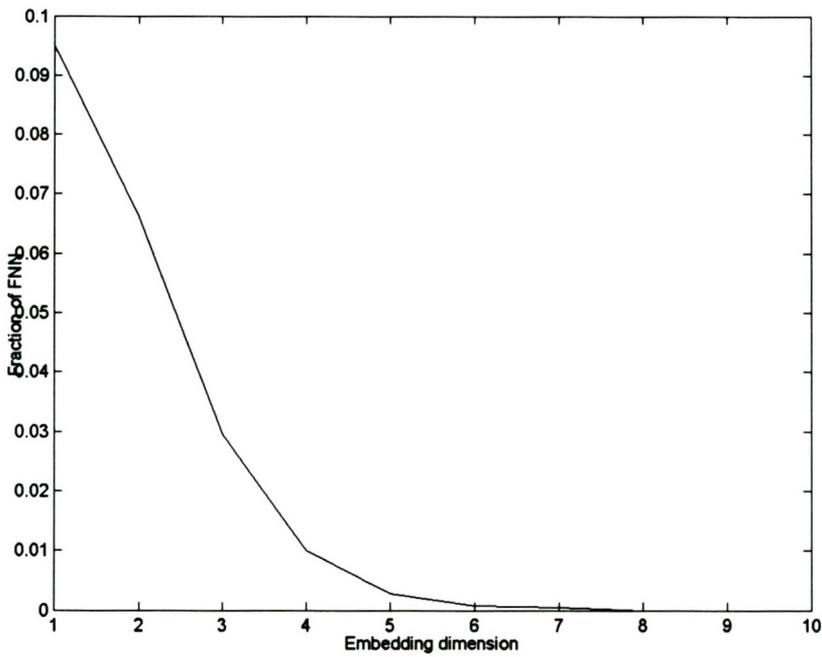
**Figure 11.32: FNN plot of FAG mill discharge 2 flow rate ( $x_{10}$ ) with  $k = 3$**



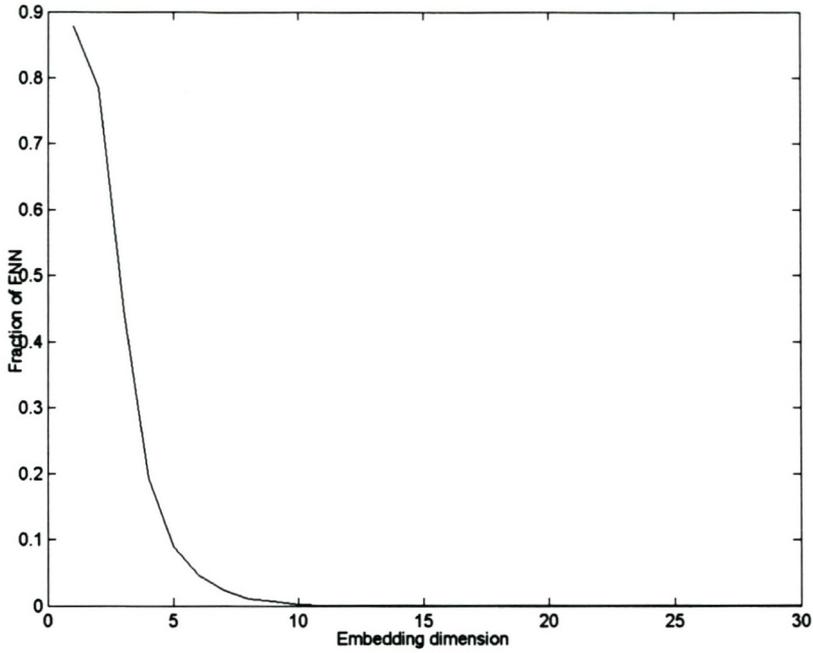
**Figure 11.33: FNN plot of FAG mill discharge 2 flow rate ( $x_{10}$ ) with  $k = 4$**



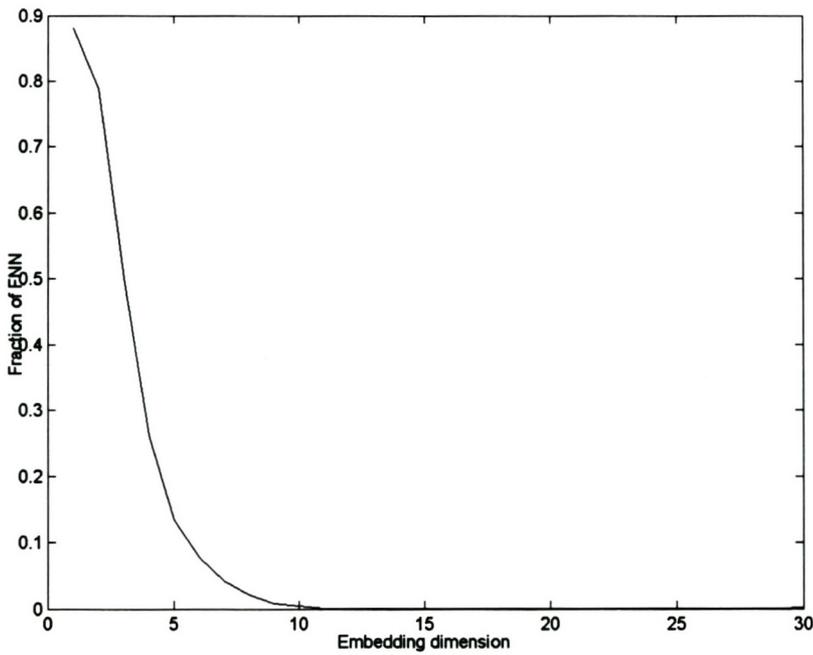
**Figure 11.34: FNN plot of FAG mill rougher feed 1 flow rate ( $x_{11}$ ) with  $k = 3$**



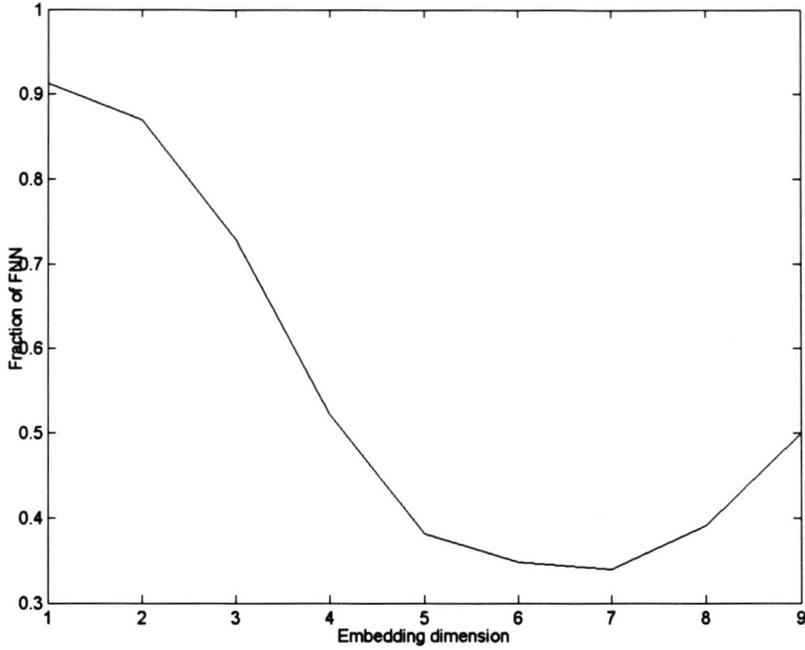
**Figure 11.35: FNN plot of FAG mill rougher feed 1 flow rate ( $x_{11}$ ) with  $k = 5$**



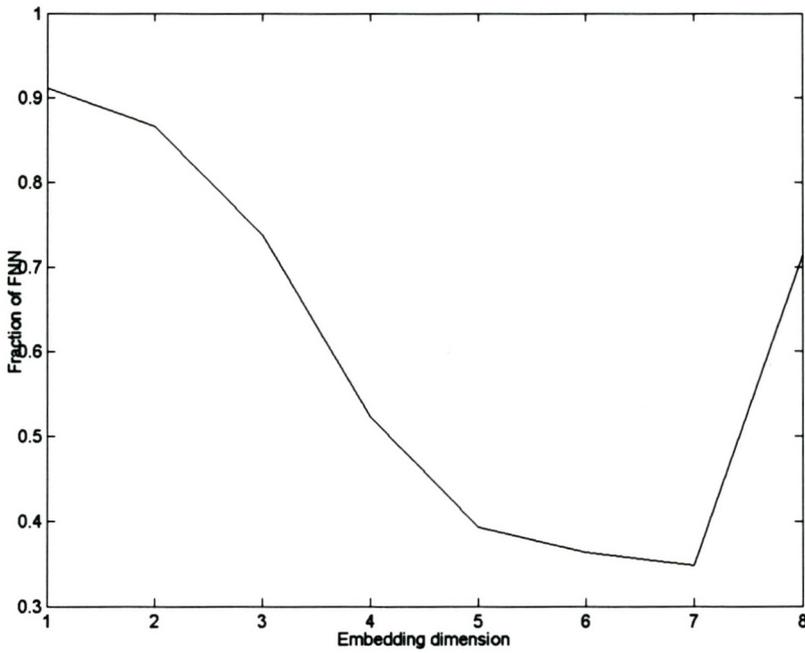
**Figure 11.36: FNN plot of FAG mill rougher feed 2 flow rate ( $x_{12}$ ) with  $k = 3$**



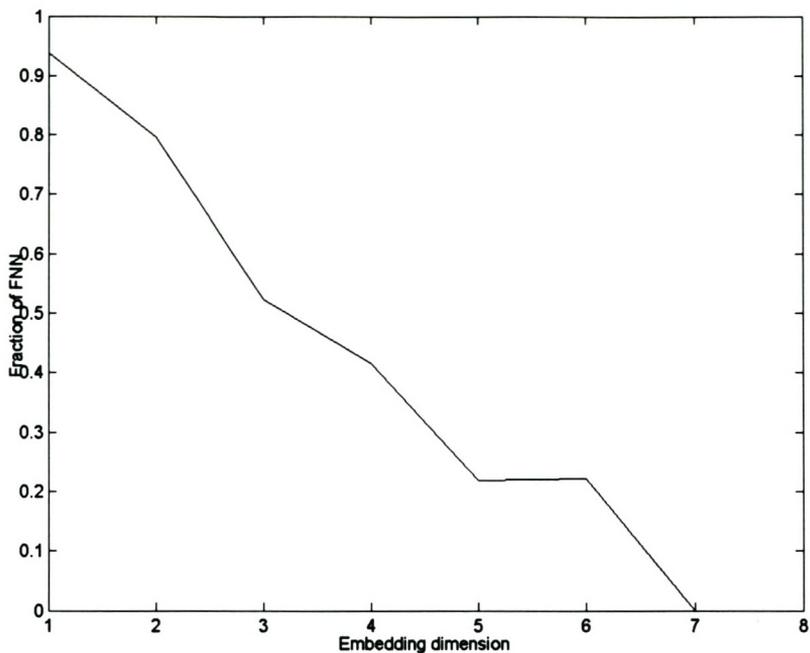
**Figure 11.37: FNN plot of FAG mill rougher feed 2 flow rate ( $x_{12}$ ) with  $k = 6$**



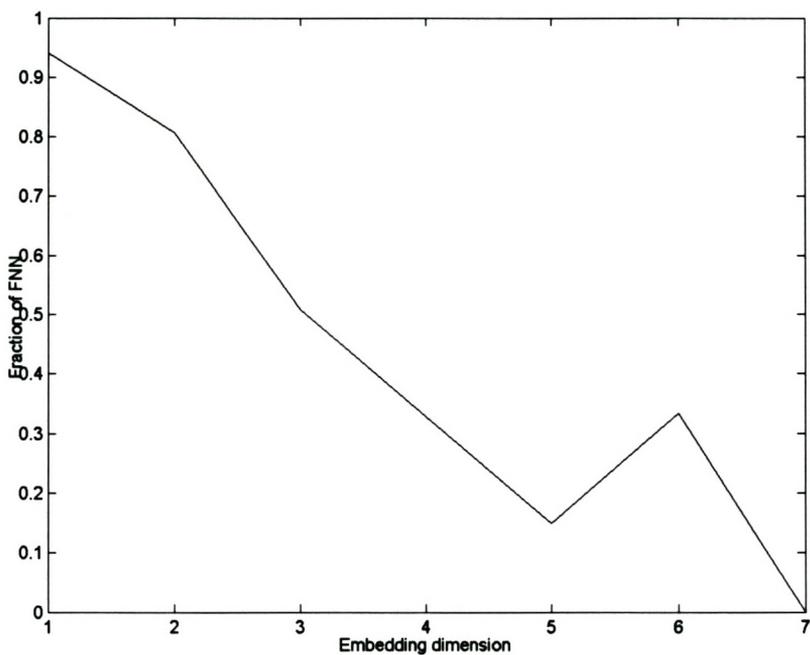
**Figure 11.38: FNN plot of FAG mill feed size distribution %  $-31.5\text{mm}$  ( $x_{13}$ ) with  $k = 22$**



**Figure 11.39: FNN plot of FAG mill feed size distribution %  $-31.5\text{mm}$  ( $x_{13}$ ) with  $k = 32$**

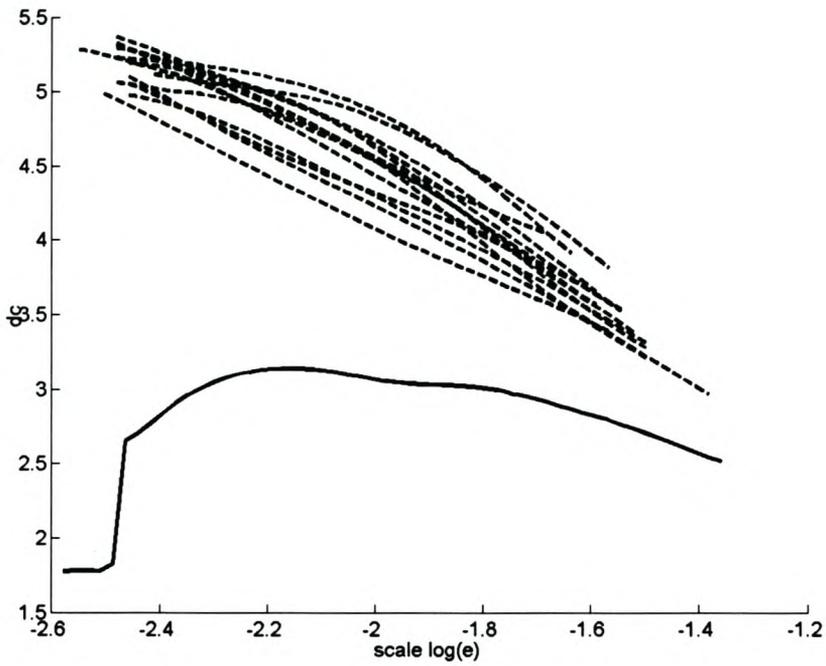


**Figure 11.40: FNN plot of FAG mill product size distribution % -75 $\mu\text{m}$  ( $x_{14}$ ) with  $k = 1$**

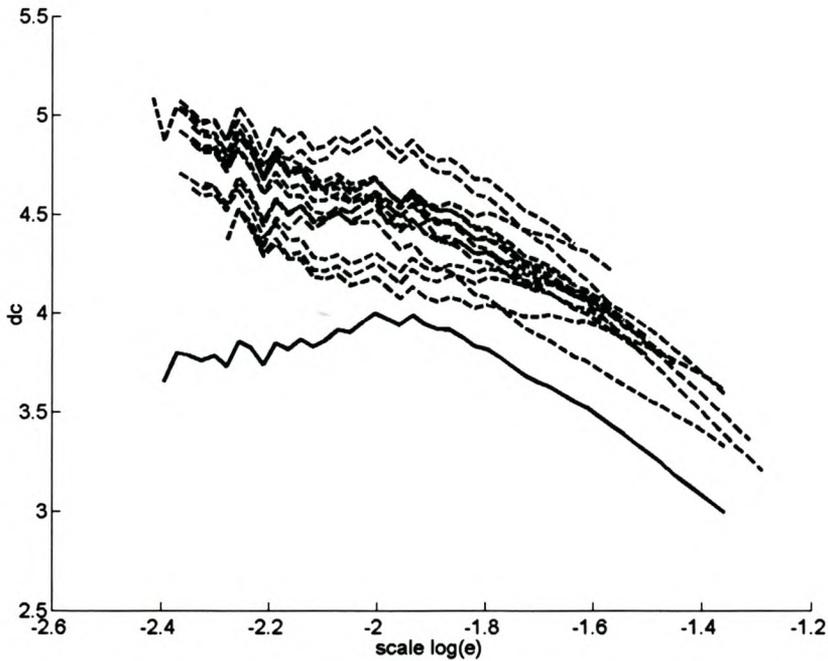


**Figure 11.41: FNN plot of FAG mill product size distribution % -75 $\mu\text{m}$  ( $x_{14}$ ) with  $k = 2$**

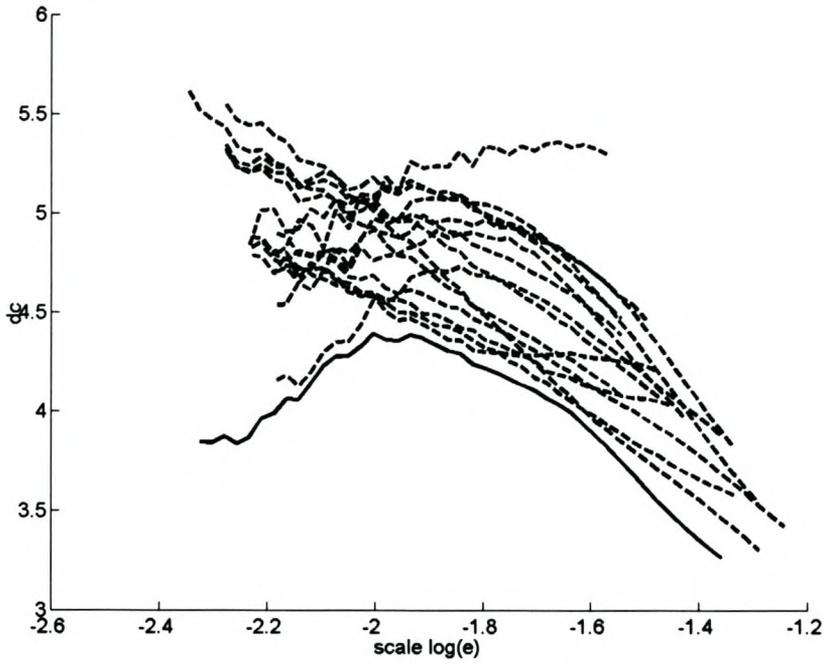
### 11.3.5 Correlation Dimension plots



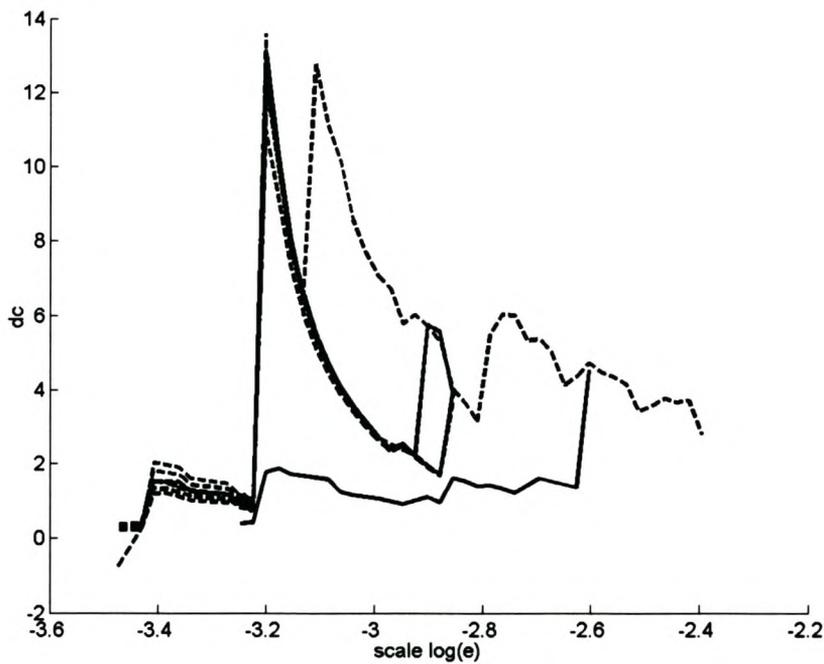
**Figure 11.42: Correlation dimension curve for FAG mill power ( $x_1$ ; solid line) and its corresponding surrogate data sets (broken lines) with  $k = 100$  and  $m = 6$**



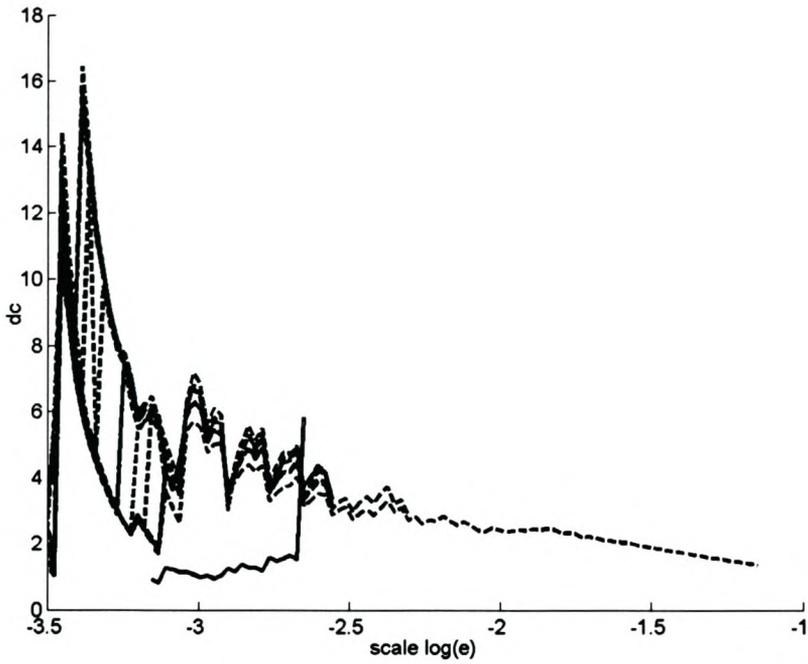
**Figure 11.43: Correlation dimension curve for FAG mill load ( $x_2$ ; solid line) and its corresponding surrogate data sets (broken lines) with  $k = 106$  and  $m = 6$**



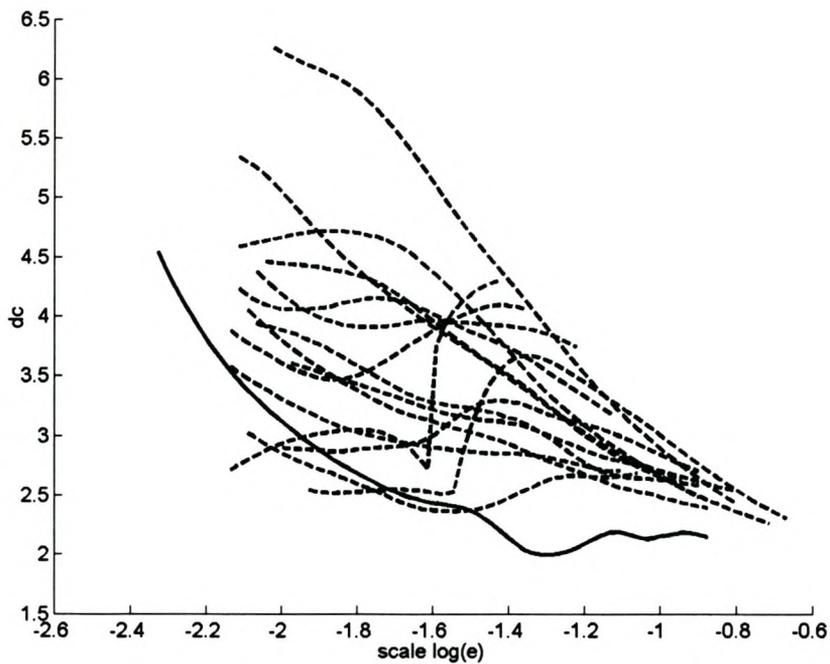
**Figure 11.44: Correlation dimension curve for FAG mill load ( $x_2$ ; solid line) and its corresponding surrogate data sets (broken lines) with  $k = 100$  and  $m = 8$**



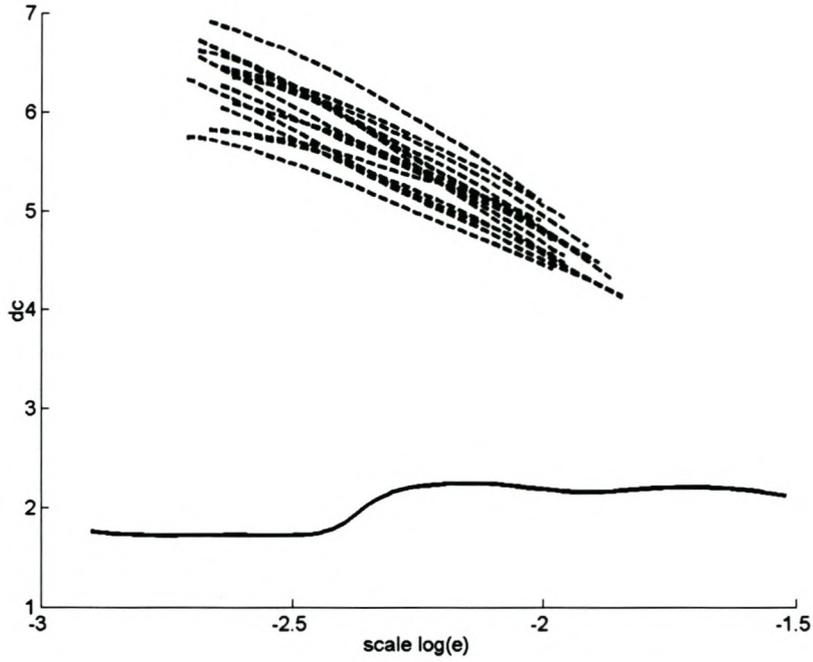
**Figure 11.45: Correlation dimension curve for FAG mill coarse feed rate ( $x_3$ ; solid line) and its corresponding surrogate data sets (broken lines) with  $k = 240$  and  $m = 2$**



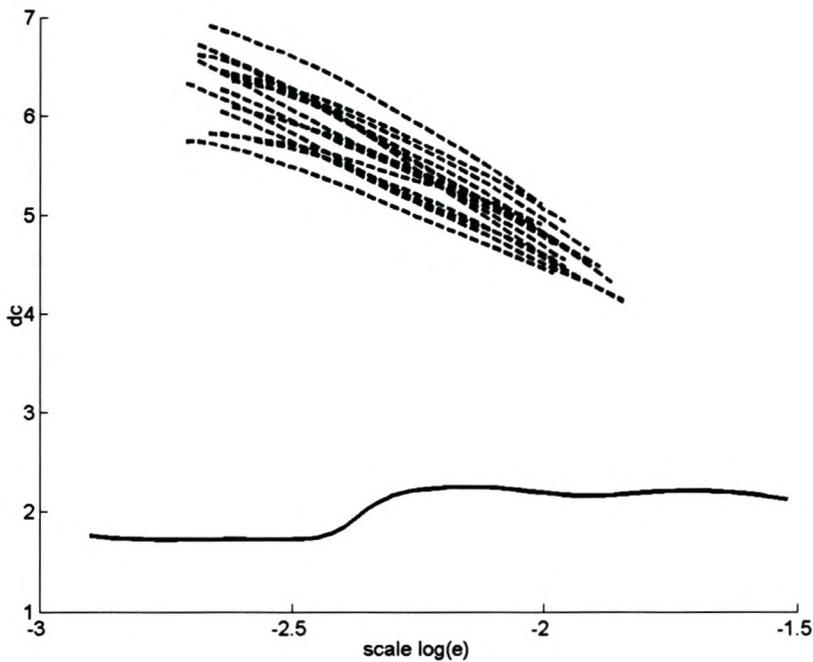
**Figure 11.46: Correlation dimension curve for FAG mill coarse feed rate ( $x_3$ ; solid line) and its corresponding surrogate data sets (broken lines) with  $k = 100$  and  $m = 5$**



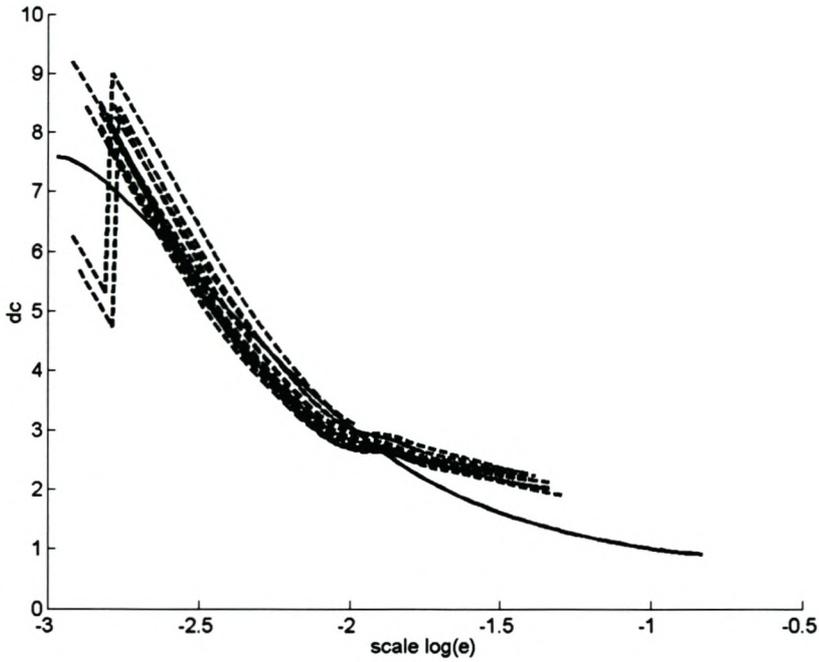
**Figure 11.47: Correlation dimension curve for FAG mill fine feed rate ( $x_4$ ; solid line) and its corresponding surrogate data sets (broken lines) with  $k = 16$  and  $m = 18$**



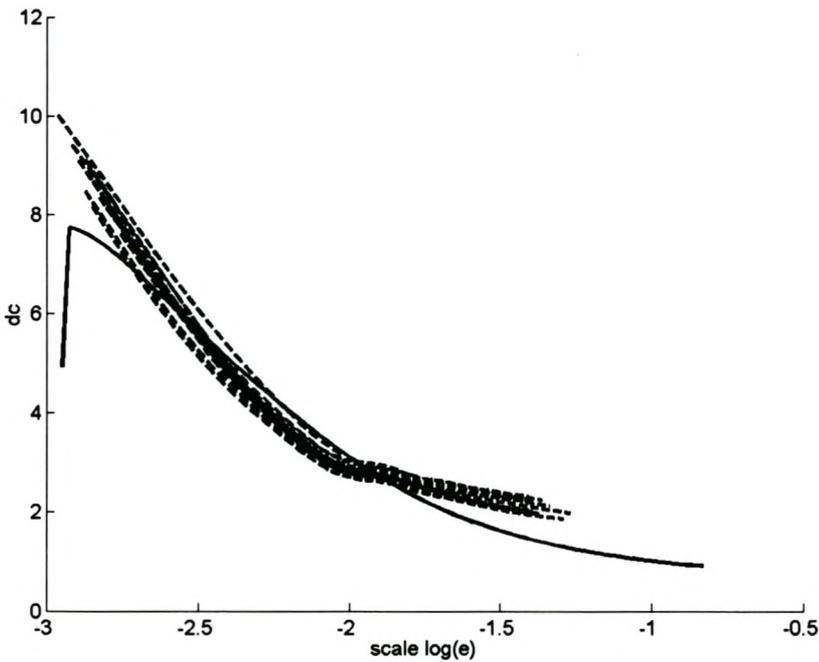
**Figure 11.48: Correlation dimension curve for FAG mill inlet water flow rate ( $x_7$ ; solid line) and its corresponding surrogate data sets (broken lines) with  $k = 42$  and  $m = 8$**



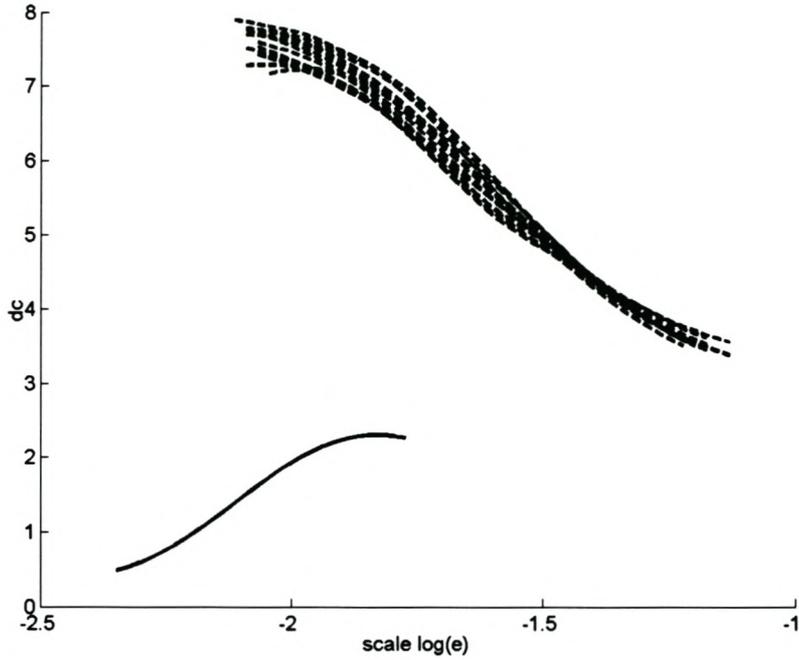
**Figure 11.49: Correlation dimension curve for FAG mill inlet water flow rate ( $x_7$ ; solid line) and its corresponding surrogate data sets (broken lines) with  $k = 42$  and  $m = 8$**



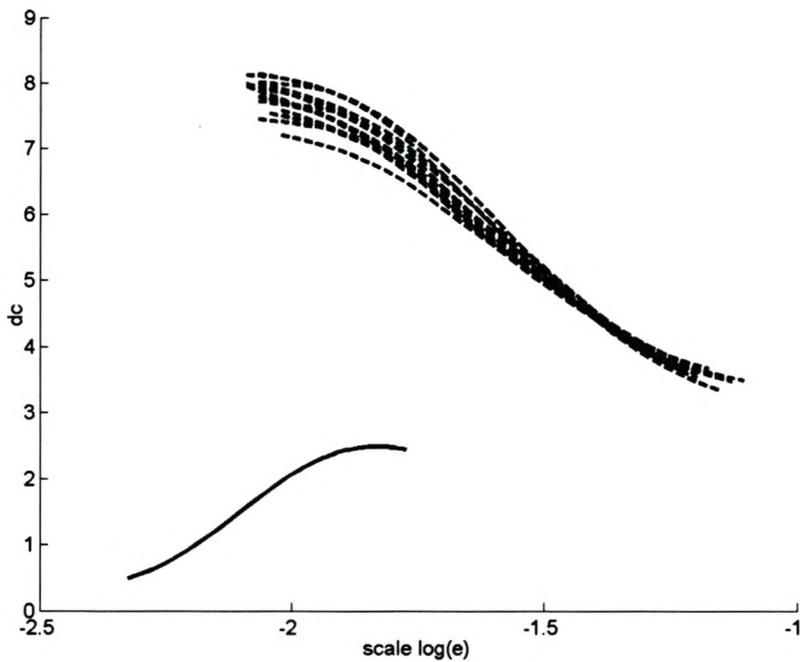
**Figure 11.50: Correlation dimension curve for FAG mill discharge 2 flow rate ( $x_{10}$ ; solid line) and its corresponding surrogate data sets (broken lines) with  $k = 3$  and  $m = 15$**



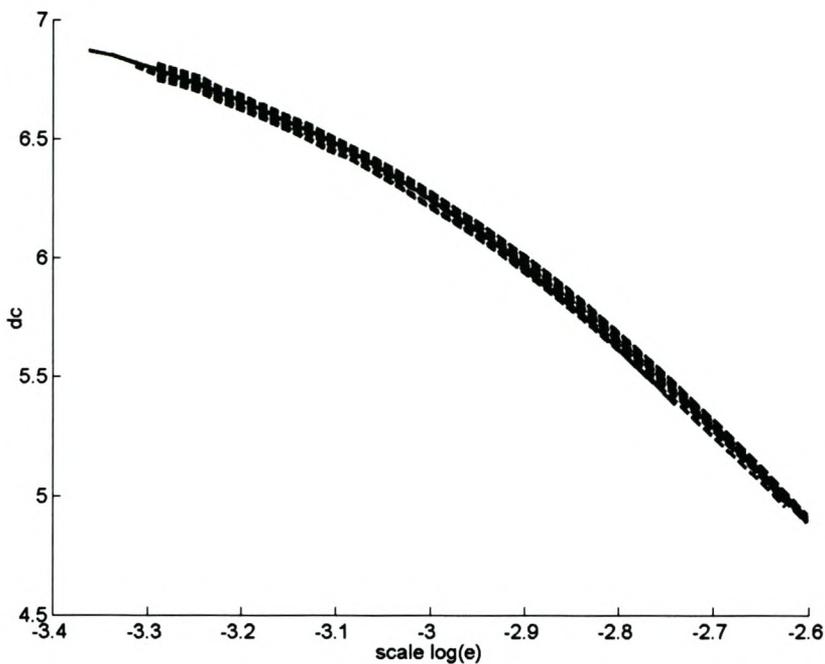
**Figure 11.51: Correlation dimension curve for FAG mill discharge 2 flow rate ( $x_{10}$ ; solid line) and its corresponding surrogate data sets (broken lines) with  $k = 4$  and  $m = 15$**



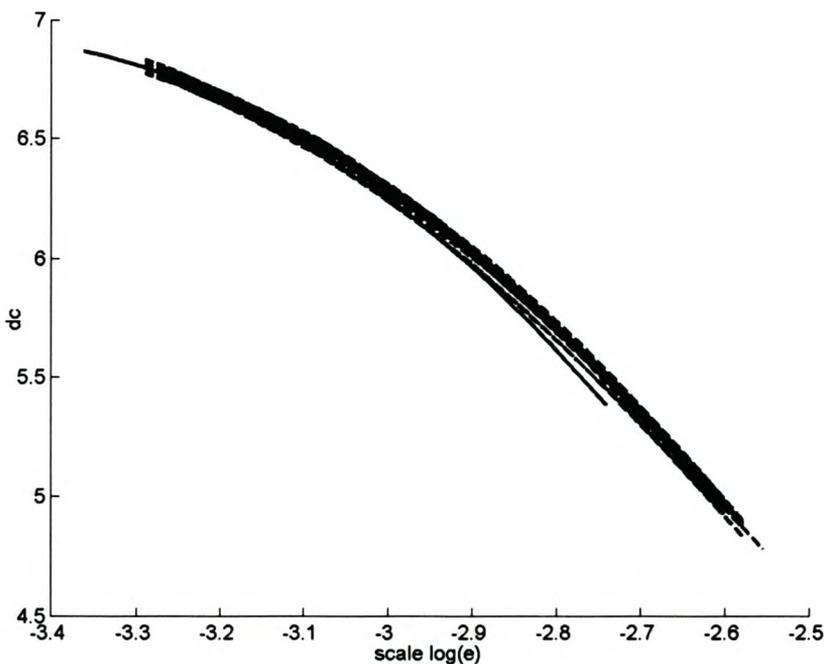
**Figure 11.52: Correlation dimension curve for FAG mill rougher feed 2 flow rate ( $x_{12}$ ; solid line) and its corresponding surrogate data sets (broken lines) with  $k = 3$  and  $m = 11$**



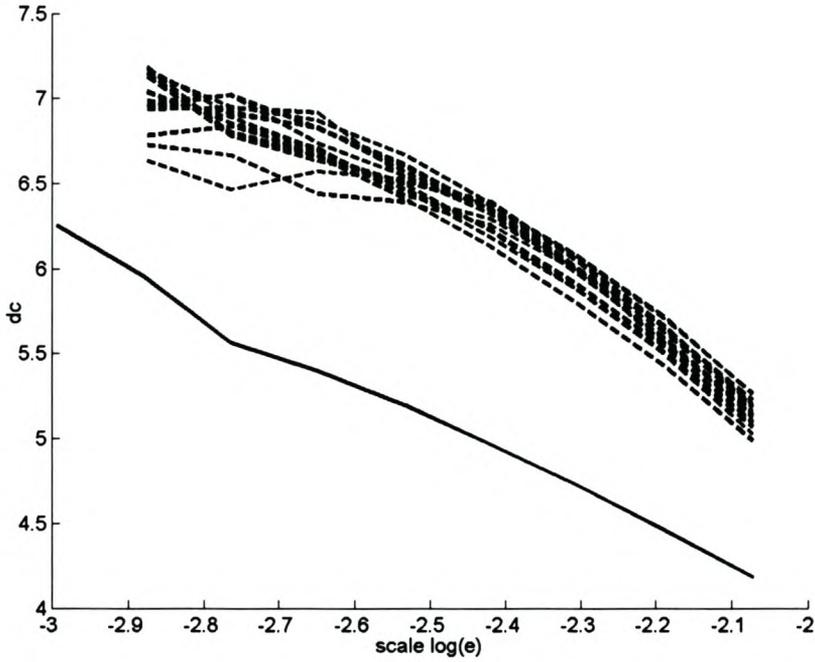
**Figure 11.53: Correlation dimension curve for FAG mill rougher feed 2 flow rate ( $x_{12}$ ; solid line) and its corresponding surrogate data sets (broken lines) with  $k = 6$  and  $m = 11$**



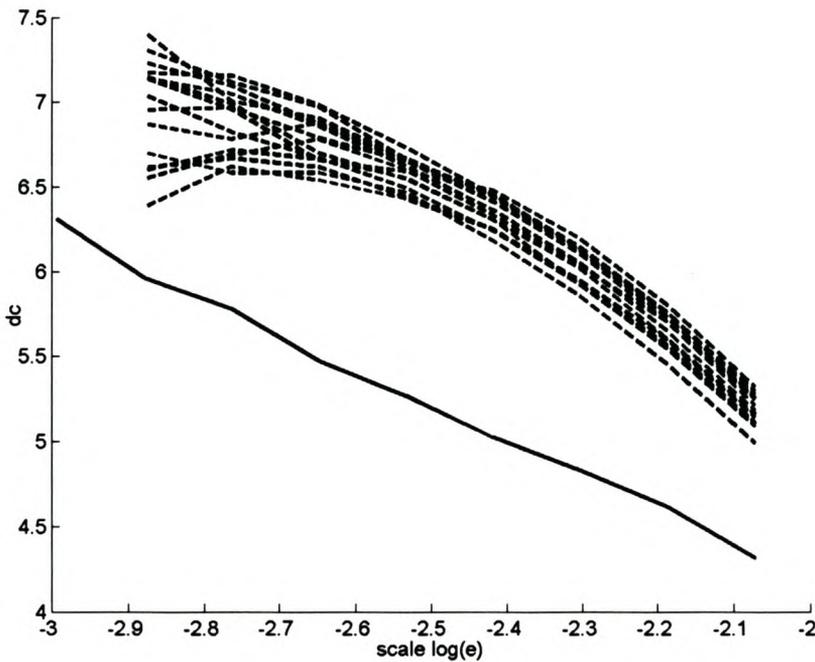
**Figure 11.54: Correlation dimension curve for FAG mill feed size distribution % -31.5mm ( $x_{13}$ ; solid line) and its corresponding surrogate data sets (broken lines) with  $k = 22$  and  $m = 7$**



**Figure 11.55: Correlation dimension curve for FAG mill feed size distribution % -31.5mm ( $x_{13}$ ; solid line) and its corresponding surrogate data sets (broken lines) with  $k = 32$  and  $m = 7$**

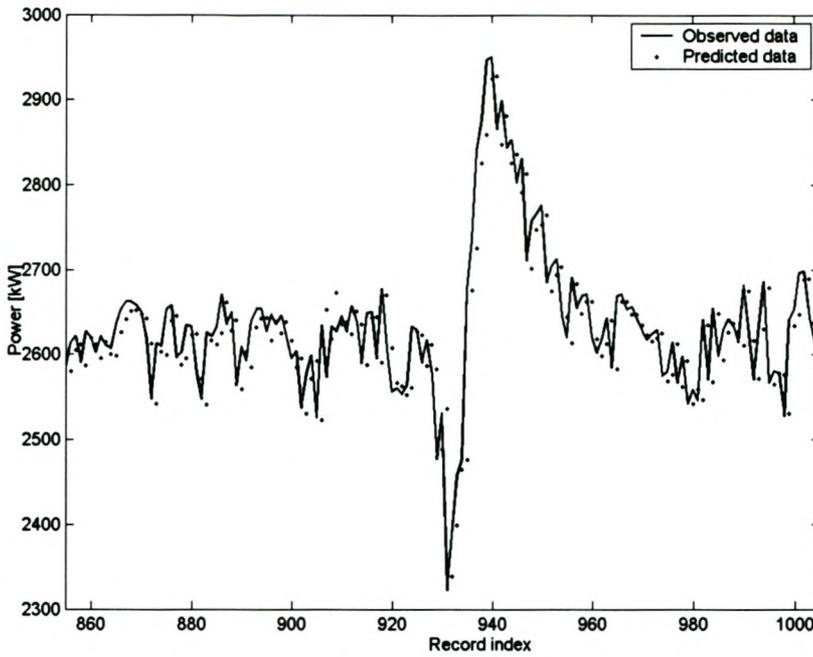


**Figure 11.56: Correlation dimension curve for FAG mill product size distribution % -75 $\mu\text{m}$  ( $x_{14}$ ; solid line) and its corresponding surrogate data sets (broken lines) with  $k = 1$  and  $m = 7$**

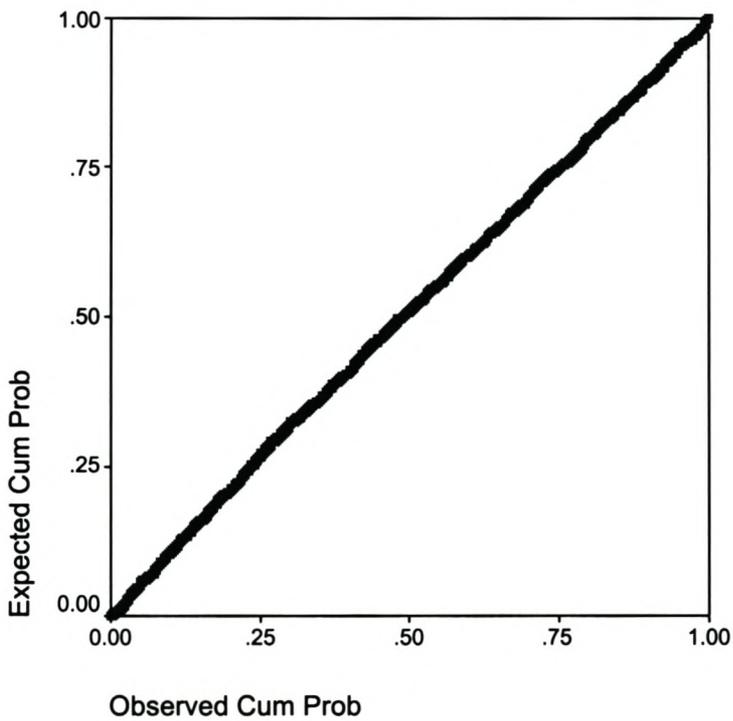


**Figure 11.57: Correlation dimension curve for FAG mill product size distribution % -75 $\mu\text{m}$  ( $x_{14}$ ; solid line) and its corresponding surrogate data sets (broken lines) with  $k = 2$  and  $m = 7$**

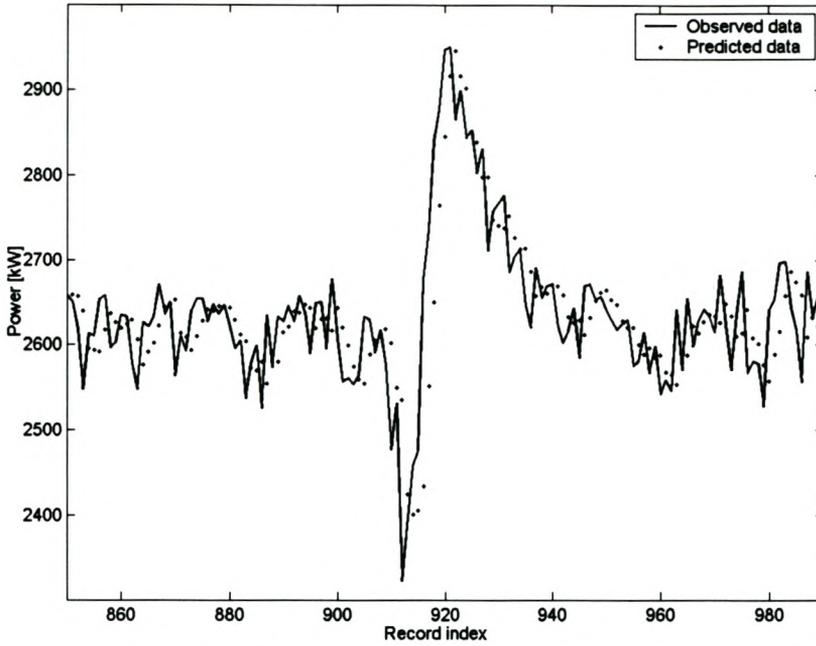
### 11.3.6 LS-SVM modelling plots



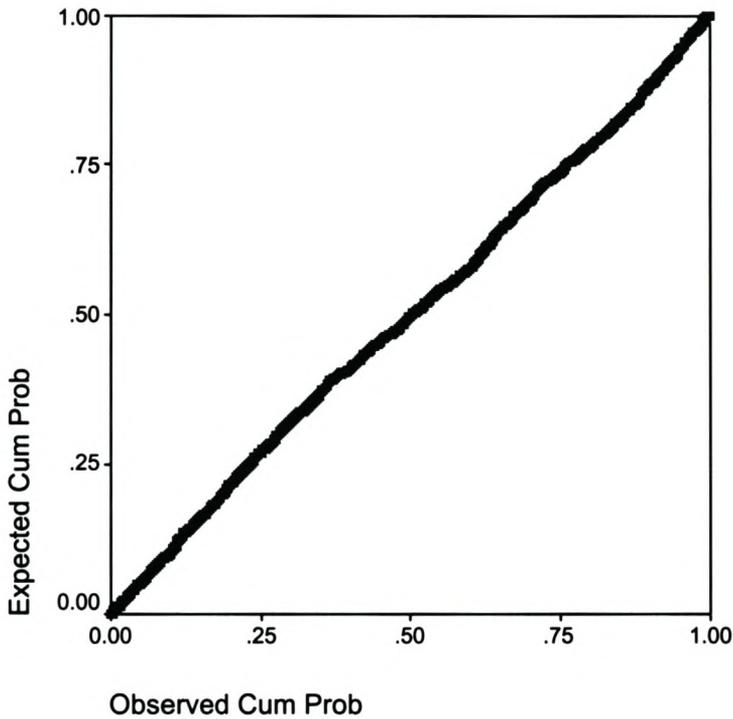
**Figure 11.58: One-step prediction of FAG mill power ( $x_1$ ) using *SVMnone* ( $R^2 = 84.17\%$ )**



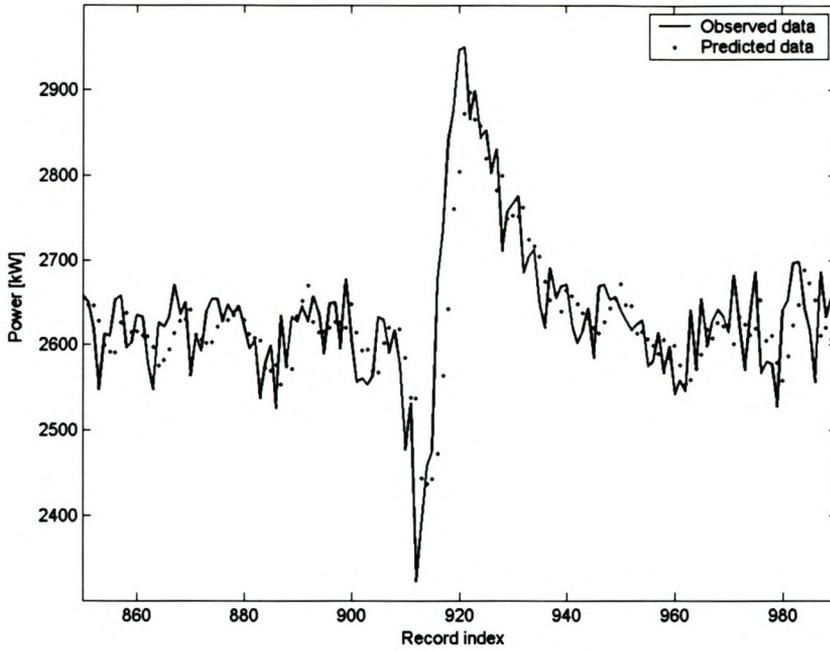
**Figure 11.59: Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using *SVMnone***



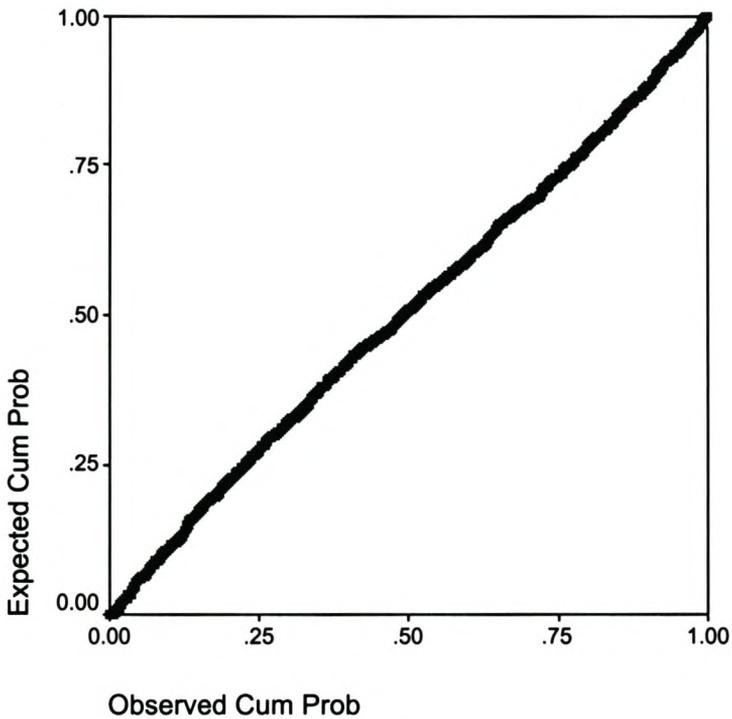
**Figure 11.60: One-step prediction of FAG mill power ( $x_1$ ) using  $SVM_0$  ( $R^2 = 85.46\%$ )**



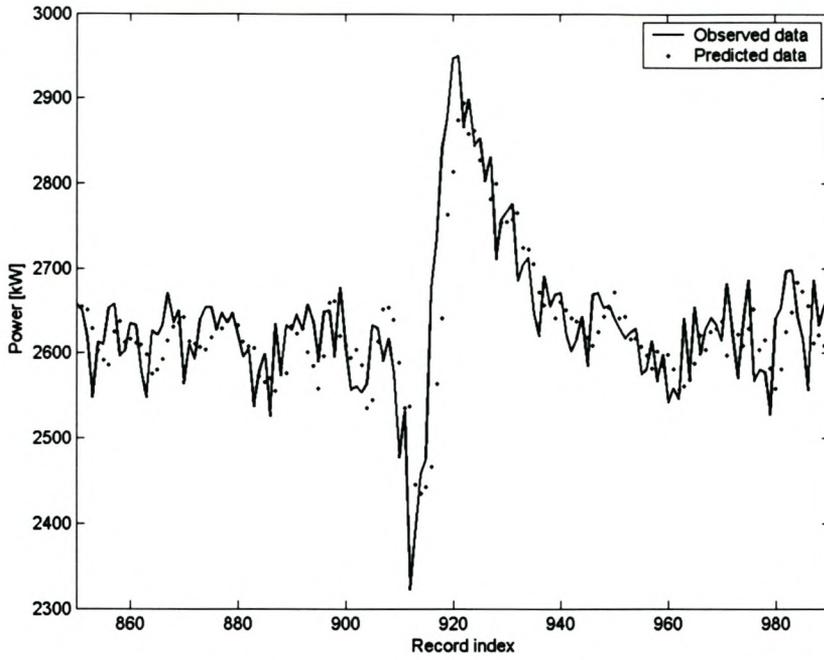
**Figure 11.61: Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using  $SVM_0$**



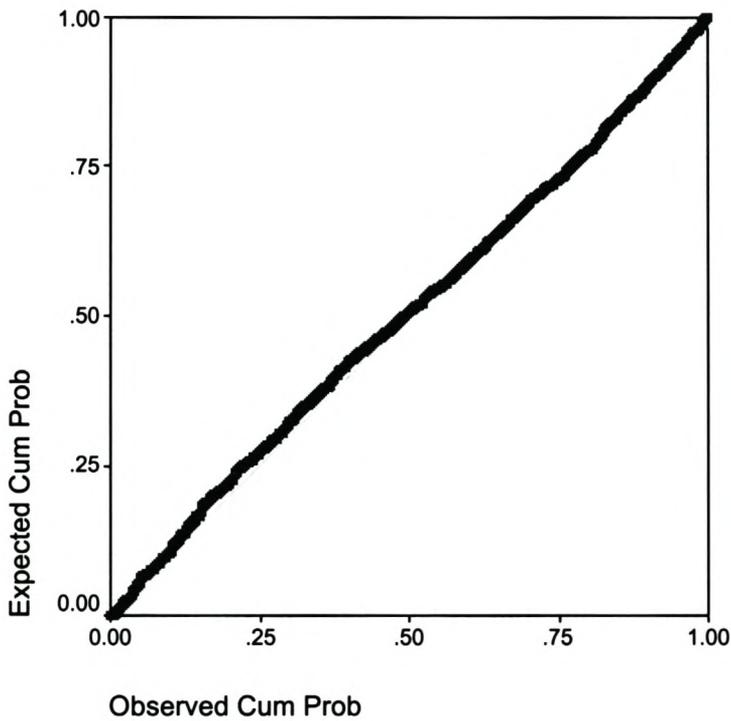
**Figure 11.62: One-step prediction of FAG mill power ( $x_1$ ) using  $SVM_1$  ( $R^2 = 85.82\%$ )**



**Figure 11.63: Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using  $SVM_1$**

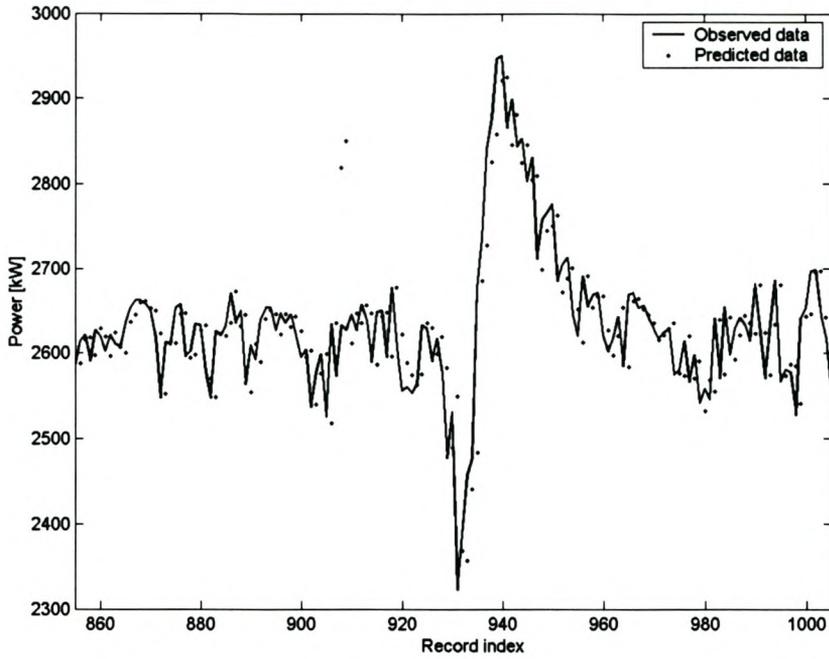


**Figure 11.64: One-step prediction of FAG mill power ( $x_1$ ) using  $SVM_4$  ( $R^2 = 85.21\%$ )**

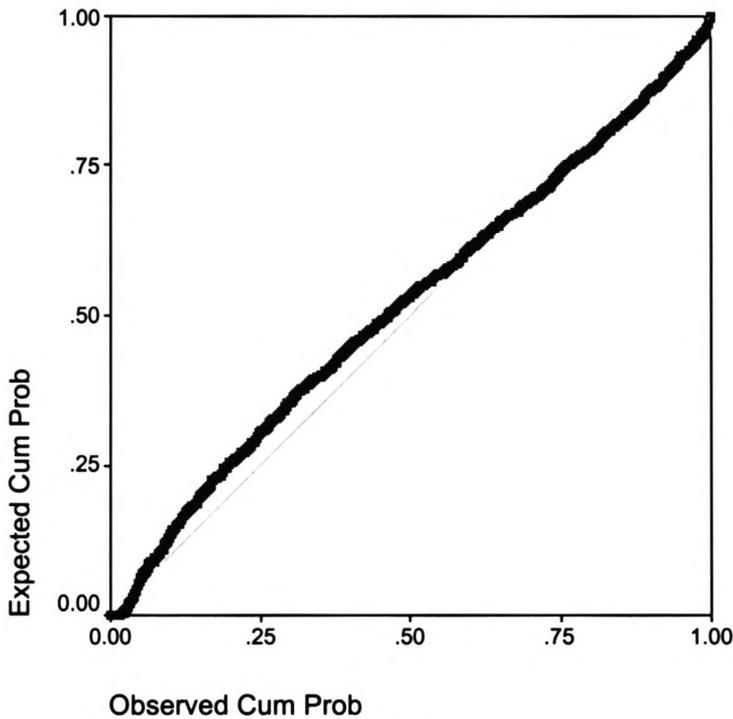


**Figure 11.65: Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using  $SVM_4$**

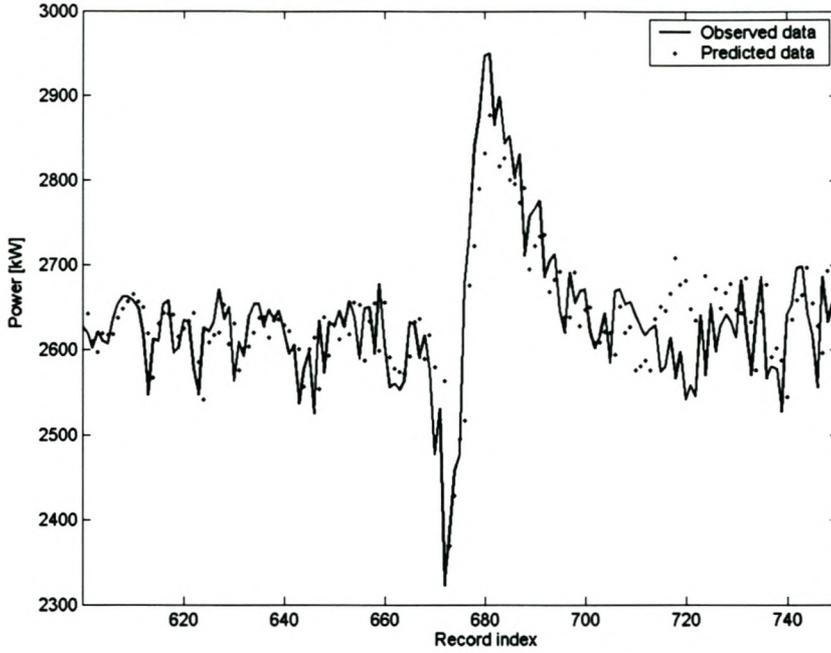
**11.3.7 RBF neural network modelling plots**



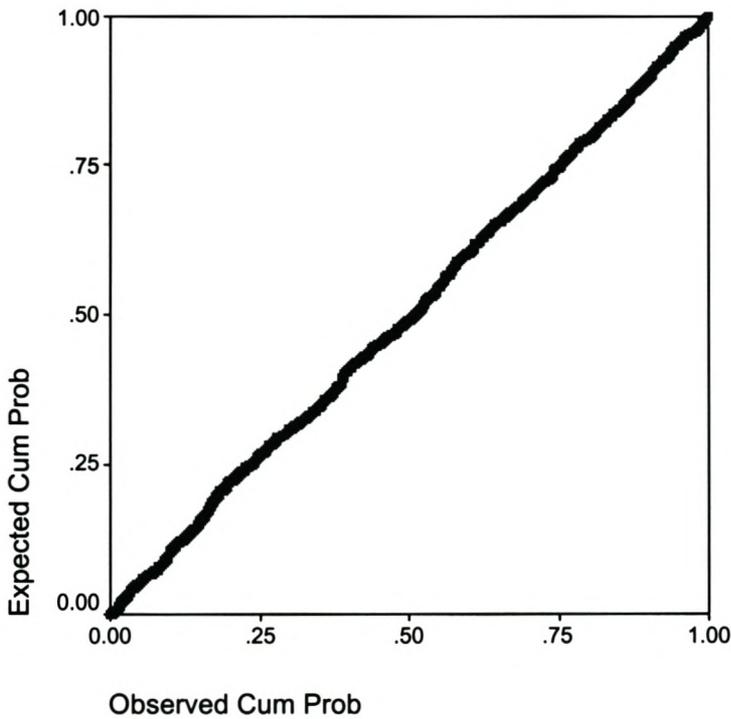
**Figure 11.66: One-step prediction of FAG mill power ( $x_1$ ) using *RBFnone* ( $R^2 = 77.99\%$ )**



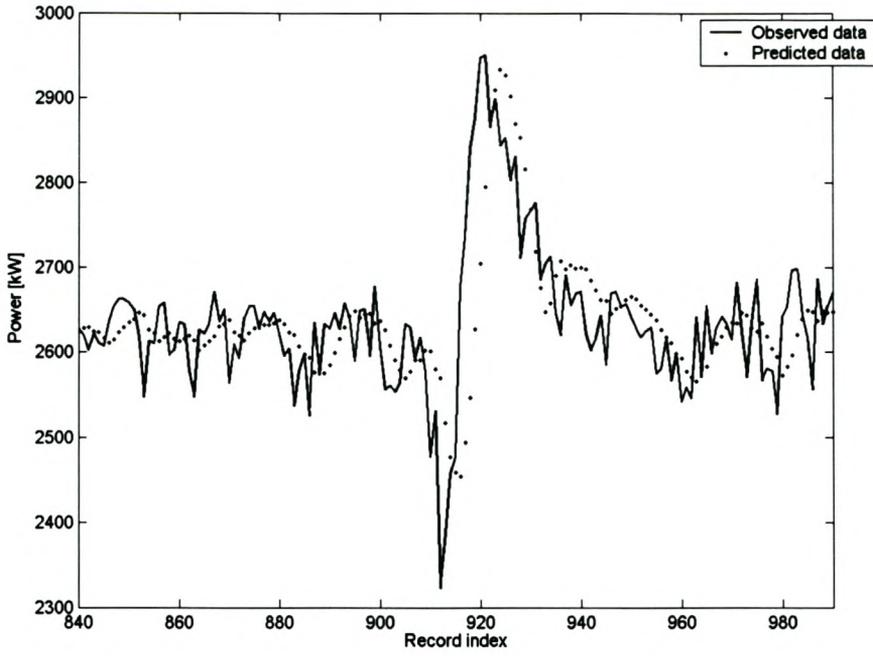
**Figure 11.67: Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using *RBFnone***



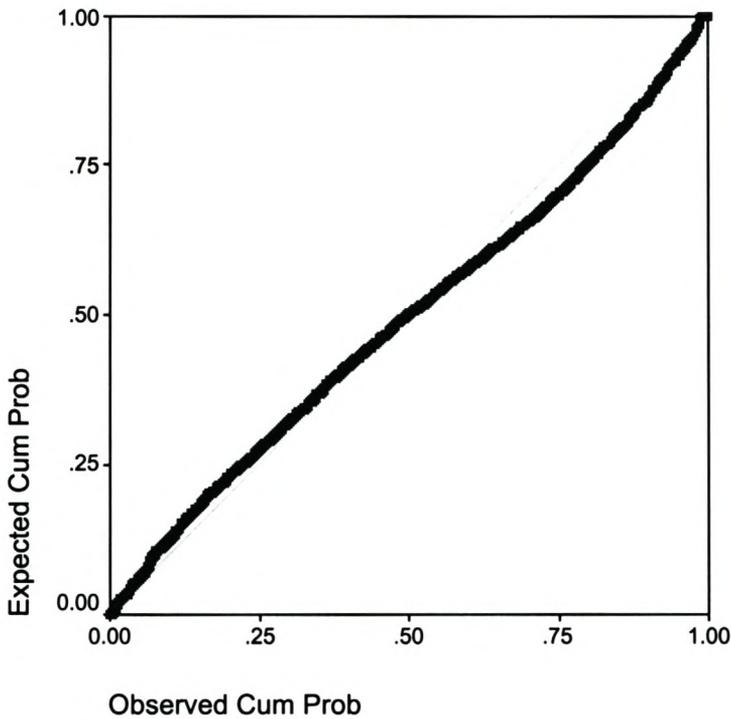
**Figure 11.68: One-step prediction of FAG mill power ( $x_1$ ) using *RBFMOD\_0* ( $R^2 = 46.37\%$ )**



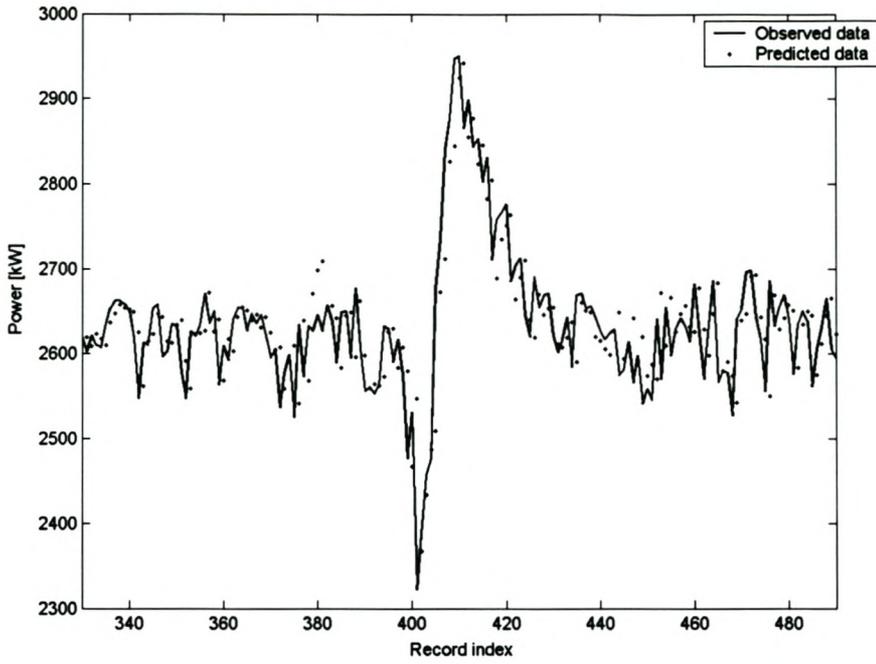
**Figure 11.69: Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using *RBFMOD\_0***



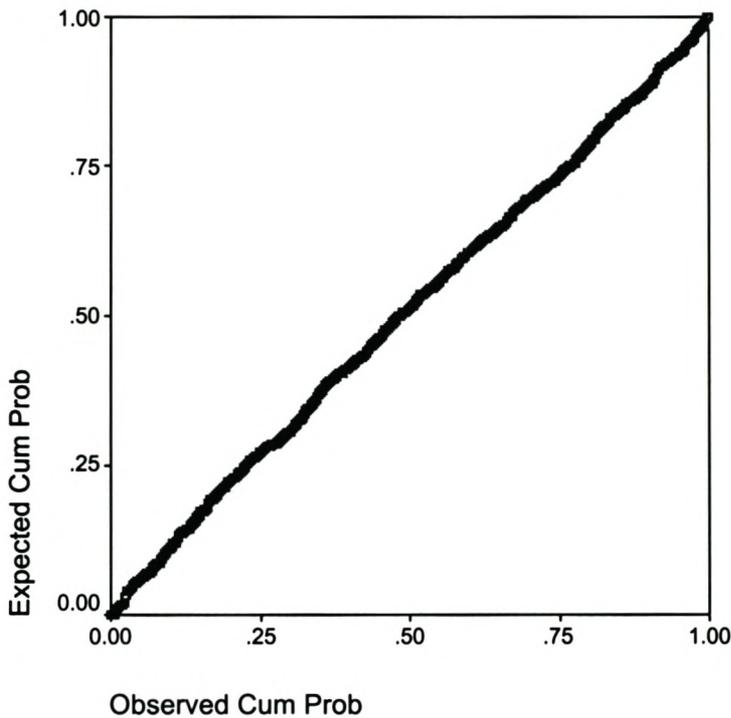
**Figure 11.70: One-step prediction of FAG mill power ( $x_1$ ) using *RFSSA\_0* ( $R^2 = 82.41\%$ )**



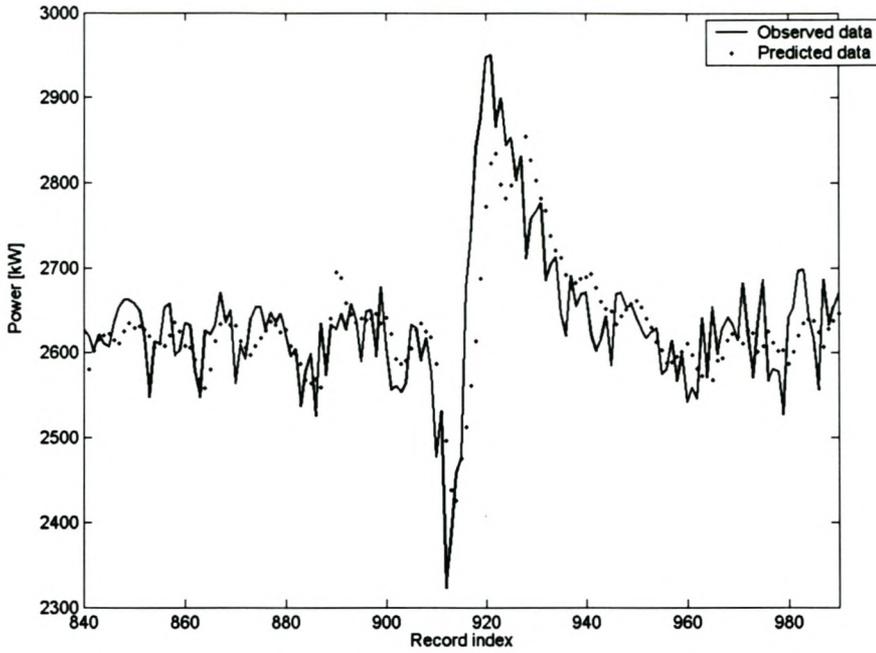
**Figure 11.71: Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using *RFSSA\_0***



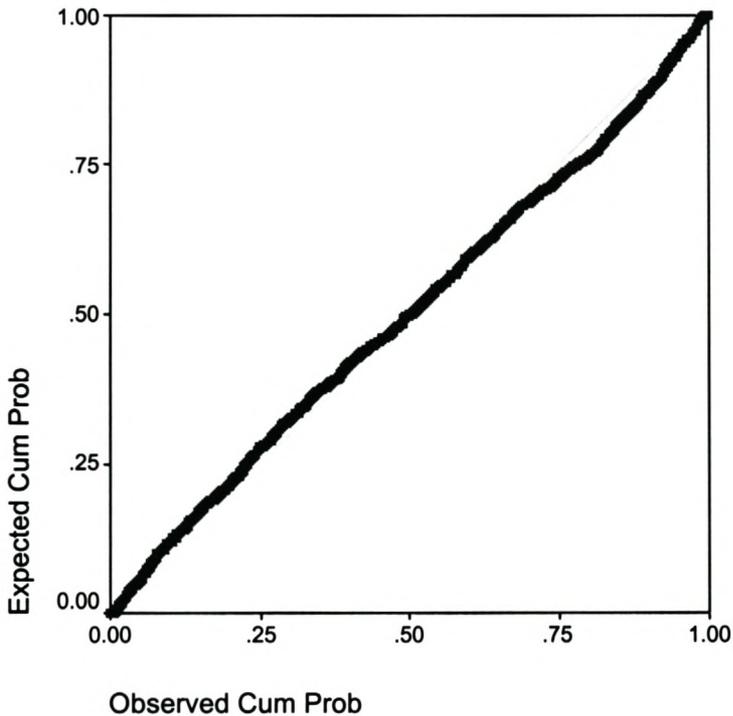
**Figure 11.72: One-step prediction of FAG mill power ( $x_1$ ) using *RBFMOD\_1* ( $R^2 = 39.91\%$ )**



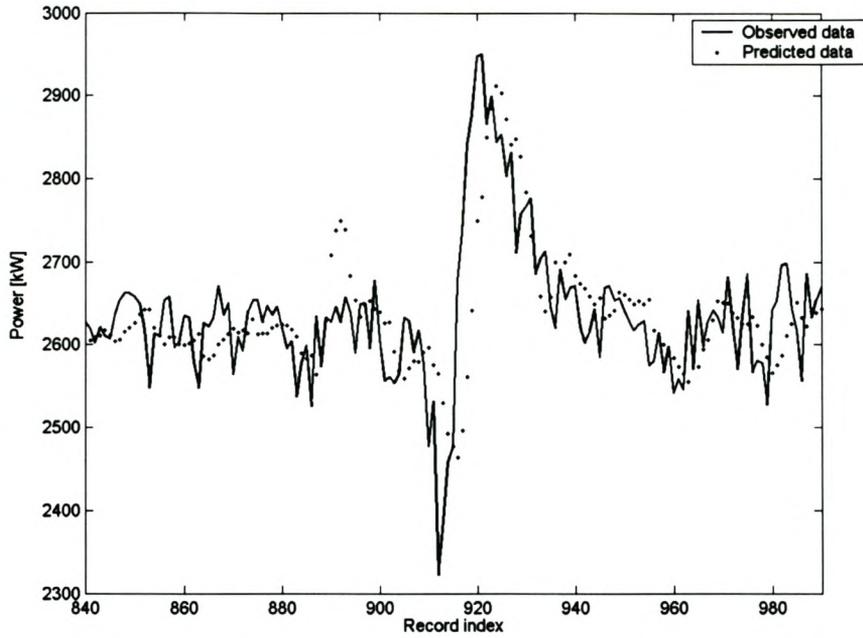
**Figure 11.73: Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using *RBFMOD\_1***



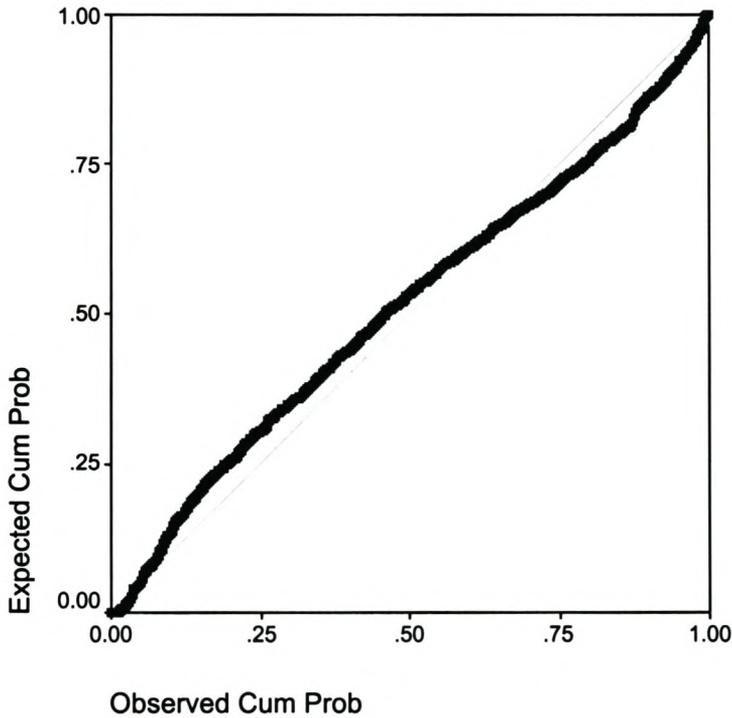
**Figure 11.74: One-step prediction of FAG mill power ( $x_1$ ) using *RFSSA\_1* ( $R^2 = 82.69\%$ )**



**Figure 11.75: Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using *RFSSA\_1***



**Figure 11.76: One-step prediction of FAG mill power ( $x_1$ ) using *RFSSA\_2* ( $R^2 = 78.44\%$ )**



**Figure 11.77: Normal P-P plot of the residual values for one-step prediction of the FAG mill power ( $x_1$ ) using *RFSSA\_2***

### 11.3.8 Free-run prediction results obtained from multi-dimensional time series models

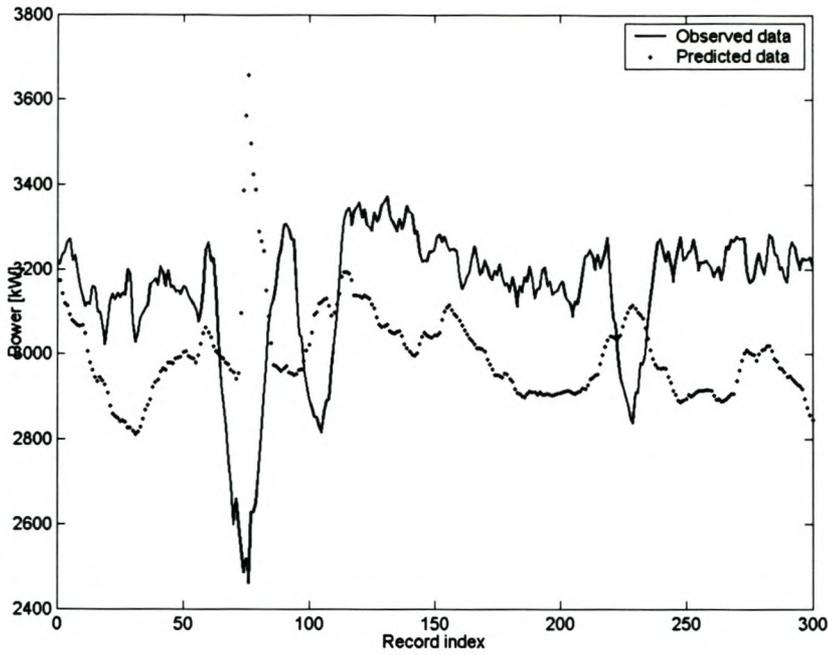


Figure 11.78: Free-run prediction using *MLPMOD\_1*

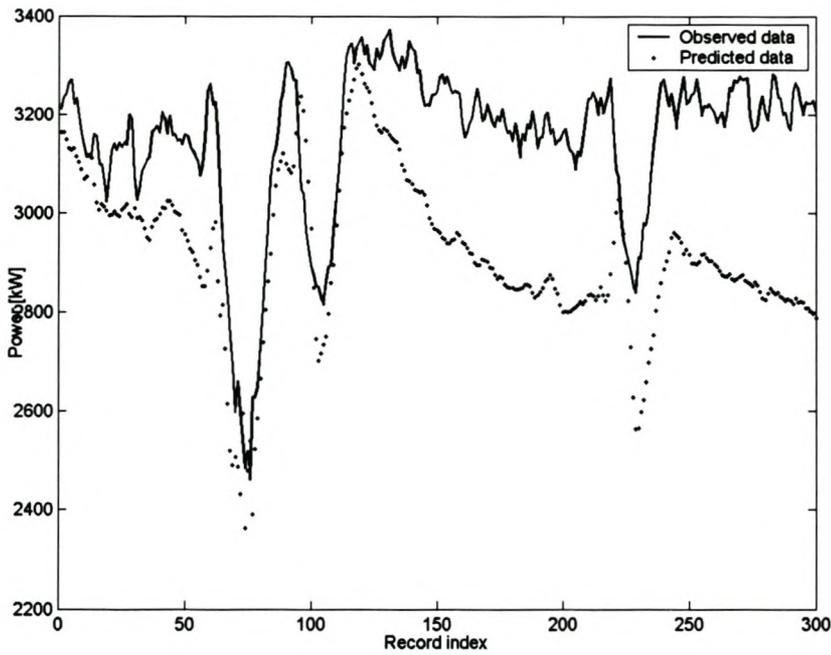
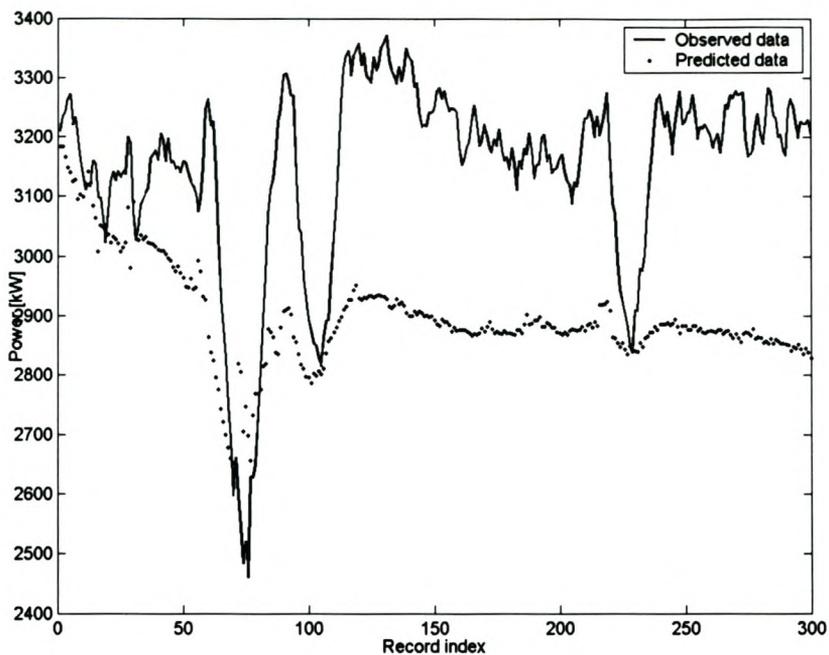
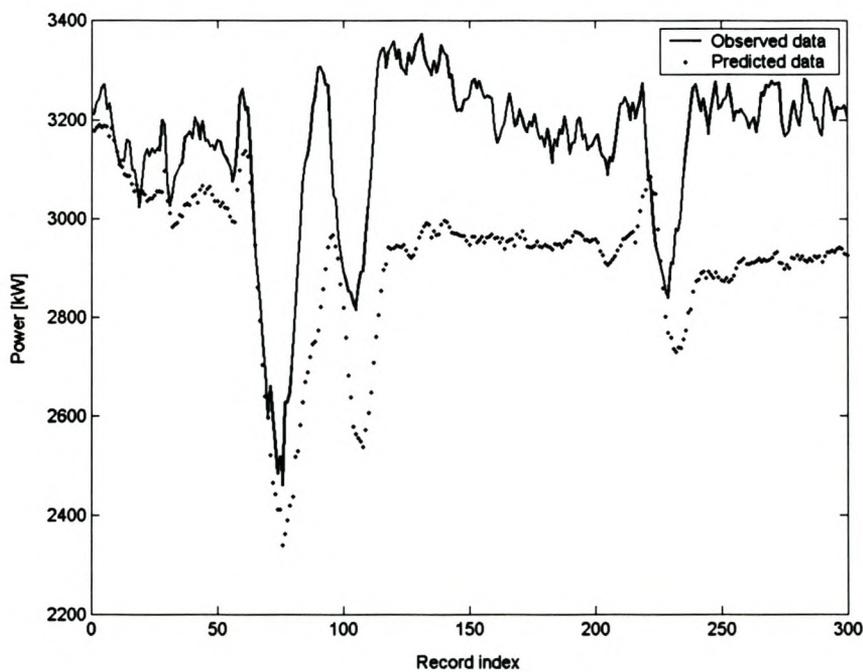


Figure 11.79: Free-run prediction using *MLPSSA\_1*



**Figure 11.80: Free-run prediction using *MLPSSA\_2***



**Figure 11.81: Free-run prediction using *SVM\_1***

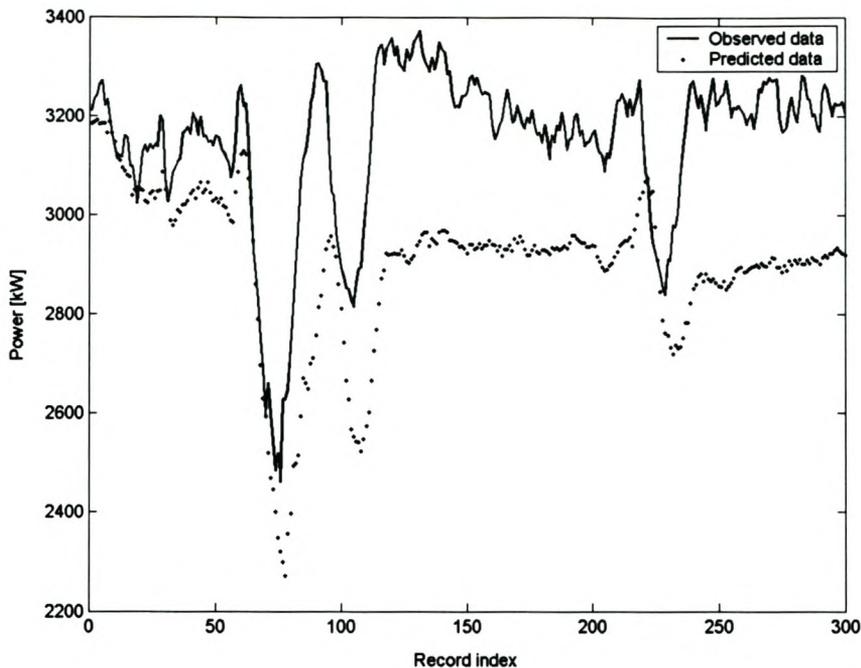


Figure 11.82: Free-run prediction using *SVM\_4*

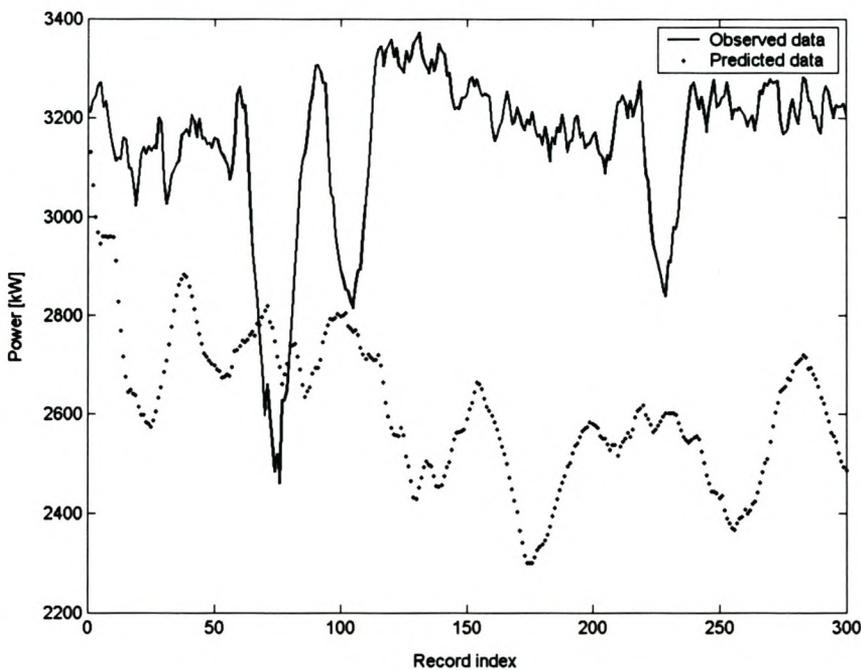
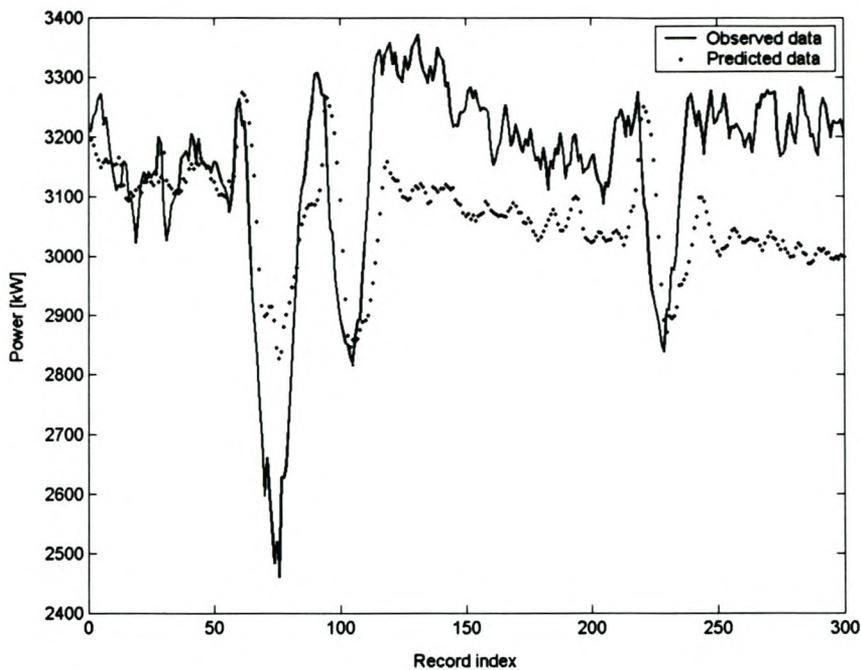
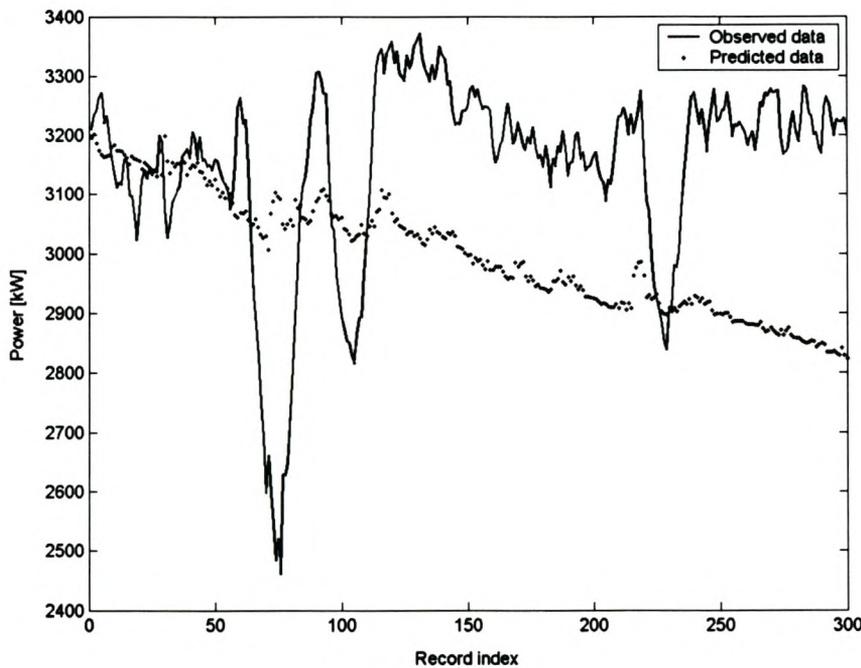


Figure 11.83: Free-run prediction using *RBFMOD\_1*

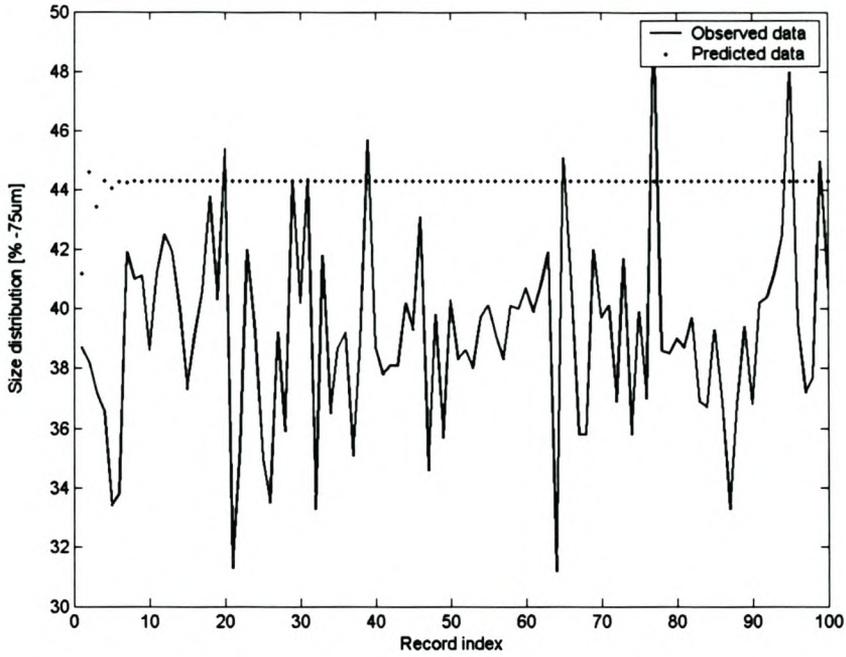


**Figure 11.84: Free-run prediction using *RBFSSA\_1***

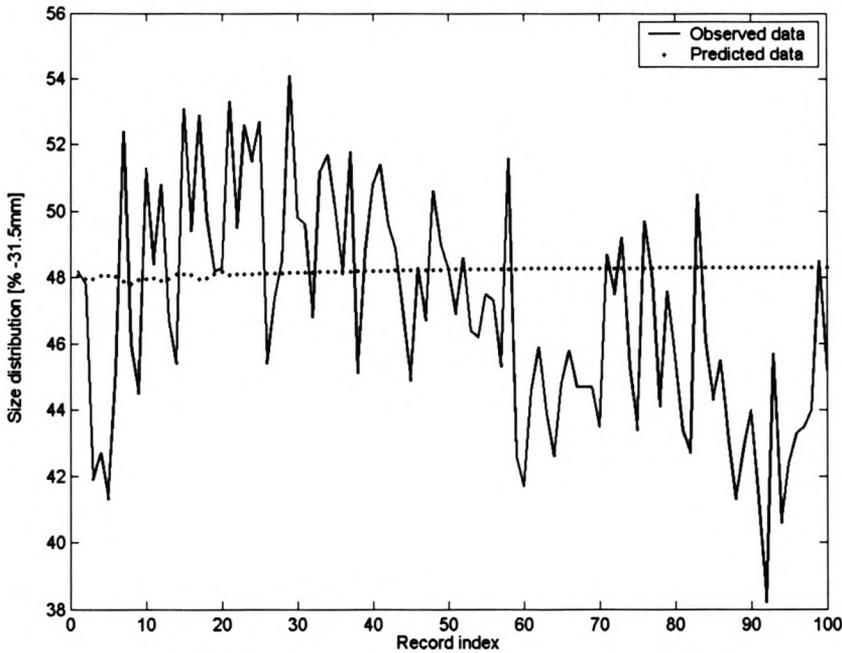


**Figure 11.85: Free-run prediction using *RBFSSA\_2***

### 11.3.9 Wipfrag and Malvern modelling plots



**Figure 11.86: Free-run prediction of FAG mill product size distribution (% - 75µm;  $x_{14}$ )**



**Figure 11.87: Free-run prediction of FAG mill feed size distribution (% - 31.5mm;  $x_{13}$ )**