

**MONITORING OF  
FROTH SYSTEMS  
USING  
PRINCIPAL COMPONENT ANALYSIS**

*by*

**MOHAMED KHARVA**

Thesis submitted in partial fulfilment  
of the requirements for the Degree

**MASTER OF SCIENCE IN ENGINEERING  
(MINERAL PROCESSING)**

in the Department of Chemical Engineering  
at the University of Stellenbosch

*Supervised by*

**Prof. C. Aldrich**

**STELLENBOSCH**

**MARCH 2002**

## **DECLARATION**

I the undersigned, hereby declare that the work contained in this thesis is my own original work, except where specifically acknowledged in the text and has not previously in its entirety or in part thereof been submitted at any university for a degree.

**Signature** :

**Date** : October 2001

## **ABSTRACT**

Flotation is notorious for its susceptibility to process upsets and consequently its poor performance, making successful flotation control systems an elusive goal. The control of industrial flotation plants is often based on the visual appearance of the froth phase, and depends to a large extent on the experience and ability of a human operator. Machine vision systems provide a novel solution to several of the problems encountered in conventional flotation systems for monitoring and control.

The rapid development in computer vision, computational resources and artificial intelligence and the integration of these technologies are creating new possibilities in the design and implementation of commercial machine vision systems for the monitoring and control of flotation plants. Current machine vision systems are available but not without their shortcomings. These systems cannot deal with fine froths where the bubbles are very small due to the segmentation techniques employed by them. These segmentation techniques are cumbersome and computationally expensive making them slow in real time operation.

The approach followed in this work uses neural networks to solve the problems mentioned above. Neural networks are able to extract information from images of the froth phase without regard to the type and structure of the froth. The parallel processing capability of neural networks, ease of implementation and the advantages of supervised or unsupervised training of neural networks make them potentially suited for real-time industrial machine vision systems. In principle, neural network models can be implemented in an adaptive manner, so that changes in the characteristics of processes are taken into account.

This work documents the development of linear and non-linear principal component models, which can be used in a real-time machine vision system for the monitoring, and control of froth flotation systems.

Features from froth images of flotation processes were extracted via linear and non-linear principal component analysis. Conventional linear principal component analysis and three layer autoassociative neural networks were used in the extraction of linear principal components from froth images. Non-linear principal components were extracted from froth images by a three and five layer autoassociative neural network, as well as localised principal component analysis based on k-means clustering. Three principal components were extracted for each image. The correlation coefficient was used as a measure of the amount of variance captured by each principal component.

The principal components were used to classify the froth images. A probabilistic neural network and a feedforward neural network classifier were developed for the classification of the froth images. Multivariate statistical process control models were developed using the linear and non-linear principal component models. Hotellings  $T^2$  statistic and the squared prediction error based on linear and non-linear principal component models were used in the development of multivariate control charts.

It was found that the first three features extracted with autoassociative neural networks were able to capture more variance in froth images than conventional linear principal components, the features extracted by the five layer autoassociative neural networks were able to classify froth images more accurately than features extracted by conventional linear principal component analysis and three layer autoassociative neural networks. As applied, localised principal component analysis proved to be ineffective, owing to difficulties with the clustering of the high dimensional image data. Finally the use of multivariate statistical process control models to detect deviations from normal plant operations are discussed and it is shown that Hotellings  $T^2$  and squared prediction error control charts are able to clearly identify non-conforming plant behaviour.

## OPSOMMING

Flottasie is berug daarvoor dat dit vatbaar vir prosesversteurings is en daarom dikwels nie na wense presteer nie. Suksesvolle flottasiebeheerstelsels bly steeds 'n ontwykende doelwit. Die beheer van nywerheidsflottasie-aanlegte word dikwels gebaseer op die visuele voorkoms van die skuimfase en hang tot 'n groot mate af van die ervaring en vaardighede van die menslike operateur. Masjienvisiestelsels voorsien 'n vindingryke oplossing tot verskeie van die probleme wat voorkom by konvensionele flottasiestelsels ten opsigte van monitering en beheer.

Die vinnige ontwikkeling van rekenaarbeheerde visie, rekenaarverwante hulpbronne en kunsmatige intelligensie, asook die integrasie van hierdie tegnologieë, skep nuwe moontlikhede in die ontwerp en inwerkingstelling van kommersiële masjienvisiestelsels om flottasie-aanlegte te monitor en te beheer. Huidige masjienvisiestelsels is wel beskikbaar, maar is nie sonder tekortkominge nie. Hierdie stelsels kan nie fyn skuim hanteer nie, waar die borreltjies baie klein is as gevolg van die segmentasietegnieke wat hulle aanwend. Hierdie segmentasietegnieke is omslagtig en rekenaargesproke duur, wat veroorsaak dat dit stadig in reële tyd-aanwendings is.

Die benadering wat in hierdie werk gevolg is, wend neurale netwerke aan om die bovermelde probleme op te los. Neurale netwerke is instaat om inligting te onttrek uit beelde van die skuimfase sonder om ag te slaan op die tipe en struktuur van die skuim. Die parallelle prosesseringsvermoëns van neurale netwerke, die gemak van implementering en die voordele van die opleiding van neurale netwerke met of sonder toesig maak hulle potensieel nuttig as reële tydverwante industriële masjienvisiestelsels. In beginsel kan neurale netwerke op 'n aanpassende wyse geïmplementeer word, sodat veranderinge in die kenmerke van die prosesse deurlopend in aanmerking geneem word.

Kenmerke van die beelde van die skuim tydens die flottasieproses is verkry by wyse van lineêre en nie-lineêre hoofkomponentsanalise. Konvensionele lineêre

hoofkomponentsanalise en drie-laag outo-assosiatiewe neurale netwerke is gebruik in die onttrekking van lineêre hoofkomponente uit die beelde van die skuim. Nie-lineêre hoofkomponente is uit die beelde van die skuim onttrek by wyse van 'n drie- en vyf-laag outo-assosiatiewe neurale netwerk, asook deur 'n gelokaliseerde hoofkomponentsanalise wat op k-gemiddelde trosanalise gebaseer is. Drie hoofkomponente is vir elke beeld onttrek. Die korrelasiekoëffisiënt is gebruik as 'n maatstaf van die afwyking wat deur elke hoofkomponent aangetoon is.

Die hoofkomponente is gebruik om die beelde van die skuim te klassifiseer. 'n Probabilistiese neurale netwerk en 'n voorwaarts voerende neurale netwerk is vir die klassifisering van die beelde van die skuim ontwerp. Multiveranderlike statistiese prosesbeheermodelle is ontwerp met die gebruik van die lineêre en nie-lineêre hoofkomponentmodelle. Hotelling se  $T^2$  statistiek en die gekwadreerde voorspellingsfout, gebaseer op lineêre en nie-lineêre hoofkomponentmodelle, is gebruik in die ontwikkeling van multiveranderlike kontrolekaarte.

Dit is gevind dat die eerste drie eienskappe wat met behulp van die outo-assosiatiewe neurale netwerke onttrek is, instaat was om meer variansie by beelde van skuim vas te vang as konvensionele lineêre hoofkomponente. Die eienskappe wat deur die vyf-laag outo-assosiatiewe neurale netwerke onttrek is, was instaat om beelde van skuim akkurater te klassifiseer as daardie eienskappe wat by wyse van konvensionele lineêre hoofkomponentanalise en drie-laag outo-assosiatiewe neurale netwerke onttrek is. Soos toegepas, het dit geblyk dat gelokaliseerde hoofkomponentsanalise nie effektief is nie, as gevolg van die probleme rondom die trosanalise van die hoë-dimensionele beelddata. Laastens word die aanwending van multiveranderlike statistiese prosesbeheermodelle, om afwykings in normale aanlegoperasies op te spoor, bespreek. Dit word aangetoon dat Hotelling se  $T^2$  statistiek en gekwadreerdevoorspellingsfoutbeheerkaarte instaat is om afwykende aanlegwerksverrigting duidelik aan te dui.

***TO MY BELOVED CREATOR  
FOR GIVING ME THE  
TIME AND  
STRENGTH  
TO COMPLETE  
THIS WORK***

## **ACKNOWLEDGEMENTS**

Prof. Chris Aldrich for his time and effort for assisting and guiding me through this wonderful world of research.

Juliana Steyl for always making my life so much easier and pleasant whenever I was in Stellenbosch.

To my family for their support and encouragement in furthering my studies.

To my wife who made it very difficult for me to complete this work but the person who brings a lot of joy in my life.

# CONTENTS

<b>Abstract .....</b>	<b>i</b>
<b>Opsomming.....</b>	<b>iii</b>
<b>Acknowledgements .....</b>	<b>vi</b>
<b>Contents .....</b>	<b>vii</b>
<b>List Of Tables.....</b>	<b>xii</b>
<b>List Of Figures .....</b>	<b>xv</b>
<b>Acronyms.....</b>	<b>xviii</b>
<b>List Of Symbols.....</b>	<b>xixx</b>
Roman Alphabet .....	xix
Greek Alphabet.....	xxii
<b>1 Introduction.....</b>	<b>1</b>
1.1 Background .....	1
1.2 Motivation for this Work.....	3
1.3 Objectives of this Work.....	5
1.4 Overview of this Work.....	6
<b>2 Image Analysis of Froth Systems.....</b>	<b>8</b>

2.1	Contributions Made to the Field of Image Analysis in Flotation.....	9
2.2	Image Processing Algorithms .....	18
2.2.1	Power Spectrum Method.....	19
2.2.2	Spatial Grey Level Dependence Matrix.....	22
2.2.3	Neighbouring Grey Level Dependence Matrix.....	25
2.2.4	Bubble Segmentation via the Morphological Watershed Algorithm .....	30
2.3	Summary .....	36
<b>3</b>	<b>Principal Component Analysis .....</b>	<b>38</b>
3.1	Conventional Linear Principal Component Analysis.....	39
3.2	Non-Linear Principal Component Analysis.....	42
3.3	Principal Component Analysis Using an Autoassociative Neural Network.....	44
3.3.1	Linear Principal Component Analysis Using an Autoassociative Neural Network.....	46
3.3.2	Non-Linear Principal Component Analysis Using an Autoassociative Neural Network.....	48
3.4	Two Approaches For Performing Principal Component Analysis Using an Autoassociative Neural Network.....	52
3.5	Methodology For Training an Autoassociative Neural Network Using the Sequential Method.....	56
3.6	Pre-Processing of Images to be Used as Inputs to an Autoassociative Neural Network.....	57
3.7	Formation of the Input Matrix for the Autoassociative Neural Network.....	61
3.8	Summary .....	63
<b>4</b>	<b>Experimental Results.....</b>	<b>65</b>
4.1	Overview of The Flotation Process and Classification of Surface Froth Images .....	66
4.1.1	Classification of Images.....	67
4.1.2	Number of Images Used for Training and Testing the Autoassociative Neural Network Models.....	69
4.2	Calculation of the Correlation Coefficient and the Z-Statistic.....	70

4.3	Calculation of the Amount of Variance Captured by Each Principal Component .....	71
4.4	Conventional Linear Principal Component Analysis.....	73
4.5	Principal Component Analysis Using an Autoassociative Neural Network.....	76
4.5.1	Linear Principal Component Analysis Using a One Hidden Layer Autoassociative Neural Network.....	76
4.5.2	Non-Linear Principal Component Analysis Using a One Hidden Layer Autoassociative Neural Network.....	82
4.5.3	Non-Linear Principal Component Analysis Using a Three Hidden Layer Autoassociative Neural Network .....	87
4.6	Local Principal Component Analysis.....	92
4.7	Comparison of Techniques for Principal Component Analysis.....	101
4.7.1	First Principal Component.....	101
4.7.2	Second Principal Component.....	102
4.7.3	Third Principal Component.....	103
4.7.4	Total Variance Captured by the First Three Principal Components.....	104
4.8	Significance of a Difference Between Correlation Coefficients of the Three Autoassociative Neural Networks.....	106
4.8.1	Significance of Differences Based on the First Principal Component .....	107
4.8.2	Significance of Differences Based on the Second Principal Component .....	108
4.8.3	Significance of Differences Based on the Third Principal Component .....	110
4.9	Classification of Images .....	112
4.9.1	Input and Target Data Used for Training and Testing the Classifiers .....	114
4.9.2	Probabilistic Neural Network Classifier.....	117
4.9.3	Feedforward Backpropagation Neural Network Classifier.....	124
4.9.4	Probabilistic Neural Network Classifiers Vs Feedforward Backpropagation Neural Network Classifiers.....	128
4.10	Summary .....	130

<b>5</b>	<b>Multivariate Statistical Process Control.....</b>	<b>134</b>
5.1	Theory Behind the Development of Multivariate Statistical Process Control Charts Based on Principal Component Models .....	137
5.1.1	Conventional Linear Principal Component Models.....	137
5.1.2	Multivariate Statistical Process Control Charts.....	139
5.1.3	Principal Component Models Based on Autoassociative Neural Networks .....	144
5.2	Case Study One .....	147
5.2.1	Defining the Normal Operating Conditions.....	147
5.2.2	Standardisation of Data Used for Conventional Linear Principal Component Models .....	149
5.2.3	Conventional Linear Principal Component Models.....	149
5.2.4	Principal Component Models Using Autoassociative Neural Networks .....	159
5.3	Case Study Two.....	177
5.3.1	Conventional Linear Principal Component Models.....	178
5.3.2	Non-Linear Principal Component Model-Autoassociative Neural Network With 3 Hidden Layers .....	183
5.4	Summary .....	188
<b>6</b>	<b>Conclusions.....</b>	<b>191</b>
	<b>References.....</b>	<b>197</b>
	<b>Appendix A: Specifications Of The Autoassociative Neural Networks .....</b>	<b>205</b>
	<b>Appendix B: Froth Images .....</b>	<b>208</b>

## LIST OF TABLES

4.1	Eight different classes of images with the number of images in each class.....	68
4.2	Number of images used from each class for training and testing the autoassociative neural network models.....	69
4.3	Amount of variance captured by the first three linear principal components for each class of data .....	74
4.4	R and Z values of images of classes C1 to C8 for the extraction of the first linear principal component for each class of images .....	77
4.5	R and Z values of images of classes C1 to C8 for the extraction of the second linear principal component for each class of images .....	78
4.6	R and Z values of images of classes C1 to C8 for the extraction of the third linear principal component for each class of images .....	79
4.7	Amount of variance captured by the first three linear principal components.....	80
4.8	R and Z values of images of classes C1 to C8 for the extraction of the first non-linear principal component for each class of images.....	82
4.9	R and Z values of images of classes C1 to C8 for the extraction of the second non-linear principal component for each class of images.....	83
4.10	R and Z values of images of classes C1 to C8 for the extraction of the third non-linear principal component for each class of images.....	84
4.11	Amount of variance captured by the first three non-linear principal components for each class of data .....	85
4.12	R and Z values of images of classes C1 to C8 for the extraction of the first non-linear principal component for each class of images.....	88
4.13	R and Z values of images of classes C1 to C8 for the extraction of the second non-linear principal component for each class of images.....	88
4.14	R and Z values of images of classes C1 to C8 for the extraction of the third non-linear principal component for each class of images.....	89

4.15	Amount of variance captured by the first three non-linear principal components for each class of images.....	90
4.16	Number of images belonging to each cluster for each class of images.....	92
4.17	Weightings of each cluster for each class of images .....	95
4.18	Variance captured by the first three piecewise linear principal components of each class of images .....	96
4.19	Comparison of the amount of variance captured by local PCA and conventional LPCA.....	100
4.20	Comparison of the amount of variance captured by the first principal component using the five principal component analysis techniques for each class of images.....	102
4.21	Comparison of the amount of variance captured by the second principal component using the five principal component analysis techniques for each class of images.....	103
4.22	Comparison of the amount of variance captured by the third principal component using the five principal component analysis techniques for each class of images.....	104
4.23	Comparison of the total amount of variance captured by the first three principal components using the five principal component analysis techniques for each class of images.....	105
4.24	z-Statistics between autoassociative neural networks A, B and C for the extraction of the first principal component from images of classes C1 to C8.....	108
4.25	z-Statistics between autoassociative neural networks A, B and C for the extraction of the second principal component from images of classes C1 to C8.....	109
4.26	z-Statistics between autoassociative neural networks A, B and C for the extraction of the third principal component from images of classes C1 to C8.....	110
4.27	Number of images used to train and test the classifiers.....	115
4.28	Coding of each class of images to be used as the target output during training of the classifier model .....	116
4.29	Number of classifiers developed.....	117

4.30	Percentage of correctly classified non-linear principal components of the training and test data sets by the probabilistic neural network classifier 1.....	118
4.31	Percentage of correctly classified non-linear principal components of the training and test data sets by the probabilistic neural network classifier 2.....	120
4.32	Percentage of correctly classified linear principal components of the training and test data sets by the probabilistic neural network classifier 3.....	122
4.33	Percentage of correctly classified non-linear principal components of the training and test data sets by the feedforward backpropagation neural network classifier 4.....	125
4.34	Percentage of correctly classified non-linear principal components of the training and test data sets by the feedforward backpropagation neural network classifier 5.....	126
4.35	Percentage of correctly classified linear principal components of the training and test data sets by the feedforward backpropagation neural network classifier 6.....	127
4.36	Comparison in the performance of probabilistic neural network classifiers and feedforward backpropagation neural network classifiers.....	128
5.1	Categories to which each class of images belongs.....	147
5.2	The percentage of the total variability explained by the first five linear principal components.....	150
5.3	Correlation Coefficients between original data and reconstructed data using the conventional linear principal component model based on C3.....	151
5.4	Mean SPE between original data and reconstructed data of the model.....	152
5.5	The percentage of the total variability explained by the first five linear principal components.....	155
5.6	Correlation Coefficients between original data (C4) and reconstructed data.....	156
5.7	Mean SPE between original data and reconstructed data of the model.....	157
5.8	Correlation coefficients between original data and reconstructed data.....	160
5.9	Mean SPE between original data and reconstructed data of the model.....	162
5.10	Correlation coefficients between original data and reconstructed data.....	163
5.11	Mean SPE between original data and reconstructed data of the model.....	165

5.12	Correlation coefficients between original data and reconstructed data.....	166
5.13	Mean SPE between original data and reconstructed data of the model.....	168
5.14	Correlation coefficients between original data and reconstructed data.....	169
5.15	Mean SPE between original data and reconstructed data of the model.....	171
5.16	Correlation coefficients between original data and reconstructed data.....	172
5.17	Mean SPE between original data and reconstructed data of the model.....	174
5.18	Correlation coefficients between original data and reconstructed data.....	175
5.19	Mean SPE between original data and reconstructed data of the model.....	177
5.20	The percentage of variance in the data accounted for by each linear principal component.....	178
5.21	Correlation coefficients between original data and reconstructed data.....	180
5.22	SPE associated with each variable over the 99 observations of the NOC, Condition 1 and 2.....	182
5.23	Correlation coefficients between original data and reconstructed data.....	185
5.24	SPE associated with each variable defining the NOCs of the process .....	187

## LIST OF FIGURES

1.1	Summary of each chapter and the way they are linked .....	7
2.1	Illustration of binary reconstruction. In the left image, the dark grey regions represents the markers and the lighter grey regions the mask. Only connected components of the mask, which contain at least a pixel of a marker, are kept and the rest are eliminated. The result is shown in the right-hand image. ....	32
2.2	Watershed terminology .....	34
2.3	Watershed segmentation of flotation froth .....	35
3.1	Autoassociative neural network architecture with a single hidden layer which has a linear activation function .....	46
3.2	Autoassociative neural network architecture with a single hidden layer which has a non-linear activation function.....	49
3.3	Autoassociative neural network architecture with three hidden layers, which have non-linear activation functions, and an output layer with linear activation functions.....	51
3.4	Autoassociative neural network architecture for simultaneous linear and non-linear principal component analysis .....	53
3.5	Neural network architecture for sequential linear and non-linear principal component analysis using an autoassociative neural network.....	55
3.6	Segmentation of a single image (248*248 Pixels) into 64 sub-images (31*31 pixels).....	58
3.7	Rasterization of sub-images to form a vector X which is used as an input into the autoassociative neural network.....	60
4.1	Flowsheet of how the images were split into different classes.....	67
4.2	Plot of the first three linear principal components for images of class C1 and C2 obtained during simulation of the test set by the conventional linear principal component model.....	75

4.3	Plot of the first three linear principal components for images of class C1 and C2 obtained during simulation of the test set by the autoassociative neural network with one hidden layer (Linear activation function) .....	81
4.4	Plot of the first three non-linear principal components for images of class C1 and C2 obtained during simulation of the test set by the autoassociative neural network with one hidden layer (Non-linear activation function) .....	86
4.5	Plot of the first three non-linear principal components for images of class C1 and C2 obtained during simulation of the test set by the autoassociative neural network with three hidden layers (Non-linear activation function).....	91
4.6	Illustration of piecewise linear principal component analysis.....	94
4.7	Plot of the first three non-linear principal components for the 2 and 3 phase froths .....	100
4.8	Process for an image to be classified to a class .....	114
4.9	Plot of the first three non-linear principal components, for images from the test set, for classes C1-C8 which were obtained from an autoassociative neural network with three hidden layers (non-linear activation functions).....	119
4.10	Plot of the first three non-linear principal components, for images from the test set, for classes C1-C8 which were obtained from an autoassociative neural network with a single hidden layer (non-linear activation function).....	121
4.11	Plot of the first three linear principal components for images from the test set for classes C1-C8 which were obtained from an autoassociative neural network with a single hidden layer (Linear activation function).....	123
5.1	SPE control chart indicating a process deviation.....	141
5.2	T <sup>2</sup> control chart indicating a process deviation.....	143
5.3	Multivariate statistical process control via an autoassociative neural network and feedforward backpropagation neural network .....	145
5.4	MSPC models developed using conventional linear principal component models and non-linear principal component models .....	148
5.5	Plot of the control charts based on the conventional linear principal component model of C3.....	153
5.6	Plot of the control charts based on the conventional linear principal component model of C4.....	158

5.7	Plot of the four control charts for the normal operating condition, Condition 1 and Condition 2.....	161
5.8	Plot of the four control charts for the normal operating condition, Condition 1 and Condition 2.....	164
5.9	Plot of the four control charts for the normal operating condition, Condition 1 and Condition 2.....	167
5.10	Plot of the four control charts for the normal operating condition, Condition 1 and Condition 2 .....	170
5.11	Plot of the four control charts for the normal operating condition, Condition 1 and Condition 2 .....	173
5.12	Plot of the four control charts for the normal operating condition, Condition 1 and Condition 2 .....	176
5.13	Plot of the four control charts based on a conventional linear principal component model for the normal operating condition.....	179
5.14	Plot of the four control charts for the normal operating condition, Condition 1 and Condition 2 .....	181
5.15	Plot of the four controls based on the first two non-linear principal components, which defines the normal operating condition of the process.....	184
5.16	Plot of the four control charts based on the first two non-linear principal components for the normal operating condition, Condition 1 and Condition 2.....	186
B.1	2-phase froth - 100% CTAB .....	208
B.2	2-phase froth - 100% CTAB .....	209
B.3	2-phase froth – 90% CTAB.....	209
B.4	2-phase froth - 80% CTAB .....	210
B.5	3-phase froth - 100% CTAB, quartz .....	210
B.6	3-phase froth - 90% CTAB, quartz .....	211
B.7	3-phase froth – 90% CTAB, quartz.....	211
B.8	3-phase froth - 80% CTAB, quartz .....	212

## ACRONYMS

AANN	Autoassociative Neural Network
LPCA	Linear Principal Component Analysis
MSPC	Multivariate Statistical Process Control
NLPCA	Non-Linear Principal Component Analysis
NOC	Normal Operating Condition
PCA	Principal Component analysis
SPE	Squared Prediction Error

## LIST OF SYMBOLS

### Roman Alphabet

$a'$	Direction parameter in spatial grey level dependence matrix method and grey level parameter in neighbouring grey level dependence matrix method (Refers to Section 2.2.2 and 2.2.3)
$a$	Number of principal components retained in the principal component model
$a_j$	Ring sample signature of power spectrum
$d$	Distance metric of the spatial grey level dependence matrix and neighbouring grey level dependence matrix method
$E$	Energy measure of the spatial grey level dependence matrix
$f(i, j, d, a')$	Probability density function of spatial grey level dependence matrix method
$g$	Number of grey levels in an image
$I$	Inertia measure of the spatial grey level dependence matrix
$k, l, m, n$	Co-ordinate pairs $(k, l)$ and $(m, n)$ of the space metric $d$
$L_x$	Horizontal spatial domain element used in the calculation of the correlation parameter of the neighbouring grey level dependence matrix
$L_y$	Vertical spatial domain element used in the calculation of the correlation parameter of the neighbouring grey level dependence matrix
$m$	Number of observations or samples in a data set
$m_a$	Number of annular rings used in the ring sampling geometry of the power spectrum method

$m_s$	Number of slits used in the split sampling geometry of the power spectrum method of the matrix $f(i, j, d, a)$
$m_w$	Number of wedges used in the wedge sampling geometry of the power spectrum method
$n$	Number of observations or samples in a data set which defines the normal operating Condition of a process
$n_1, n_2$	Integer values of two dimensional discrete space signal $x$
$N_x$	Number of resolution cells in the horizontal direction used in calculation of correlation parameter of the spatial grey level dependence matrix
$N_y$	Number of resolution cells in the vertical direction used in calculation of correlation parameter of the spatial grey level dependence matrix
$P$	Power spectrum function $P(\omega_1, \omega_2)$
$p$	Principal components obtained from a principal component analysis
$Q$	Probability distribution function of neighbouring grey level dependence matrix method
$Q(k, s)$	Feature matrix of neighbouring grey level dependence matrix method
$q$	Number of variables in a data set
$R$	Normalising factor used in neighbouring grey level dependence matrix method
$R_a$	Ratio of feature measures of spatial grey level dependence matrix measures in different directions
$S$	Standard deviation of the data set which defines the normal operating condition of the process
$s$	Number of possible neighbours to a pixel in an image in the neighbouring grey level dependence matrix $Q(k, s)$
$s_j$	Parallel- slit sample signature of the power spectrum
$s_i^2$	Estimated variance of the principal components
$t$	Scores obtained from principal component analysis
$t_{i, new}$	New scores obtained from a principal component model

$t_{\alpha/2}$	The upper $\alpha/2$ point of the T-distribution with $n-1$ degrees of freedom
$w_j$	Wedge signature measurement of power spectrum
$x$	Two dimensional discrete space function $x(n_1, n_2)$
$X$	Two dimensional discrete space Fourier Transform of $x(n_1, n_2)$ and representing the amplitude associated with the complex exponential $e^{jw_1n_1} e^{j_1w_2n_2}$
$X^*$	Complex conjugate of $X$
$Y$	Standardised data set ( $m \times q$ ) for principal component analysis
$\hat{Y}$	Approximated standardised data set from a principal component model
$y_{new}$	New multivariate observation to be used in multivariate statistical process control model
$\hat{y}_{new}$	Approximated new multivariate observation by multivariate statistical process control model
$z$	Test statistic for the computing of the difference between two correlation coefficients

## Greek Alphabet

$\varepsilon_n$	Entropy measure as defined by the neighbouring grey level dependence matrix
$\varepsilon_s$	Entropy measure as defined by the spatial grey level dependence matrix
$\rho$	Distance parameter in polar co-ordinates of the sampling geometry used in the power spectrum
$\mu$	Mean of a data set
$\mu_x$	Mean of the row sums of the matrix $f(i, j, d, a)$
$\mu_y$	Mean of the column sums of the matrix $f(i, j, d, a)$
$\sigma$	Standard Deviation of a data set

$\sigma_x$	Standard deviation of the row sums of the matrix $f(i, j, d, a)$
$\sigma_y$	Standard deviation of the column sums of the matrix $f(i, j, d, a)$
$\omega_1, \omega_2$	Frequency components associated with Fourier Transform used in power spectrum method
$\theta$	Angle parameter in polar co-ordinates of the sampling geometry used in the power spectrum

# CHAPTER 1

## INTRODUCTION

### 1.1 BACKGROUND

Mining has been the main source of acquiring mineral resources such as gold, platinum, copper and other mineral ores. The mining of these mineral ores today form an integral part of any country's economy. With the advent of globalisation these mineral ores play a vital role in earning foreign currency for a country and are been mined on a much larger scale than before. As mineral resources are gradually depleted and becoming more difficult to extract on a cost-effective basis it has thus now become more important than ever to maximise mineral recovery from the mined ore.

Flotation is by far the most widely used technique for the concentration of minerals from ore. The process of flotation can extract minerals efficiently from very large quantities of ore very easily. Flotation is a relatively easy process to implement in industry. The equipment and basic principles used in flotation are very simple, but flotation is governed by many mechanisms and influenced by many variables, making it one of the most difficult processes to understand to date. Effective control of this process is of great significance in achieving better recovery and grade of minerals. Improved recovery and

grade of minerals increases productivity thus making the flotation process more cost effective and being able to meet the growing demands of consumer societies.

An effective method to address the problem of improved recovery and grade is to develop advanced control systems that can monitor and control the performance of flotation plants in real-time. This has proven to be a complex task due to the flotation process being difficult to describe fundamentally, owing to the stochastic nature of the froth structures and the ill-defined chemorheology of the froth (Moolman, Aldrich, Van Deventer, 1995a). Therefore flotation remains a poorly understood process that defies generally useful mathematical modelling.

Approximately two decades ago the first on-line devices for measuring the mineral content of flotation slurries became available, which resulted in the first studies in the automatic control of flotation circuits. Since then early optimism has been replaced by the slow realisation that the task of developing effective systems is very difficult (Moolman, Eksteen, Aldrich, Van Deventer, 1996b). These automatic control systems frequently develop technical problems which are not solved that lead to them becoming redundant in flotation plants.

As a result, the control of industrial flotation plants is often based on the visual appearance of the froth phase, and depends to a large extent on the experience and ability of a human operator (Moolman, Aldrich, Schmitz, Van Deventer, 1996a). These type of processes are consequently often controlled suboptimally owing to high personal turnover which leads to operators being inexperienced and lacking in fundamental understanding of plant dynamics. Inaccuracy or unreliability of manual control systems also contributes significantly on optimal control of the plant. The complexity of the task is further aggravated by the fact that different sets of metallurgical parameters may result in similar visual appearances of the froth. As a result, optimal control of the plant is not usually maintained especially where incipient erratic behaviour in the plant is difficult to detect.

Successful flotation operations are often distinguished from less successful operations by the existence of experienced operators who have the ability to take corrective actions based on their visual interpretation of the froth phase. Unfortunately there is no total agreement between different operators on the appearance and interpretation of the surface visual froth characteristics of an ideal froth. The skill is acquired through years of experience and good training. A plant can suffer a tremendous setback if this operator skill is lost. The acquired skill is specific to individual flotation plants and there is no formal source of operator knowledge.

It is not surprising that until recently the relation between flotation performance and the visual characteristics of flotation froths has received little attention in the literature, as much more value is traditionally attached to fundamental approaches than to simple observations that are difficult to quantify but readily described qualitatively (Moolman, Aldrich, Van Deventer, Stange, 1995c).

## 1.2 MOTIVATION FOR THIS STUDY

The development of machine vision systems provides a novel solution to several of the problems encountered in conventional flotation systems for monitoring and control of flotation processes. The rapid development of computer vision, computational resources, artificial intelligence and the integration of these technologies are creating new possibilities in the design and implementation of commercial vision systems for flotation. Machine vision systems have already been applied successfully in diverse areas, such as military, medical technology, character recognition and industrial inspection applications as well as froth flotation systems (Moolman, 1995). The development of an intelligent automation system, which uses machine vision as the sensor component, requires a considerable amount of background information on sensors, signal processing, electronic hardware, illumination, optics, artificial intelligence, and several other aspects. These technologies are widely dispersed through many different specialist reference books and

scientific papers making the development of an intelligent automation system, which uses machine vision as the sensor component difficult.

The use of a computer vision system as the sensor component of an automatic flotation process control system has several advantages over conventional approaches such as:

- Traditional control methods employ operators that visually observe the froth and perform control based on their observations. A machine vision system will perform control in a systematic manner rather than based on visual observations, which differ from one operator to another.
- The system will yield consistent parameters, i.e. if it is shown exactly the same froth at two different times, it will return approximately the same parameters.
- Cameras are non-contacting sensors enclosed in a hard steel casing which is water and dust proof. This makes for a low-maintenance sensing platform in a potentially corrosive environment (the froth).
- Measurements can be made much more regular and quick than human operators are able to do, and consequently necessary corrective measures can be performed regularly and quickly (Botha, 1999).
- All acquired measurement data can be accurately and meticulously recorded.

Apart from the direct benefits for industry the output of a computer vision system for flotation will give new research insight as characteristics that have previously been determined by subjective human judgement can now be described formally in terms of computed features. A successful machine vision system will increase the involvement of the operator in the process by creating a better understanding of what the ideal froth appearance should be. The results of off-line analysis can be combined with the stored images and features extracted previously, and in this way contribute to the more efficient interpretation of froth images and the training of operators (Moolman et al., 1996a).

Previous investigators (Moolman, 1995; Botha, 1999) have made considerable progress with respect to the above mentioned issues. However, there is still significant scope for improvement in these algorithms and in this thesis the use of neural networks to extract features from digitised froth images is considered. The potential advantages of this approach are its simplicity and potentially rapid calculation ability.

### **1.3 OBJECTIVES OF THIS STUDY**

The goal of this work was to reduce images of the surface froth of a flotation process using conventional LPCA and, linear and non-linear PCA via an AANN. The linear and non-linear principal components extracted via the AANN model for each froth image are then to be used to classify to which class of froth each image belonged. The AANN models used to extract the linear and non-linear principal components of each image are also to be used to illustrate the technique of MSPC of a flotation process.

This study aims to:

- Asses the features extracted from digitized images of the surface froths by use of linear and non-linear principal component analysis (PCA) in terms of the variance in the data explained by the principal components.
- To asses the power of the features extracted by linear and non-linear PCA with regard to the classification of surface froth images by a feedforward backpropagation neural network classifier and a probabilistic neural network classifier.
- To investigate to what extent can multivariate statistical process control (MSPC) schemes be used, for the monitoring of froth flotation plants, with PCA and the statistics associated with PCA.

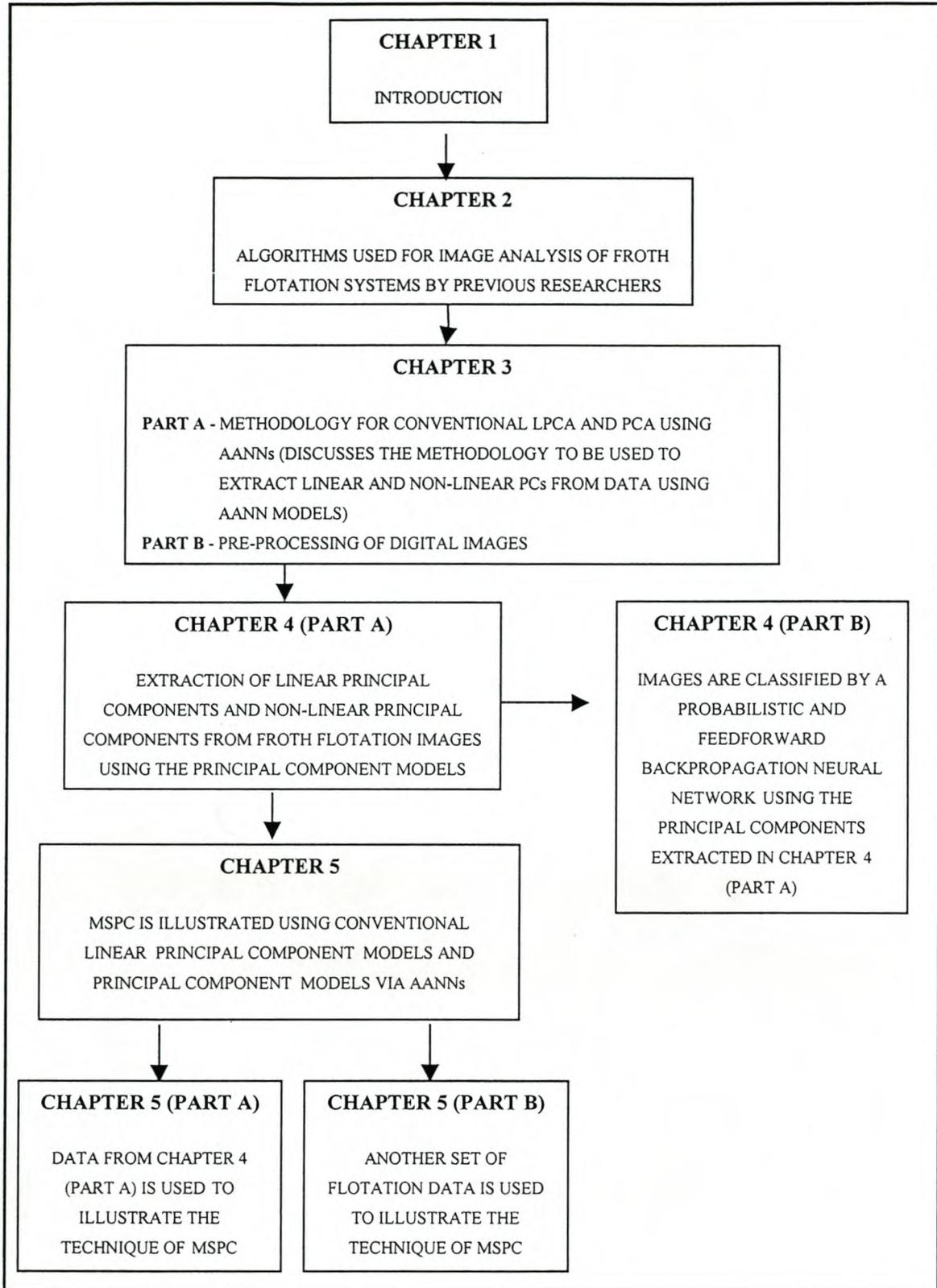
## 1.4 OVERVIEW OF THIS WORK

Chapter 2 documents the specific image processing algorithms and principles that have been used in the machine vision analysis of flotation froth. This chapter is divided into two sections. The first section deals with contributions made to the field of image analysis of froth flotation systems. The second section discusses four low-level image algorithms used in image processing of froth surfaces in flotation processes.

In Chapter 3 PCA and non-linear principal component analysis (NLPCA) is introduced and discussed as a means to data dimensionality reduction and feature extraction. Conventional LPCA and PCA via AANNs are discussed. Two methodologies are introduced to perform PCA via an AANN. The first methodology uses an AANN to extract the principal components simultaneously while the second methodology uses an AANN to extract the principal components sequentially. The chapter is concluded with a discussion on the pre-processing of digital froth flotation images to be used in Chapter 4. Results of conventional LPCA and PCA using an AANNs are compared in the extraction of features from digital images in Chapter 4. The comparison is based on the calculation of the correlation coefficient and Fishers Z Transformation. The significance of difference between correlation coefficients of the different AANNs is computed by using the z statistic. Principal components extracted for each image were used for classification by a probabilistic neural network model and a feedforward backpropagation neural network.

Chapter 5 discusses the use of conventional linear principal component models for MSPC of flotation processes as well as principal component models via AANN models. Two sets of flotation data are used to illustrate the technique of MSPC.

Chapter 6 contains a summary of this work and re-iterates the important knowledge that was gained. Possibilities of future work on this subject are also discussed. Figure 1.1 gives an overview of the layout of this thesis along with a short description of what is done in each chapter.



*Figure 1.1: Summary of each chapter and the way they are linked*

## CHAPTER 2

### IMAGE ANALYSIS OF FROTH SYSTEMS

Froth systems has long been known to be one of the most poorly understood and difficult processes to control. The control of industrial froth flotation systems is complex and difficult. In practice the control of industrial froth flotation systems is often based on the visual appearance of the froth phase, and depends to a large extent on the experience and ability of a human operator. As a result, these types of processes are frequently controlled suboptimally owing to high personal turnover, inaccurate modelling due to lack of understanding of the flotation process and unreliable control systems etc.

To overcome the above-mentioned problems a new approach to the automatic control of the froth flotation process has being taken up in the last ten years. The new approach uses on-line machine vision systems to perform image analysis of the surface froth of flotation processes. Computational vision is used as the sensor component. A large number of researchers have proposed different image analysis techniques for the analysis of the surface froth of flotation processes.

## 2.1 CONTRIBUTIONS MADE TO THE FIELD OF IMAGE ANALYSIS IN FLOTATION

One of the early investigations of machine vision systems in froth flotation was done by Woodburn, Stockton & Robins (1989) in the characterisation of three phase froth structures prevalent in flotation ore concentrators in work pertaining to the demineralisation of low rank coals. Woodburn et al (1989) processed images off-line by grouping together contiguous pixels of brightness greater than or equal to a certain threshold. Frequency distributions were then derived from the calculation of the area, perimeter and moments of area of these features about the x-axes and y-axes. From the segmentation method used by Woodburn et al. (1989) it can be concluded that they segmented froth images using bubble highlights. The images had to be made and processed off-line and therefore could not be used in an on-line automatic control system of a froth flotation process.

A further contribution was made to froth flotation using computer vision analysis by Symonds & De Jager (1992) in 1992. They proposed the segmentation of froth images into bubbles using mathematical morphological<sup>1</sup> image processing techniques. These bubbles could then be counted and analysed to characterise the segmented froth images. The morphological segmentation algorithm that they proposed was a residual edge detector using a spherical structuring element. By opening and closing the image with a spherical structuring element<sup>2</sup> (“Rolling ball algorithm”) and the subtracting the opened and closed images from the original image, a peak image and a valley image could be obtained (Botha, 1999). Although morphological image processing techniques was an improvement over conventional techniques, the image analysis was complicated due to the complexity involved in trying to identify individual bubbles in the froth and a number

<sup>1</sup> Mathematical morphological is an approach to image processing that uses shape for the extraction of image features.

<sup>2</sup> The closing or opening of an image using a spherical structuring element, is known as the “Rolling Ball Algorithm”

of pre-processing and post-processing steps were required to segment and analyse images.

The methods proposed by Woodburn et al. (1989) and Symonds & De Jager (1992) were not entirely successful due to the difficulty involved in trying to identify individual bubbles in the froth structure.

Due to the lack of success of the methods employed by Woodburn et al. (1989) and Symonds and De Jager (1992) a new approach was taken by Moolman (1995). His approach towards image analysis was an indirect approach that used a texture classification rather than a direct edge detection approach. Moolman (1995) characterised the textural properties of images of the froth surface obtained from copper and platinum floatation plants using second order statistics. The textural features that were extracted were numerical features based on the power spectrum of the fast Fourier transform, neighbouring grey level dependence matrix and spatial grey level dependence matrix. These textural features were used to train self-organised mapping and learning vector quantization neural networks for classification of images based on these textural features. Using this technique Moolman (1995) was able to accurately classify froth type and detect froth mobility.

Eksteen (1995) developed a knowledge base for the research and development of vision-base automatic froth control systems from a chemical engineering viewpoint. His work utilises the image processing software and algorithms developed by Moolman (1995). This work concluded that a vision based control system for flotation control is commercially viable if all the parameters discussed by Eksteen (1995) are considered in the design and implementation of such a system. As a general overview of the factors pertinent to the development of a complete vision-based automatic flotation control system, this is a valuable work. From an image processing point of view, it contains a more detailed discussion of the methods employed by Moolman (1995).

It is important to note that by this time, there existed two distinct approaches towards image processing of the froth surface of flotation images. The first approach attempted to analyse images of the surface froth in a hierarchical and direct manner by attempting an accurate segmentation of the froth surface into bubbles, and then calculating higher level parameters using this segmentation (Symonds & Jager, 1992; Woodburn et al., 1989). The second approach related distinct froth types to textural properties and could indirectly derive other froth surface visual parameters from texture (Moolman, 1995).

In 1996 Hargrave, Miles, & Hall (1996) used grey level measurements in predicting coal flotation performance. They have shown that images of the coal flotation froth can be described in terms of its grey level and bubble size distribution. Their method is based on the fact that texture of froths are good qualitative as well as quantitative indicators of the performance of the floatation process. Their work showed that correlation's were observed between a calculated mean grey level of froth and various flotation parameters such as ash content and froth mass flow rates.

In 1997 Sader-Kazemi & Cilliers (1997) made the first successful attempt at complete bubble segmentation. They used the morphological watershed algorithm to segment froth images into bubbles. All the components used in the algorithm were standard image processing routines, which are available in most image processing software. They investigated coal flotation on a laboratory scale, fluorspar separation on a pilot scale and segmentation of a rock pile into individual rocks. They have shown that their algorithm is able to accurately detect bubbles of different sizes in coal flotation and fluorspar separation, which is often problematic. The algorithm was also effectively used to segment a rock pile into individual rocks. This algorithm using reconstruction and the watershed as the critical steps, can be generally applied to flotation froths from a variety of minerals and conditions (Sader-Kazemi & Cilliers, 1997). This algorithm requires no manual intervention and may therefore be suitable for application in automated industrial froth characterisation and control systems.

Niemi, Ylinen & Hyotyniemi (1997) has applied image analysis to images of the froth surface of apatite, sphalerite and pyrite, in two industrial flotation plants in Finland. They extract features from froth images using neurocomputing methods for automatic construction of features, which is closely related to statistical factor analysis. These methods aim to construction of new variables, on the basis of statistical and physical variables observed which are often correlated with each other (Niemi et al., 1997). The generalisation of the generalised hebbian algorithm has been used by Niemi et al. (1997) to extract features from the froth images. The features capture statistical relationships between the data, and the number of features to be extracted from the images is an adjustable parameter in the algorithm. A fast estimate of the average size of bubble distributions was obtained by means of a Fourier analysis of the bright/dark texture of the froth images. Four features were extracted using the image histogram data and used as input to the generalisation of the generalised hebbian algorithm by Niemi et al. (1997). However, they found that no more than two features were needed. Niemi et al. (1997) analysed black and white images of the froth and their brightness distributions by means of neural networks and has proved that induced parameters can be automatically found which relate to the froth structure.

Banford, Atkas & Woodburn (1998) have done work in the regulation of the performance of coal flotation cells by using image analysis to define a desired bubble size in the concentrate. They used an image processing technique based on edge detection. The technique involved identifying the minima on a linear scan of the pixel intensity function against pixel position. Banford et al. (1998) used this technique to overcome the problem associated with curved surfaces of bubbles. Curved surfaces of bubbles create the problem of variation in surface reflectance, which makes it difficult to use a single brightness threshold to differentiate between features (Banford et al., 1998). The distribution of feature areas computed from the segmented images has been used by Banford et al. (1998) to estimate the froth bubble size distributions and specific surface areas. This technique is effective in distinguishing the individual features of froth with distinctively different structures and has successfully been used by Banford et al. (1998) to characterise froth structure off-line. With their technique, they have also been able to

estimate the specific surface of the froth bubbles, and relate these to reagent addition and cell performance.

An expert system for the supervision of flotation plants based on Aceflot has been developed. Aceflot is one of the first real time analysers of the characteristics of the froth that is formed on the surface of flotation cells and has been developed by Cipriano, Guarini, Vidal, Soto, Sepulveda, Mery & Briseno (1998). The Aceflot analyser is based on image processing and measures several physical variables of the froth, including colorimetric, geometric and dynamic information. Aceflot is an analyser that measures the colour, number, size, shape, density, speed and stability of the froth bubbles produced on the surface of a flotation cell (Cipriano et al., 1998). Aceflot also determines the magnitude of the speed and the moving direction of the bubbles, as well as froth stability, which is a measurement of the rate of bubble explosions. Aceflot also keeps a historical register of the measured variables and displays current results and trends in time (Cipriano et al., 1998). An expert system based on Aceflot detects abnormal operation states and proposes corrective actions, such as changing level set points of modifying the dose of reagents.

Aceflot measures the colour of the froth by taking the mean intensity value of each of the red, green and blue image components. An edge detection algorithm is used to determine the number, size, shape and density of froth bubbles. The algorithm operates by localising the centre of the bubbles by recognising the shiniest zones of the image, and the edges of the bubbles by recognising the darkest zones. Once bubble centres and edges are detected it is possible to compute the area, perimeter, diameter and circularity of the bubbles. The froth velocity and stability are computed through the processing of consecutive images, at a rate of 20 frames per second. The stability is estimated comparing two consecutive image frames, and evaluating a measurement of the rate of change in appearance. Aceflot has been installed in a copper mine in Chile.

Mintek South Africa has taken a different approach towards the development of an on-line machine vision system for froth flotation. The approach they use is not based on

image analysis of the froth but rather on the measuring of the red, blue and green intensity values of the froth by a colour meter. Their objective in the development of the colour meter was to provide a simple and inexpensive means for measuring the grade of the froth online by measuring the colour of the froth. The colour meter consists of a probe with red, blue, green diodes mounted inside a waterproof casing. A Polaroid lens is placed in front of the diodes to polarise the light before reaching the diodes. Each diode is connected to a box where the signal received from the diode is converted to a voltage signal, which in turns is sent to a computer program. In 1999 Mintek South Africa tested their colour meter on a copper nickel flotation pilot plant. Tests on the colour metre have shown that it is very insensitive to subtle changes in the shade of dark colours but can register change in bright colours. The change in the level of the froth and ambient light conditions also has a significant influence on the readings of the colour meter. It can be concluded that the colour meter has not been entirely successful.

The Julius Kruttschnitt Mineral Research Centre in Australia developed the first commercial system for the on-line monitoring of flotation froth which is been currently marketed as JKFrothCam. JKFrothCam captures, digitises and analyses images from flotation froths on-line to assist control room operators and to provide data for automatic process control. JKFrothCam is based on a combination of texture spectrum analysis and pixel tracing techniques to quantify three surface characteristics of the froth used by operators to evaluate flotation performance:

- Static characteristics: Bubble size and froth texture.
- Dynamic characteristics: Froth velocity.

JKFrothCam has been successfully used on a plant scale at BHP Coal Preparation Plant in central Queensland (Australia) to control a Microcell flotation column. Control of the Microcell flotation column by JKFrothCam has given superior metallurgical performance compared to the cells operating on manual control. JKFrothCam has also successfully been used at BHP Minerals copper flotation plant in Chile. In a statistically designed two week cell trial, process control using JKFrothCam increased copper recovery by 1.2% at

the 92% confidence level, compared with manual operation. They expect to improve copper recovery with a more comprehensive control strategy using JKFrothCam.

In 1999 Botha (1999) derived a complete set of flotation froth surface computational vision analysis algorithms. These algorithms are able to perform a complete static and dynamic analysis on digital images of the froth. The static analysis includes a full bubble segmentation where the perimeter of each bubble is determined using watershed morphology and a number of other parameters could be calculated from the segmentation results. Textural parameters are also calculated as part of the static results and can be used as auxiliary indicators of other parameters. The dynamic analysis yields parameters that are an indication of froth mobility.

The algorithms that Botha (1999) has developed as discussed above have been implemented by him in a practical software package for integration in online-plant automatic control systems. The sensor platform, consisting of the algorithm software, a graphical user interface and other functionality, is able to acquire live video from the multiple input channels and perform complete froth analysis in real-time (Botha, 1999). The shortcomings of Botha's (1999) software package is that if the froth is very fine then the bubble size cannot be accurately measured and it is a very cumbersome technique in that it takes time to run the algorithm. The whole system is presently being commercially marketed.

Wright, Sweet, Francis, Jager, Bradshaw & Cilliers (2000) have developed a machine vision system to perform on-line analysis of froth images for a platinum flotation plant. The system analyses froth images and determines bubble sizes and shapes, froth colour characteristics and froth velocity information. The system is able to identify perturbations in reagent addition rates by tracking changes in bubble size information. The speed at which the system segments and analyses images is said to be sufficiently high for it to be a useful real-time monitoring tool for froth flotation (Wright et al., 2000). Wright et al. (2000) extracts bubble size and shape information from a froth image by segmenting an image using the fast watershed transform. The fast watershed transform is able to

segment an image in a few seconds, as opposed to several minutes for classical versions of the algorithm, making it suitable for use in an on-line analysis system. Following the segmentation, size and shape information is extracted from the bubbles. Parameters such as average bubble size, bubble size distributions, eccentricity (the ratio of the major and minor axes of the bubble) and circularity are calculated. Parameters related to the whole froth, such as froth density (the number of bubbles per unit area of the froth) and mean froth colours are also calculated (Wright et al., 2000). The froth velocity is determined by using a block matching algorithm known as pixel tracing. The on-line analysis system comes with a user interface, which is written in Visual Basic.

Hytyniemi & Ylinen (2000) of Finland have used image analysis of flotation froth, in an experiment to find features corresponding to the poisoning phenomena in a flotation cell. The approach they present which combines the feature extraction and classification phases, is applied to the automatic “structuring” of the image information. Only intensity histogram and Fourier transform information is used; it needs to be noted that the colour information, the velocity of the froth, and many other factors that also convey information have not been used by them in their experiment. They use a new sparse coding method, which is based on a generalisation of the generalised hebbian algorithm, which is realised by a modification of the kohonen network, for image analysis of flotation froth. Hytyniemi & Ylinen (2000) have used image data from a floatation plant in Finland where the ore containing the minerals sphalerite and pyrite is purified. They applied the generalisation of the generalised hebbian algorithm first to only the intensity histogram data of the images and after that, the histogram data was augmented with power spectrum data, which was calculated using the fast Fourier transform. From just using the histogram data they were able to detect the poisoning phenomena and when they used the histogram data together with the fast Fourier transform data the poisoning phenomena became even more distinguishable and was detected earlier. From the experimental analysis they have shown that their method can predict poisoning before ordinary instruments. The one drawback of this technique, stated by Hytyniemi & Ylinen (2000) is that there is a high sensitivity to irrelevant noise peaks.

It is important to note that some types of froth images are more difficult to analyse and classify than others. The base metals, coal and platinum froth images each have their own characteristics. With the base metal froths operators relate the grade of the froth to the colour of the froth. A base metal froth of high grade results in images in which these minerals are responsible for a high light intensity in the bright spots on top of the bubbles. On the other hand, the grade of platinum and coal froths cannot be associated with the colour and mineralogy of the froth owing to the complexity of the ore and the uniform grey colour of the froth. The difference in colour of different grades of platinum and coal froths is more subtle than that of base metal froths (Moolman, 1995). In platinum flotation plants operators attach little significance to the colour of platinum froths in determining the grade but attach more significance on the structure and dynamics of the froth in determining the grade.

The image analysis techniques presented above have been developed on specific types of mineral froths. Although of more recent time their application has been increased to other mineral froths with a fair amount of success. The one area where present image analysis techniques still has difficulty is in analysing fine froths. This has proven to be an elusive goal in the field of image analysis of froth systems. At present there are only two commercially available systems for the online monitoring of froth flotation systems namely JKFrothCam and Frothmaster. It could not be determined from the literature if the other froth flotation systems, which have been developed, are commercially available.

## 2.2 IMAGE PROCESSING ALGORITHMS

From the review of work done in the field of image analysis in froth flotation processes in the previous section it is clear that two distinct approaches have been attempted. Both these approaches have been successful to a certain degree. The one approach used analysed froth surfaces in a direct fashion by attempting an accurate segmentation of the froth surface into bubbles. Higher level parameters were then calculated using this segmentation. The other approach used an indirect method to analyse the froth. Distinct froth types were related to textural properties and these textural properties were used indirectly to calculate surface visual parameters.

These two approaches have led to a number of different image processing algorithms being used in image processing of froth surfaces. A discussion of the most well established image processing algorithms of each approach will be discussed in a little more detail. With regard to the indirect approach three statistical techniques would be discussed. They are the power spectrum method, which is derived from the Fast Fourier Transform, spatial grey level dependence matrix and the neighbouring grey level dependence matrix. In the direct approach the different stages of the complete bubble segmentation via the watershed algorithm is discussed since it has very successfully been used in developing a real-time machine vision system.

### 2.2.1 POWER SPECTRUM METHOD

The Fast Fourier Transform is extensively used in signal processing, on which there is an abundance of information available in the field of image analysis. The Fast Fourier Transform is a powerful and well established technique for the extraction of useful features from digital images. The use of the Fast Fourier Transform in feature extraction in two-dimensional spaces (i.e. images) is less common (Moolman, 1995). By making use of the power spectrum of a Fast Fourier Transform, a pattern space closely related to the pre-transformed image is generated.

The Fourier Transform (Equation 2.1) can be used to transform the digitised image from the spatial to the frequency domain.

$$X(\omega_1, \omega_2) = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} x(n_1, n_2) e^{-j\omega_1 n_1} e^{-j\omega_2 n_2} \quad (2.1)$$

where  $\omega_1, \omega_2$  is the frequency components associated with the Fourier Transform used in the power spectrum, and  $n_1$  and  $n_2$  are integer values of two dimensional discrete space signal  $x$ .

The inverse Fourier Transform (Equation 2.2) can be used to reconstruct the image from the frequency domain.

$$x(n_1, n_2) = \frac{1}{(2\pi)^2} \int_{\omega_1=-\pi}^{\pi} \int_{\omega_2=-\pi}^{\pi} X(\omega_1, \omega_2) e^{j\omega_1 n_1} e^{j\omega_2 n_2} d\omega_1 d\omega_2 \quad (2.2)$$

The power spectrum can be defined by Equation 2.3 below which contains only real numbers. The asterisk (\*) denotes the complex conjugate and  $P$  denotes the sample power spectrum. The power spectrum of an image is unique, but the converse is not necessarily true.

$$P(w_1, w_2) = X(w_1, w_2) \cdot X^*(w_1, w_2) = |X(w_1, w_2)|^2 \quad (2.3)$$

where  $P$  is the power spectrum function

The summation of coefficients in various geometric forms is sometimes used in reducing the dimensionality of the power spectrum (Caudill, 1990). The features most commonly used with the power spectral method are:

- Ring sampling geometry
- Wedge sampling geometry, and
- Parallel slit sampling geometry

The ring and wedge sampling geometry's can be presented in a simpler mathematical form by transforming the sampling space to the polar co-ordinate system. The ring sampling geometry can be written as Equation 2.4.

$$a_j = \int_0^\pi \int_\rho^{\rho_j + \Delta\rho} P(\rho, \theta) \rho \, d\rho \, d\theta \quad j = 1, 2, \dots, m_a \quad (2.4)$$

( $m_a$  is the number of annular rings)

$$\text{where } \rho = \left[ (w_1)^2 + (w_2)^2 \right]^{0.5} \quad (\text{Distance parameter in polar co-ordinates}) \quad (2.5)$$

$$\text{and } \theta = \tan^{-1} \left( \frac{w_2}{w_1} \right) \quad (\text{Angle parameter in polar co-ordinates}) \quad (2.6)$$

and  $a_j$  is the ring sample signature

The wedge sampling geometry can be written as Equation 2.7.

$$w_j = \int_{\rho_{\min}}^{\rho_{\max}} \int_{\theta_j}^{\theta_j + \Delta\theta_j} P(\rho, \theta) \rho \, d\rho \, d\theta \quad j = 1, 2, \dots, m_w \quad (2.7)$$

( $m_w$  is the number of wedges)

and the spilt sampling geometry can be written as Equation 2.8.

$$s_j = \int_{-w_{2\min}}^{w_{2\max}} \int_{w_{1j}}^{w_{1j} + \Delta u} P(w_1, w_2) dw_1 dw_2 \quad j = 1, 2, \dots, m_s \quad (2.8)$$

( $m_s$  is the number of slits)

where it is assumed that the power spectrum is first rotated to a desired direction ( an angle  $\theta$  from a reference direction) before the measurements are taken (Eksteen, 1995).

The angular distribution of values in the power spectrum is sensitive to the directionality of the texture in the image (Weszka, Dyer & Rosenfeld, 1976). A texture with many edges or lines in a given direction  $\theta$  will have high power spectrum values around the orthogonal direction ( $\theta + \pi/2$ ). In a non-directional texture the power spectrum would also be non directional. The directional features are best extracted by using the averages of wedge-shaped regions about the origin (Moolman, 1995).

The frequency domain gives a highly compact description of the image, since most of the energy of the image is concentrated around the origin. This idea was extended by Moolman (1995) to reduce the dimensionality of the power spectrum of the Fast Fourier Transform, as follows

- Set all coefficients smaller than a limit L equal to zero.
- Obtain the inverse Fast Fourier Transform, reconstruct the image and decide if the result is acceptable.
- If so, accept the value of L and retain the reduced Fast Fourier Transform.
- If not, reduce the limit L and repeat the procedure until an acceptable solution is obtained.

Weszka et al. (1976) has found that the power spectrum method yields good classification results. The drawback of the power spectrum method is that it gives a frequency representation of an image, which is not easily interpreted. The Power spectrum method may be strongly influenced by aperture effects, which influence the power spectrum of an

image. It is reported by Eklundh (1979) that phase information does not yield useful texture measures. Haralick (1979) has said that transforms have been used with some degree of success, but that these methods have contributed little to the understanding of texture, because they only use some mathematical transformation, which assigns numbers to the transformed images in a non-specific way.

### 2.2.2 SPATIAL GREY LEVEL DEPENDENCE MATRIX

Haralick, Shanmugan & Dinstein (1973) defined texture as the spatial (statistical) distribution of tonal variations within a band, where tone is based on the varying shades of grey of the resolution cells. Haralick et al. (1973) developed this technique on the assumption that the textural information in an image was contained in the overall spatial relationship between the grey tones in the image.

The spatial grey level dependence matrix is based on the estimation of the second-order joint conditional probability density functions.  $f(i, j, d, a')$ ,  $a' = 0^\circ, 45^\circ, 90^\circ, 135^\circ$ . Each  $f(i, j, d, a')$  is the probability of going from grey level  $i$  to grey level  $j$ , given that the intersample spacing is  $d$  and the direction is given by angle  $a'$ . If an image has  $g$  grey levels, then the density functions can be represented as  $g \times g$  matrices. Each matrix can be computed from a digital image by counting the number of times each pair of grey levels occurs with a separation distance  $d$  and in the direction specified by angle  $a'$ . It is assumed that the textural information is sufficiently specified by the full set of four spatial grey level dependence matrices (Moolman, 1995).

A set of measures for characterising these matrices have been developed by Haralick et al. (1973). The features most often used are

$$\text{Energy : } E = \sum_i \sum_j [f(i, j, d, a')]^2 \quad (2.9)$$

This is a measure of the homogeneity of the image. For an image, which is not homogeneous, the matrix will have a large number of small entries off the diagonal, and hence the energy will be small. The diagonal and region close to the diagonal represent transitions between similar grey levels (Moolman, 1995).

$$\text{Entropy : } \varepsilon_s = \sum_i \sum_j [f(i, j, d, a') \cdot \log(f(i, j, d, a'))] \quad (2.10)$$

The entropy is a measure of the complexity of the image.

$$\text{Inertia : } I = \sum_i \sum_j [(i - j)^2 f(i, j, d, a')] \quad (2.11)$$

The inertia indicates the number of local variations in the image.

$$\text{Local Homogeneity : } LH = \sum_i \sum_j \left[ \frac{f(i, j, d, a')}{1 - (i + j)^2} \right] \quad (2.12)$$

Local Homogeneity shows the tendency of similar grey levels to be clustered.

$$\text{Correlation : } CORR = \sum_i \sum_j \left[ (i - \mu_x)(j - \mu_y) \cdot \left( \frac{f(i, j, d, a')}{\sigma_x \sigma_y} \right) \right] \quad (2.13)$$

The correlation is a measure of the grey level dependencies in the image.

$$\text{where } \mu_x = \frac{1}{N_x} \sum_{L_x} f(i, j, d, a') \quad (2.14)$$

is the mean of the row sums of the matrix

$$\text{and } \sigma_x = \sqrt{\frac{\sum_{L_x} [f(i, j, d, a') - \mu_x]^2}{N_x - 1}} \quad (2.15)$$

is the standard deviation of the row sums of the matrix and

$N_x$  - is the number of resolution cells in the horizontal direction.

$L_x = \{1, 2, \dots, N_x\}$  is the horizontal spatial domain element.

The values of  $\mu_y$  and  $\sigma_y$  are defined similarly by substituting all x-subscripts with y-subscripts and y refers to the vertical direction or column sums.

In the above Equations the distance  $d$  is defined by Haralick et al. (1973) as

$$\text{Distance : } d[(k, l)(m, n)] = \max [ |k - m|, |l - n| ] \quad (2.16)$$

where  $(k, l)$  and  $(m, n)$  are the co-ordinates of two pixels separated by a distance  $d$ .

The Ratio ( $R_a$ ) has being defined by Aldrich, Moolman, Eksteen & Van Deventer (1995) to account for the extraction of directional information in an image.

$$\text{Ratio : } R_a = \frac{E_{a=90}}{E_{a=0}} \quad (2.17)$$

One of the difficulties in using the spatial grey level dependence matrix is the choosing of the intersampling spacing  $d$  and that it is angularly dependent. The discriminatory power of the spatial grey level dependence matrix improves when several intersampling spacing distances are used. The spatial grey level dependence matrix is insensitive to  $180^\circ$  rotations of textures. Connors & Harlow (1980) have found that the energy, entropy, correlation, local homogeneity and inertia measures which are defined in Equations 2.10-2.15, respectively, and which are commonly used with the spatial grey level dependence matrix do not contain all the important texture-context information contained in the spatial grey level dependence matrices. Connors & Harlow (1980) compared the spatial grey level dependence matrix method with that of the Power spectrum method and

concluded that the spatial grey level dependence matrix is more powerful than the Power spectrum method. A similar conclusion was reached by Weszka et al. (1976).

### **2.2.3 NEIGHBOURING GREY LEVEL DEPENDENCE MATRIX**

The neighbouring grey level dependence matrix methods were developed as an alternative method for the extraction of useful texture features from an image. Some of the statistical approaches for texture feature extraction and classification are one-dimensional, such as the Power spectrum method, grey level difference statistics, second order grey level statistics and grey level run statistics. They are based on angularly dependant features. The neighbouring grey level dependence matrix method of feature extraction was developed by Sun & Wee (1983) to overcome the problem of angular dependent features experienced in the above methods.

Texture is composed of elements or “local orders” which are smaller than the area where they are repeated. Therefore, an image texture is described by the numbers, types, and properties of its elements and their spatial organisation or layout (Sun & Wee, 1983). Therefore the interaction between an element and its surrounding neighbours in an image is very critical. Haralick et al. (1973) extended the concept of the texture transform to the general definition of pixel co-occurrence, which is two-dimensional and does not necessarily impose an angular direction on the co-occurrence matrices.

The angular dependant features present a special problem. To solve this problem, Haralick et al. (1973) suggested that angularly dependent features not be used directly. Instead, he suggested that two functions of the features, their average and range (which are invariant under rotation), be used as the input of the classifier. The above procedure provides a solution to the problem of using angularly dependent features for texture classification.

Texture can be described by tone and its statistical characteristics of variation. Texture features of an image are defined by the overall spatial relationship, which the grey tones have in the image to one another. An approach using angularly independent features, by considering the relationship between an element and all its neighbouring elements at one time instead of one direction at a time, and tries to eliminate the angular dependency. While at the same time reducing the calculation required to process an image.

It is assumed that a grey tone spatial dependence matrix of an image can adequately specify this texture information. This matrix, named neighbouring grey level dependence matrix, is computed from the grey tone relationship between every element in the image and all of its neighbours at a certain distance. All of the texture features are derived from this neighbouring grey level dependence matrix.

### 2.2.3.1 Texture Features Extracted From The $Q$ Matrix

The assumption in characterising image texture is that all the texture information is contained in the neighbouring grey level dependence matrix  $Q$ .

The neighbouring grey level dependence matrix as calculated on an image or image region contains information about the relationship between each of the pixels and all its neighbouring pixels at a certain distance  $d$ . The matrix is represented as a two dimensional array  $Q$  of size  $k$  rows by  $s$  columns, where  $k$  is the number of possible grey levels and  $s$  is the number of possible neighbours to a pixel in an image. The  $Q$  matrix is computed for a specific positive integer value of distance  $d$  and difference  $a'$ . For every pixel, the number of times that its grey value and a neighbouring pixel's grey value differ by  $a'$  or less are counted. The value at row  $i$  and column  $j$  in the  $Q$  matrix is incremented where  $i$  is the grey value of the pixel under examination, and  $j$  is the value itself.

This matrix relates to the texture information in an image or image region. It is invariant to image rotation as well as to linear grey level transformations or shifts of the grey level

spectrum. For example, if the brightness of an image were to be increased without resulting in any values being truncated, the neighbouring grey level dependence matrix would look exactly the same as in the darker.

The following five features or parameters were used to extract information from the  $Q$  matrix. They are the Small Number Emphasis (SNE), Large Number Emphasis (LNE), Number Non Uniformity (NNU), Second Moment (SM) and Entropy (ENT). Each of these scalars indicates some qualitative measure of the image.

$$\text{Small Number Emphasis : } SNE = \frac{\sum_{k=1}^k \sum_{s=1}^s [Q(k, s) / s^2]}{R} \quad (2.18)$$

The SNE can be seen as a measure of the fineness of the image.

$$\text{Large Number Emphasis : } LNE = \frac{\sum_{k=1}^k \sum_{s=1}^s [s^2 Q(k, s)]}{R} \quad (2.19)$$

The LNE can be seen as a measure of the coarseness of the image.

$$\text{Second Moment : } SM = \frac{\sum_{k=1}^k \sum_{s=1}^s [Q(k, s)]^2}{R} \quad (2.20)$$

The SM is related to the homogeneity of the image.

$$\text{Entropy : } \varepsilon_n = \frac{-\sum_{k=1}^k \sum_{s=1}^s [Q(k, s) \log[Q(k, s)]]}{R} \quad (2.21)$$

$$\text{Number Nonuniformity: } NNU = \frac{\sum_{s=1}^s \left[ \sum_{k=1}^k Q(k, s) \right]^2}{R} \quad (2.22)$$

The NNU and ENT are also related to the coarseness of the image.

$$\text{Normalizing Factor: } R = \sum_{k=1}^k \sum_{s=1}^s [Q(k, s)] \quad (2.23)$$

$R$  is defined as the normalising factor by Sun & Wee (1983).

The major advantages of using this approach for extracting texture features from an image are:

- Texture features can be easily computed.
- Texture features are essentially invariant under spatial rotation.
- Texture features are invariant under linear grey level transformation and can be made insensitive to monotonic grey level transformation. This means that two images, which have different grey tones, could have the same texture.
- The computation and storage memory requirements to process an image using the neighbouring grey level dependence matrix approach are less than that of the spatial grey level dependence matrix approach and the Power spectrum method approach (Sun & Wee, 1983). The reasons are as follows:
  - This approach is invariant under linear grey level transformation, it does not require histogram flattening of the original images as required by the spatial grey level dependence matrix approach and the Power spectrum method approach.
  - This approach is essentially angularly independent, unlike the spatial grey level dependence matrix approach, which needs to use the average and range of the four directional features. The computation requirement is less than that of a single directional feature.

- In the neighbouring grey level dependence matrix approach, the storage memory requirement is directly proportional to the size of the  $Q$  matrix.

Sun & Wee (1983) has reported that it is complicated to attach physical meaning to which specific texture characteristics are represented by each parameter defined in Equations 2.18-2.22. The neighbouring grey level dependence matrix approach is sensitive to the coarseness difference in images. Very few comparative data on the neighbouring grey level dependence matrix and spatial grey level dependence matrix methods are available in the literature. According to the literature (Siew, Hodgson & Wood, 1988; Van Gool, Dewaele & Oosterlinck, 1983) no comments on the relative performance of the neighbouring grey level dependence matrix and spatial grey level dependence matrix are given, except that they both give comparable classification with no clear advantage with one over the other. Moolman (1995) has shown that the neighbouring grey level dependence matrix performs slightly better than the spatial grey level dependence matrix in classification of images obtained from copper and platinum flotation plants.

## 2.2.4 BUBBLE SEGMENTATION VIA THE MORPHOLOGICAL WATERSHED ALGORITHM

The previous three approaches discussed namely the neighbouring grey level dependence matrix, spatial grey level dependence matrix and power spectrum method are image processing techniques which analyse froth images from a textural approach. The present approach using the watershed algorithm does not analyse froth images via a textural approach (indirect approach) but rather a segmentation approach (direct approach).

Sader-Kazemi & Cilliers (1997) first successfully used the morphological watershed algorithm for measuring flotation froth bubble size and shape distributions in 1997. Botha (1999) has used the watershed algorithm in the analysis of the surface froth of froth flotation systems which has led to a sensor platform, consisting of the algorithm software, graphical user interface and other functionality being developed. The algorithms Botha (1999) has developed are able to perform a complete static and dynamic analysis on digital images of froth surfaces.

Static analysis includes all parameters that can be determined from a single image and thus excludes any motion and other dynamic analysis. The static analysis includes a full bubble segmentation where the perimeter of each bubble is determined using the watershed morphology. The following sections briefly describe the different stages of the complete bubble segmentation of a froth flotation image via the watershed algorithm.

### 2.2.4.1 Marker Extraction

The image to be segmented has to be pre-processed in a certain way in order to prevent over segmentation by the watershed algorithm. For each bubble that is expected to be segmented out, precisely one collection of co-ordinates within the area of that bubble is required.

The sun or spotlights that are used on industrial camera set-ups allows each bubble to display a localised reflection. These localised reflections are always available as long as there is a single dominant light source. These localised reflections are used to extract bubble markers (Sader-Kazemi & Cilliers, 1997)

The  $h$ -dome extraction by Vincient (1993) was used by Botha (1999) to extract these localised reflections. The extractions were found to be very effective and robust on froth surface images taken under a number of varying conditions and lighting by Botha (1999). This transform is able to extract light structures, i.e. regions of pixels, which are brighter than their immediate surroundings, without any size or shape. An  $h$ -dome  $D$  of image  $I$  is defined as a connected component of pixels such that every pixel  $p$  that is a neighbour of  $D$  satisfies:

$$I(p) < \min \{I(q) \mid q \in D\} \text{ and} \quad (2.24)$$

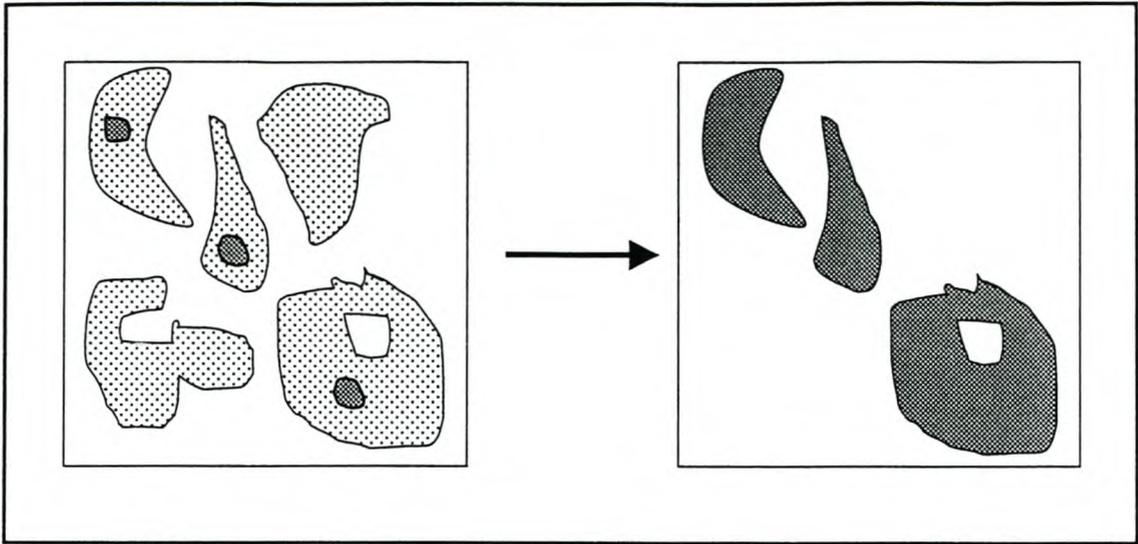
$$\max\{I(q) \mid q \in D\} - \min\{I(q) \mid q \in D\} < h \quad (2.25)$$

i.e. all the neighbouring pixels of the  $h$ -dome, or light structure, are less bright than the light structure itself, and the maximum intensity range of the pixels in the light structure is less than  $h$  (Botha, 1999). The value of a pixel  $p$  in the  $h$ -dome  $D$  is given by:

$$I(p) - \min\{I(q) \mid q \in D\} \quad (2.26)$$

i.e. the pixel values are offset at the minimum intensity value of the  $h$ -dome.

The  $h$ -dome extraction relies on a process called grey-scale reconstruction. Grey-scale reconstruction is based on binary reconstruction. Binary reconstruction  $M_I$  of mask image  $I$  from marker image  $J$  is the union of all the connected components of  $I$  which contain at least a pixel of  $J$ .  $I$  and  $J$  in this case are binary images which satisfy the condition  $J \subseteq I$ . Figure 2.1 illustrates binary reconstruction.



*Figure 2.1: Illustration of binary reconstruction. In the left image, the dark grey regions represents the markers and the lighter grey regions the mask. Only connected components of the mask, which contain at least a pixel of a marker, are kept and the rest are eliminated. The result is shown in the right-hand image.*

#### 2.2.4.2 Grey-Scale Reconstruction

Let  $I$  and  $J$  be grey-scale images both defined on the domain  $D_I$ , taking values in the discrete set  $[0, N-1]$  and satisfying  $J(p) < I(p)$  with  $p \in D_I$ . Then the grey-scale reconstruction  $M_I(J)$  of mask image  $I$  from marker image  $J$  is defined as:

$$M_I(J)(p) = \max\{k \in [0, N-1] \mid p \in M_{T_k(I)}(T_k(J))\} \quad (2.27)$$

$$\text{Where } T_k(I) = \{p \in D_I \mid I(p) \geq k\} \quad (2.28)$$

Is the binary threshold image of  $I$  at value  $k$  and so  $M_{T_k(I)}(T_k(J))$  is the binary reconstruction of mask image binary image  $T_k(I)$  from binary marker image  $T_k(J)$ . In other words, for each pixel  $p$ , the grey-scale reconstruction is the maximum intensity value  $k$  at which a contiguous region of the binary threshold image of  $J$ , where both binary images have been thresholded at intensity  $k$  (Botha, 1999).

The  $h$ -dome extraction works by subtracting a constant,  $h$ , from all pixel intensities in an image  $I$ , thus yielding image  $I-h$ .  $I$  is the grey-scale reconstructed from  $I-h$  and the resulting reconstruction is subtracted from the original image  $I$ . This yields the  $h$ -dome extraction of image  $I$ . Botha (1999) found the  $h$ -dome extraction to be very effective and robust on froth surface images taken under a number of different conditions and lighting. He found that for most froth images  $h=50$  yielded excellent results.

### 2.2.4.3 Homotopic Modification

The homotopic modification by Vincent & Soille (1991) uses the extracted markers and modify the image that is to be analysed in such a way that:

- The image's only minima are located on the extracted markers.
- All the other minima in the image are suppressed.
- The most significant intensity divisions between marker positions are preserved.

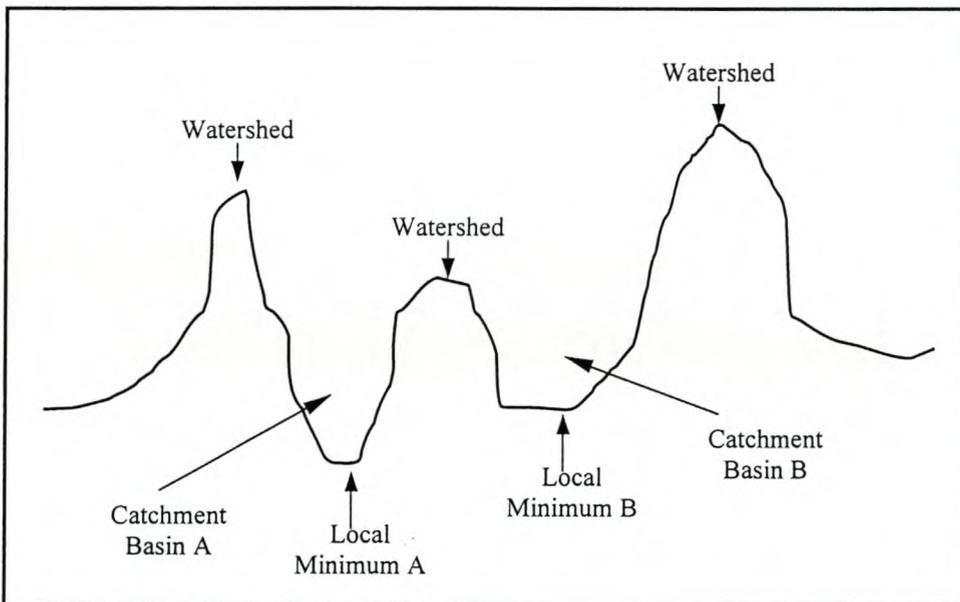
These changes prevent the watershed algorithm from over segmenting the image that is to be analysed. In order to perform this modification the dual grey-scale reconstruction  $M_I^*(J)$  of mask  $I$  from marker  $J$  is required. The homotopic modification  $H(I)$  of image  $I$  using binary marker image  $Z$  is defined as:

$$H(I) = M_{\min(I, \max(I) * M)}^* (\max(I) * Z) \quad (2.29)$$

Where the marker image  $Z$  has zeros where minima are to be enforced and ones otherwise. Seeing the difference between a modified and unmodified with the naked eye is very difficult.

#### 2.2.4.4 Watershed Segmentation

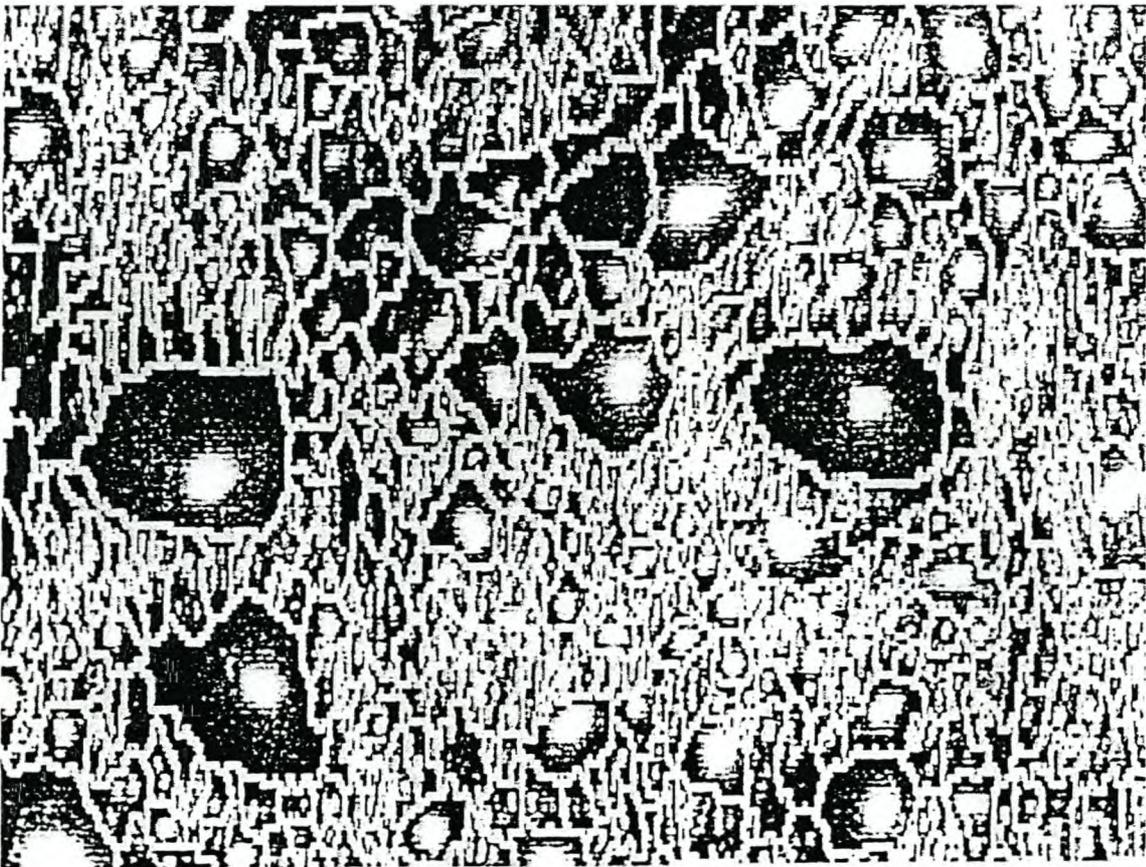
The greyscale watershed by Dougherty (1992) is a morphological segmentation algorithm and is the cornerstone of Botha's (1999) bubble segmentation approach towards analysis of froth flotation images. Any greyscale image could be visualised as a three dimensional landscape where a pixel intensity represents the altitude of the landscape at that Cartesian position (Botha, 1999). This landscape consists of valleys and hills, representing the darker and the brighter regions of an image respectively. If a drop of water were to fall at any position on the landscape, it would flow down and come to rest in the deepest part of the closest valley. This deepest part represents a local minimum in the image. The whole area surrounding it, from whence a drop would flow down into that specific local minimum, is the catchment basin of the local minimum. The borders that separate catchment basins belonging to different local minima are called watershed lines or just watersheds Botha (1999). Figure 2.2 illustrates the terms just defined.



*Figure 2.2: Watershed terminology*

The aim of the greyscale watershed is to extract watershed lines, i.e. lines of high intensity pixels separating local intensity minima in the image. Each bubble in an image of a flotation froth surface needs to be segmented out. The bubbles in the image are structurally an irregular dome. The boundaries where bubbles meet gives the watershed lines. In other words, the bubbles in a froth image could easily be segmented from one another.

Botha (1999) has implemented the highly optimised watershed algorithm from Vincent & Soille (1991). Previous to this specific implementation of the watershed, it was not possible to perform such a segmentation in near real-time. Figure 2.3 shows an image that was first homotopically modified and then the watershed was applied to the image. Note how well the watershed lines (indicated in white) follow the bubble boundaries (Botha, 1999).



*Figure 2.3: Watershed segmentation of flotation froth*

In cases where fine froth areas are present, the watershed segmentation has difficulty in segmenting the small bubbles in the fine froth areas. The watershed segmentation will segment these fine froth areas as large bubbles. This is due to the scarcity of the occurrence of relatively bright-localised reflections in fine froth and this results in a small number of markers being extracted for relatively large areas of fine froth. Another drawback of this technique is that it is cumbersome and takes time to run the algorithm.

## 2.3 SUMMARY

Chapter 2 begins with a brief introduction on why there is a need for the development of an automatic control system, which uses on-line machine vision in the flotation industry. This is followed by a discussion on the different approaches taken by researchers in performing image analysis of froth flotation surfaces. A number of researchers have taken different approaches in solving the problem.

The most common method used to solve the problem has been a direct segmentation approach. This approach has led to two commercial systems been developed for the on-line monitoring of industrial flotation plants. The one problem with the two commercial systems that have been developed is that they cannot analyse fine froths effectively due to the algorithms employed by them. Most techniques reviewed are developed specifically towards the image analysis of a particular froth such as copper, platinum or coal. It is therefore difficult to determine from the literature how robust the different techniques are with regard to the type of mineral in the froth.

Three different textural approaches towards image analysis of froth systems have been discussed. The advantage of the power spectrum method is that it is able to analyse the structure of the surface froth in terms of the regular structure of the froth, which can be effectively described by a frequency domain representation. The features extracted from the power spectrum method, spatial grey level dependence matrix and neighbouring grey level dependence matrix can all be used in the characterisation of the surface froth of

flotation cells. The grey level dependence matrices have the advantage over the power spectrum method in that features can be extracted more efficiently than in the case of the fast Fourier transform.

The features generated by the neighbouring grey level dependence matrix and the spatial grey level dependence matrix can easily be used for the classification of different froth structures (Moolman, 1995). The neighbouring grey level dependence matrix is computationally more efficient than the spatial grey level dependence matrix because of the smaller feature space involved.

Bubble Segmentation via the watershed algorithm is a direct approach towards image analysis of froth systems. The approach consists of four steps and has been effectively used in the development of a commercial software package for the analysis of froth systems. The limitation of the watershed algorithm is that it cannot segment fine froths where the bubbles are very small and that the algorithm is cumbersome and computational demanding.

Chapter 3 discusses the methodology of linear and non-linear principal component analysis for the extraction of features from froth images.

## CHAPTER 3

### PRINCIPAL COMPONENT ANALYSIS

Neural networks are widely recognised for their ability to interpret pattern based information and have attracted particular interest in the process engineering community as a framework for the representation of complex processes. In this chapter the techniques of using neural networks to extract information from digital images of froth flotation surfaces is presented.

An automatic froth monitoring system will generate large amounts of data in the form of images of the froth phase. An autoassociative neural network can be used to significantly decrease the storage space required for images by compression of the froth images. The driving force for this research is the analogy that connectionist computer vision systems have with biological vision as discussed by Moolman (1995).

The problem of dimensionality reduction is closely related to feature extraction. Feature extraction refers to identifying the salient aspects or properties of data to facilitate its use in subsequent task. Its features are a set of derived variables, which efficiently capture the information contained in the original data.

There are two different ways in which the dimensionality of a data set might be reduced. The first is to simply remove those variables, which carry the least predictive power. A

less destructive method involves projecting a high dimensional data set onto a lower dimensional space. This involves combining the set of variables into a smaller number of new variables, which describe as much of the data as possible. A common method for reducing the number of variables used in a data set involves taking PCs of the original data set.

LPCA is an objective technique used to detect and characterise optimal lower-dimensional linear structure in a multivariate data set (Monahan, 2000). Its applications include reduction of data dimensionality for data interpretation and for forecasting. LPCA can be regarded as a single example of a general class of feature extraction methods, which attempt to characterise lower-dimensional structure in large multivariate data sets.

### **3.1 CONVENTIONAL LINEAR PRINCIPAL COMPONENT ANALYSIS**

PCA is a technique for mapping multidimensional data into lower dimensions with minimal loss of information. LPCA linearly transforms a set of variables into a substantially smaller set of uncorrelated variables containing most of the information of the original set of variables. A small set of uncorrelated variables is easier to understand and work with than a large set of correlated variables (Aldrich, 1998-1999).

LPCA is a linear technique for multivariate data analysis in the sense that it uses linear functions to model relationships between variables. LPCA is used to identify and remove correlations among variables as an aid to dimensionality reduction, visualisation, and exploratory data analysis. Of all the techniques used to explore multivariate data, LPCA is probably one of the most popular. Within the class of linear methods, the optimal information preserving transformation is given by LPCA (Fukunaga & Koontz, 1970). Reduction of dimensionality by LPCA has been shown to facilitate many types of multivariate analysis, including data validation and fault detection, quality control, correlation and prediction, process monitoring and data visualisation (Kramer, 1991).

The feature variables in LPCA, are linear combinations of the original problem variables. The coefficients of the linear transformation are such that if the feature transformation is applied to the data set and then reversed, there will be a known minimum sum of squares difference between the original and reconstructed data.

Let  $X = (x_1, x_2, \dots, x_m)$  be an  $m$ -dimensional data set. The first linear principal component is defined to be the linear combination of the columns of  $X$ , which describes the greatest amount of variability

$$t_1 = p_1^T * X \quad (3.1)$$

subject to  $\|p_1\| = 1$  in the original data set. The second linear principal component is defined to be orthogonal to the first and is the linear combination defined by

$$t_2 = p_2^T * E_1 \quad (3.2)$$

$$\text{where } E_1 = (X - t_1 * p_1^T) \quad (3.3)$$

which describes the next greatest variance subject to  $\|p_2\| = 1$ . The second linear principal component therefore explains the greatest amount of variability in the residual data ( $E_1$ ). This procedure is essentially repeated until  $m$  linear principal components are calculated. In effect LPCA decomposes the observation matrix,  $X$ , as

$$X = T * P^T = \sum_i^m t_i * p_i^T \quad (3.4)$$

where  $p_i$  is the eigenvector of the covariance matrix of  $X$ . the matrix  $P$  is defined as the linear principal component loading matrix and  $T$  is defined as the matrix of linear principal component scores. The loadings provide information as to which variables

contribute the most to individual linear principal components. They are the coefficients on the principal components model.

The main attribute of LPCA which enables the dimensionality of the problem to be reduced, is that if some of the variables are collinear, then  $k$  linear principal components ( $k < m$ ) will explain the majority of the variability of the data, i.e. a smaller number of linear principal components than original variables are required are required to explain the variability in the data. The number of linear principal components that provide an adequate description of the data can be assessed using a number of techniques. Typically, cross-validation is employed. One of the features of LPCA is that the less important linear principal components often describe noise in the data.

LPCA depends critically on the scales used to measure the variables. If there are large differences between these variables, then those variables whose magnitudes are large will tend to dominate the first few linear principal components. Thus the results of LPCA vary with scales used for different variables. The lack of scale invariance implies that care needs to be taken when scaling the data. Different scaling routines can produce different results. A commonly used scaling technique is the so-called “auto-scaling” where each variable is scaled to zero mean and unit variance (Zhang, Martin, & Morris, 1997).

By construction, LPCA finds the lower-dimensional hyperplane that optimally characterises the data, such that the sum of squares of orthogonal deviations of the data points from the hyperplane is minimised. If the structure of the data is inherently linear then LPCA is an optimal feature extraction algorithm; however, if the data contains non-linear lower-dimensional structure, it will not be detectable by LPCA. Morris (1994) has pointed out that LPCA has the disadvantage in that it is a linear reduction technique and so can destroy any non-linear structure in the data that is trying to be extracted. To overcome this problem Kramer (1991) introduced a neural network based generalisation of LPCA to overcome the problem of non-linear feature extraction. This technique was referred to as NLPCA. Another solution to the non-linear feature extraction problem was put forward independently by Hastie & Stuetzle (1989), who named their method

principle curves and surfaces. NLPCA and principle curves and surfaces are closely related, and are, for a broad class of situations, essentially the same.

## 3.2 NON-LINEAR PRINCIPAL COMPONENT ANALYSIS

Conventional LPCA is a linear technique, which reduces data dimensionality by exploiting linear correlations between observed variables. If there exists non-linear correlation's between observed variables then a non-linear method will describe the data with greater accuracy and by fewer factors than a linear method (Tan & Mavrovouniotis, 1995). Linear dimensionality reduction techniques such as conventional linear principal component analysis, factor analysis, independent component analysis and projection pursuit have been well studied in the past (Chang & Ghosh, 2001). Linear techniques are attractive for their simplicity and amenability to analysis, but may be inadequate for modelling highly non-linear data. To overcome this problem researchers have developed non-linear methods such as generalised linear models, AANN, self-organising maps, and principal surfaces for dimensionality reduction.

The main difference between conventional LPCA and NLPCA is that the latter involves non-linear mappings between the original and reduced dimension space. If non-linear correlation's between variables exist, NLPCA will describe the data with greater accuracy and by fewer factors than conventional LPCA, provided that there are sufficient data to support the formulation of more complex mapping functions. While conventional LPCA identifies only linear correlations between variables, NLPCA uncovers both linear and non-linear correlations, without restriction on the character of the non-linearity's present in the data (Kramer, 1991).

Let  $Y$  represent a  $n*m$  table of data ( $n$ =number of observations,  $m$ =number of variables). LPCA is an optimal factorisation of  $Y$  into two matrices,  $T$  called the scores matrix ( $n*f$ ), and  $P$ , called the principal components loadings matrix ( $m*f$ ), plus a matrix of residuals  $E$  ( $n*m$ ):

$$Y = T*P^T + E \quad (3.5)$$

Where  $f$  is the number of factors or principal components ( $f < m$ ). In NLPCA, the mapping into feature space is generalised to allow arbitrary non-linear functionalities. A mapping in the form:

$$T = G(Y) \quad (3.5)$$

is sort where  $G$  is a non-linear vector function, composed of  $f$  individual non-linear functions;  $G = \{G_1, G_2, \dots, G_f\}$ , analogous to the columns of  $P$ , such that if  $T_i$  represents the  $i$ th column of  $T$ ,

$$T_i = G_i(Y) \quad (3.6)$$

By analogy to LPCA,  $G_1$  is referred to as the primary non-linear factor or non-linear principal component, and  $G_i$  is the  $i$ th non-linear factor or non-linear principal component of  $Y$ . The inverse transformation, restoring the original dimensionality of the data, analogous to Equation 3.4, is implemented by a second non-linear vector function

$$H = \{H_1, H_2, \dots, H_m\}:$$

$$\hat{Y} = H(T) \quad (3.7)$$

where  $\hat{Y}$  is the recreated  $Y$ . The loss of information is again measured by  $E = Y - \hat{Y}$ , and the functions  $G$  and  $H$  are selected to minimise  $\|E\|$ .  $G$  and  $H$  are referred to as the mapping and demapping functions respectively.

To generate  $G$  and  $H$ , a basis function approach is used. Cybenko (1989) has shown that functions of the following form are capable of fitting any non-linear function  $v = f(u)$  to an arbitrary degree of precision:

$$v_k = \sum_{j=1}^{N_2} w_{jk2} * \sigma \left( \sum_{i=1}^{N_1} w_{ij1} * u_i + \theta_{j1} \right) \quad (3.8)$$

where  $\sigma(x)$  is any continuous and monotonically increasing function with  $\sigma(x) \rightarrow 1$  as  $x \rightarrow \infty$  and  $\sigma(x) \rightarrow 0$  as  $x \rightarrow -\infty$ . A suitable function is the sigmoid:

$$\sigma(x) = \frac{1}{(1 + e^{-x})} \quad (3.9)$$

In this work we have opted to use AANNs to perform NLPCA. The reason for choosing this technique over other available techniques such as generalised linear models, self organising maps and principal surfaces for non-linear dimensionality reduction is due to its ease of implementation and its success in the field of image compression which has been shown by Demers & Cottrell (1993).

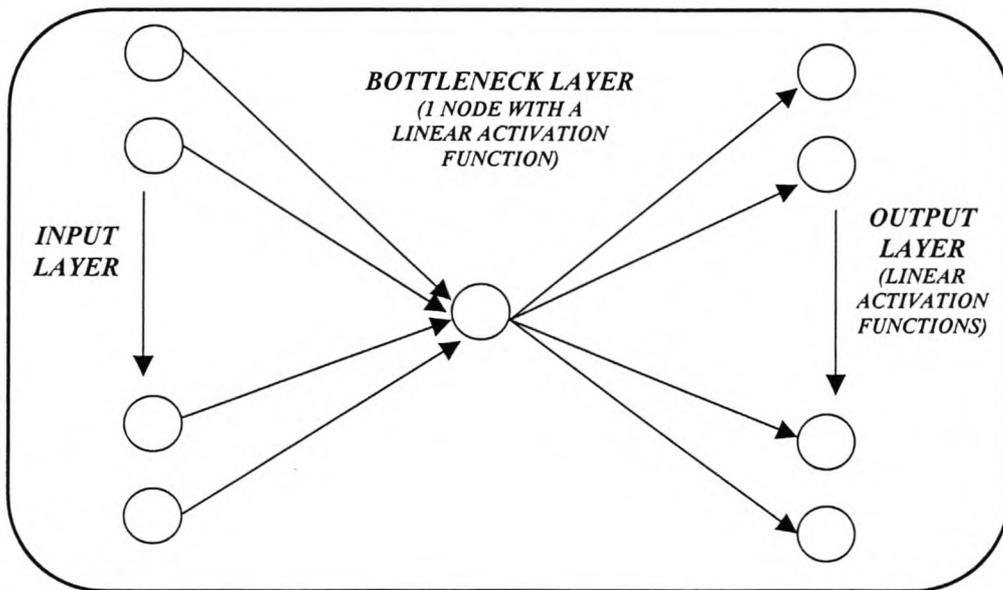
### 3.3 PRINCIPAL COMPONENT ANALYSIS USING AUTOASSOCIATIVE NEURAL NETWORKS

The AANN is a solution to the concern raised by Morris (1994) in Section 3.1 and an easy to use non-linear as well as linear projection method. In the 1990s AANNs were used as an effective data compression tool. AANNs are feedforward neural networks trained to perform the identity mapping between network input and output. Dimensionality reduction is achieved through a bottleneck layer, that is, a hidden layer with a smaller number of nodes than the input layer. Most previous work focussed on AANN architectures with a single hidden layer. Kramer (1991) pointed out that an AANN with a single hidden layer was unable to model non-linear relations between observed variables and latent variables, and consequently offered no improvement over conventional LPCA. Kramer (1991) then established a three hidden layer AANN architecture to capture non-linear correlations.

Kramers NLPCA and related methods have been successfully applied to many different problems. Tan & Mavrovouniotis (1995) uses a different version of Kramers (1991) NLPCA for data compression in which he reduces the dimensionality of data through optimising a three layer neural networks inputs. Dong & Mcavoy (1994) shows how another version of Kramers (1991) NLPCA can be used to construct control charts. Kramers (1991) NLPCA has been applied to the problem of image compression and non-linear feature extraction by De Mers & Cottrell (1993). Various other versions of Kramers (1991) NLPCA technique has been developed in recent years and has successfully been applied to many different problems. Before presenting the NLPCA method using AANNs, it would be appropriate to illustrate how LPCA can be performed using an AANN.

### 3.3.1 LINEAR PRINCIPAL COMPONENT ANALYSIS USING AN AUTOASSOCIATIVE NEURAL NETWORK

Linear AANNs have been shown by Bourland & Kamp (1988) to extract a linear combination of the principal components. A linear AANN has three layers with  $q$  nodes in the input and output layers and one node in the hidden layer as can be seen in Figure 3.1. The activation functions used in the hidden and output layers are linear.



*Figure 3.1: Autoassociative neural network architecture with a single hidden layer which has a linear activation function*

The network output is trained using the backpropagation algorithm to mimic the input vector. Once trained, the hidden layer node will produce a reduced representation ( $x$ ) corresponding to each input  $y$ .

This AANN was designed in the neural network toolbox of Matlab<sup>3</sup>. A normal feedforward network (newff<sup>4</sup>) in the neural network toolbox of Matlab could not be used to design this network. This is due to the function newff been only able to give the output of the output layer whereas for this study the output of the output layer as well as the output of the hidden layer (bottleneck layer) was required. The output of the hidden layer gives the linear factor or linear principal component, which is of importance to this study. A custom AANN had to be designed in the neural network toolbox of Matlab.

The activation function<sup>5</sup> in the single node of the hidden layer as well as the nodes in the output layer were linear (Purelin<sup>6</sup> activation function). The number of inputs into the network was one (i.e. the input into the AANN was a single matrix of size  $m*q$  where  $m$  is the number of observations and  $q$  is the number of variables in the matrix). The network contained two bias connections. Bias one was connected to the single node in the hidden layer and bias two was connected to the nodes of the output layer. The input was connected to the bottleneck layer and the bottleneck layer was connected to the output layer. The number of nodes in the output layer is equal to the number of variables in the single input matrix which is  $q$ . The target matrix during training of the network is equal to the input matrix. Refer to Figure 3.1 to view the network architecture. Two outputs were captured from the network. The first output captured was from the single hidden layer and the second output captured was from the second layer (which is the output layer).

It is important to understand that this AANN architecture does not perform NLPCA but rather LPCA, which is similar to conventional LPCA. This is due to the single node in the hidden layer having a linear activation function.

<sup>3</sup> Matlab is a software package used by engineers. It has a number of toolboxes for different tasks.

<sup>4</sup> Is a function in the neural network toolbox of Matlab which allows a feedforward network to be designed

<sup>5</sup> Activation function is a mathematical function which takes the weighted activation values coming into a node, sums them, and translates the result to a position along a given scale.

<sup>6</sup> Is the name given to the linear activation function in Matlab.

### 3.3.2 NON-LINEAR PRINCIPAL COMPONENT ANALYSIS USING AN AUTOASSOCIATIVE NEURAL NETWORK

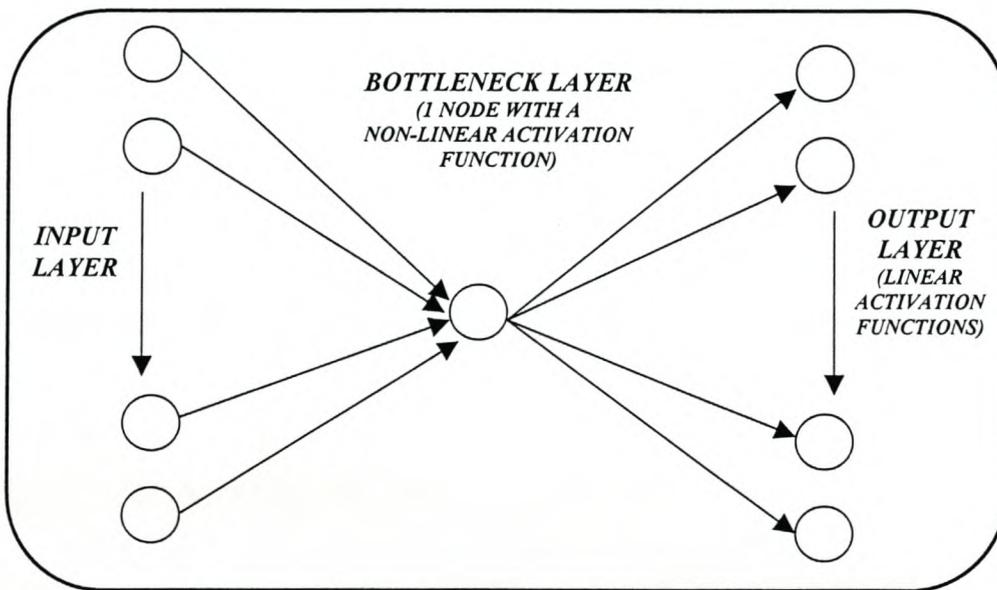
The NLPCA method of Kramer (1991) uses AANNs training procedures to generate non-linear features. Dimensionality reduction is performed by AANNs, which are feedforward neural networks trained by backpropagation. Dimensionality reduction occurs through a bottleneck, that is, a hidden layer with a smaller number of nodes than the input and output layers. The nodes of the hidden layers contains non-linear transfer functions which generate non-linear features.

The AANN is trained to perform an identity mapping, where the input is approximated at the output layer. Since there are fewer nodes in the bottleneck layer than in the input or output layer, the bottleneck nodes must encode the information in the inputs for the subsequent layers to reconstruct the input. If network training finds an acceptable solution, then a good representation of the input must exist in the bottleneck layer. This implies that data compression caused by the bottleneck layer may force hidden nodes to represent significant features in the data.

For any task where the hidden layer is required to extract features from the inputs in order to generalise or reduce the dimensionality of a data set, it is essential for the network to fan in to the bottleneck layer (i.e. the bottleneck layer should contain fewer nodes than the input layer). NLPCA is a direct generalisation of the AANN implementation of LPCA as discussed in Section 3.3.1. Two types of AANNs can be used to perform NLPCA. The first type of AANN uses a single hidden layer while the second type uses three hidden layers. These two types of AANNs will be discussed in more detail in the following sections.

### 3.3.2.1 Autoassociative Neural Network -1 Hidden Layer (Non-Linear Activation Function)

This AANN is identical to the one used in Section 3.3.1 with one major difference. That been that the activation function used in the single node of the hidden layer is a sigmoidal<sup>7</sup> activation function as can be seen in Figure 3.2. This AANN can represent any continuous function under weak assumptions (Cybenko, 1989). This AANN was designed exactly the same as that in Section 3.3.1.



*Figure 3.2: Autoassociative neural network architecture with a single hidden layer which has a non-linear activation function*

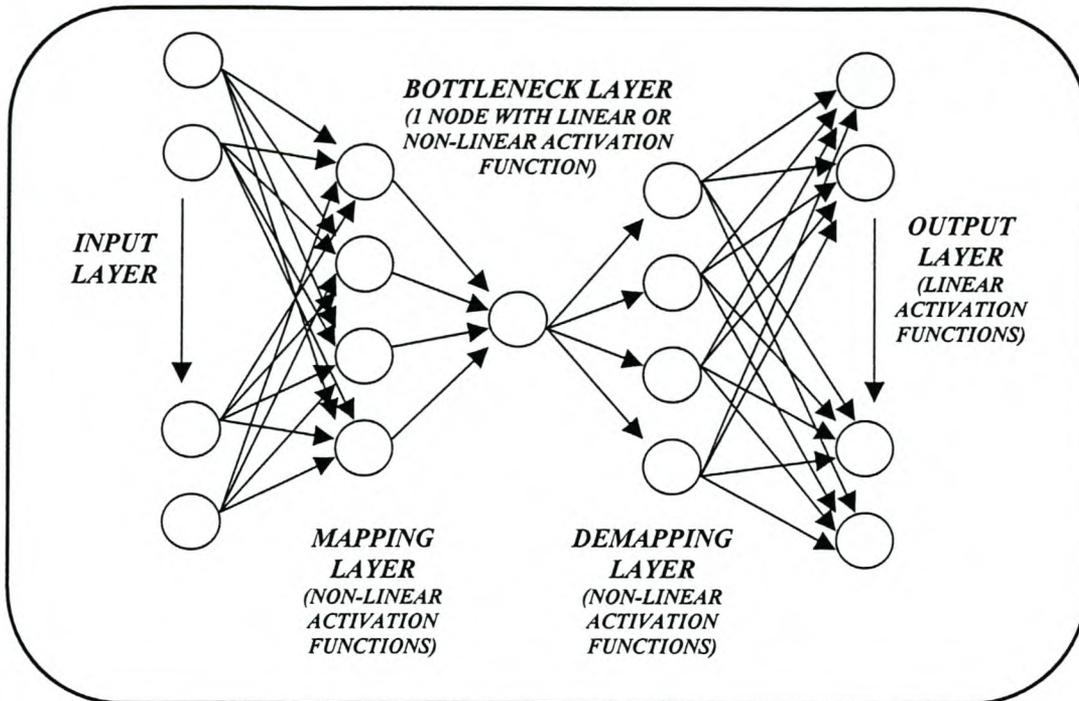
<sup>7</sup> Logistic and tanh functions are called sigmoidal transfer functions and they are non-linear transfer functions.

### 3.3.2.2 Autoassociative Neural Network-3 Hidden Layers (Non-Linear Activation Function)

NLPCA with an AANN with three hidden layers modifies the AANNs with single hidden layers by adding hidden layers with non-linear activation functions between the input and bottleneck layers and between the bottleneck layer and the output layers giving a network with a total of five layers as can be seen in Figure 3.3. The five-layer AANN has  $p$  nodes in the input layer, one or more nodes<sup>8</sup> in the third layer (Bottleneck Layer) and  $p$  nodes in the output layer. The nodes in layers two and four must have non-linear activation functions so that layers one, two and three and layers three, four and five can represent arbitrary smooth functions. The nodes in layer three and five could either have a linear or non-linear activation functions. Direct connections are allowed between layers one and three and between three and five, but direct connections are not allowed to cross the bottleneck (layer three). As with LPCA networks data compression takes place because the  $p$ -dimensional inputs must pass through the  $k$ -dimensional bottleneck layer<sup>9</sup> before reproducing the inputs. Once the AANN has been trained, the bottleneck node activation function gives the scores.

<sup>8</sup> The bottleneck layer could have any number of nodes as long as they are less than  $p$

<sup>9</sup> For compression to take place  $k < p$ . It is important to note that the bottleneck layer could have one node or  $k$  nodes as long as  $k$  is less than  $p$ .



*Figure 3.3: Autoassociative neural network architecture with three hidden layers, which have non-linear activation functions, and an output layer with linear activation functions.*

It can be seen that AANNs are able to perform both LPCA as well as NLPCA depending on whether the activation functions used in the nodes of the hidden layers are linear or non-linear. Cottrell, Munroe & Zisper (1987) have used three layer AANNs to perform image compression.

### **3.4 TWO APPROACHES FOR PERFORMING PRINCIPAL COMPONENT ANALYSIS USING AUTOASSOCIATIVE NEURAL NETWORKS**

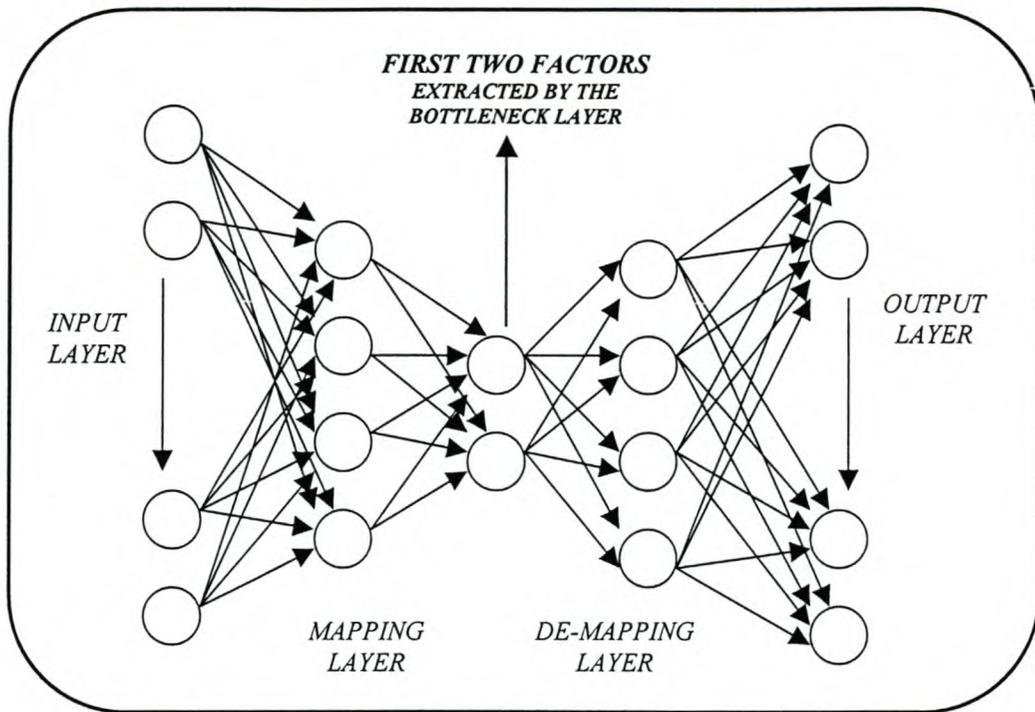
Two different approaches exist using AANN to perform LPCA and NLPCA, they are simultaneous or sequential obtaining of the linear and non-linear factors using an AANN. The simultaneous approach works by training a single AANN to extract all factors, while the sequential approach works by training as many AANNs as factors to be extracted.

#### **3.4.1 SIMULTANEOUS PRINCIPAL COMPONENT ANALYSIS USING AUTOASSOCIATIVE NEURAL NETWORKS**

The simultaneous approach works by pre specifying the number of factors<sup>10</sup> required to be extracted from a data set. Once the number of factors to be extracted from the data set are known then the bottleneck layer of the AANN must be provided with as many nodes as the number of factors to be extracted (Refer to Figure 3.4). Thus the number of factors required are extracted from training a single AANN. Figure 3.4 shows an AANN in which two factors are extracted by the bottleneck layer. If three factors were to be extracted, then the bottleneck layer in Figure 3.4 would require three nodes.

---

<sup>10</sup> Factors, features and principal components have the same meaning in this thesis and are used interchangeably.



**Figure 3.4: Autoassociative neural network architecture for simultaneous linear and non-linear principal component analysis**

Figure 3.4 illustrates simultaneous principal component extraction using a five layer AANN. In a similar manner, simultaneous principal component<sup>11</sup> extraction can be performed using a three layer AANN. If three principal components were required to be extracted simultaneously from a three layer AANN then the single hidden layer would require three nodes. Thus the number of principal components to be extracted determines the number of nodes in the single hidden layer. The drawback of using this method is that it is difficult to determine the amount of variance captured by each extracted principal component.

<sup>11</sup> Principal component here refers to either linear or non-linear principal components depending on the type of activation functions used in the nodes of the single hidden layer of a three layer AANN.

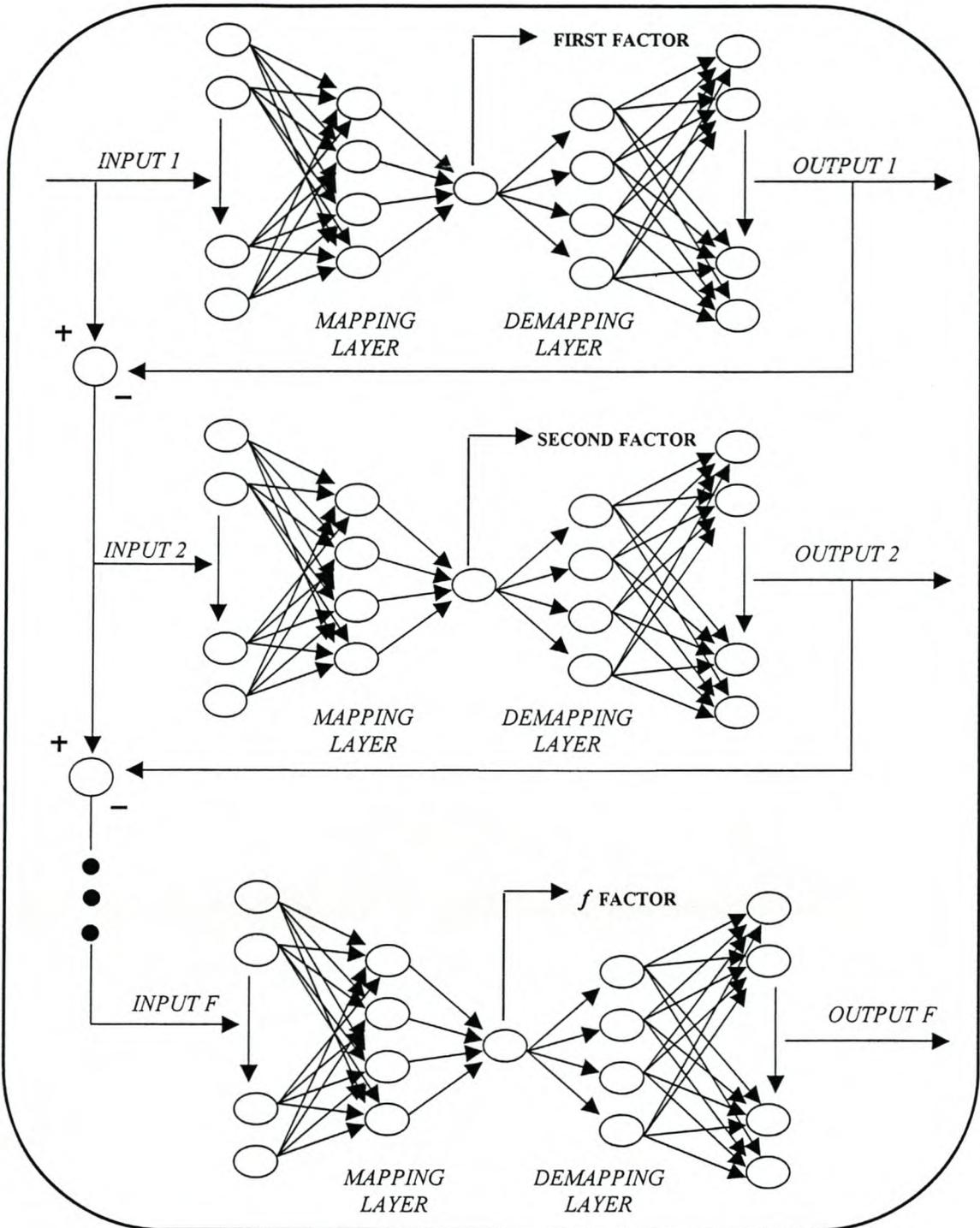
### 3.4.2 SEQUENTIAL PRINCIPAL COMPONENT ANALYSIS USING AUTOASSOCIATIVE NEURAL NETWORKS

The sequential method of LPCA and NLPCA via an AANN is similar in spirit to the recursive procedure of linear principal component calculation often used in conventional LPCA. This method also requires the number of factors to be extracted to be known as in the simultaneous method. Let's say  $f$  non-linear factors are desired. Then a series of  $f$  separate AANNs, each containing a single node in the bottleneck layer, are trained. Only the primary factor of the input matrix is extracted at each stage. The residual from the previous network becomes the input (and target output) for the succeeding network, and the training proceeds serially (Refer to Figure 3.5). It is important to remember that each of the  $f$  AANNs must have the same architecture, that is they could either be a three or five layer AANN. The reason for this is so that the features extracted by the network all have the same characteristics. If one factor is extracted by a five layer network and another by a three layer network then the factors being mapped are not of the same standard due to the difference in network architecture. In a similar manner a three layer AANN can be used to perform sequential PCA. The sequential version of LPCA and NLPCA has a few potential advantages over the simultaneous version of LPCA and NLPCA using AANNs. These advantages are:

- Sequential training procedures allows rescaling of the residual matrix between steps, which may improve convergence.
- The sequential training forces each bottleneck node to model a separate factor in the data.
- The amount of variance each factor accounts for can directly be calculated.

One of the difficulties in using the sequential method is that the number of factors to be extracted is the number of AANNs that need to be trained whereas with the simultaneous method only one AANN has to be trained irrespective of the number of factors to be extract. In light of the above the *sequential method of factor extraction* using an AANN was opted for to perform LPCA and NLPCA in this study due to the importance of being

able to directly calculate the amount of variance captured by each factor which is of critical importance in the following chapters.



*Figure 3.5: Neural network architecture for sequential linear and non-linear principal component analysis using an autoassociative neural network*

### 3.5 METHODOLOGY FOR TRAINING AN AUTOASSOCIATIVE NEURAL NETWORK USING THE SEQUENTIAL METHOD

The following methodology was used to train and test the AANN models for the extraction of the first three principal components. This methodology applies to both a three and five layer AANN irrespective if linear or non-linear principal components are been extracted. . The following discussion is related to Figure 3.5.

Let the input matrix to AANN 1 be  $P (m * q)$ .

Let the target matrix of AANN 1 be  $T (T = P)$ .

Then AANN 1 is trained with  $P$  as input and  $T$  as the target. AANN 1 yields the first principal component and *Output 1*. The residual matrix  $R1$  is the input to AANN 2 is given by:

$$R1 = P - \text{Output 1}$$

AANN 2 is then trained with  $R1$  as the input and  $R1$  as the target. AANN 2 yields the second principal component and *Output 2*. The residual matrix  $R2$  is the input into to AANN 3 is given by:

$$R2 = R1 - \text{Output 2}$$

AANN 3 is then trained with  $R2$  as the input and  $R2$  as the target. AANN 3 yields the third principal component. In a similar manner any number of principal components can be extracted.

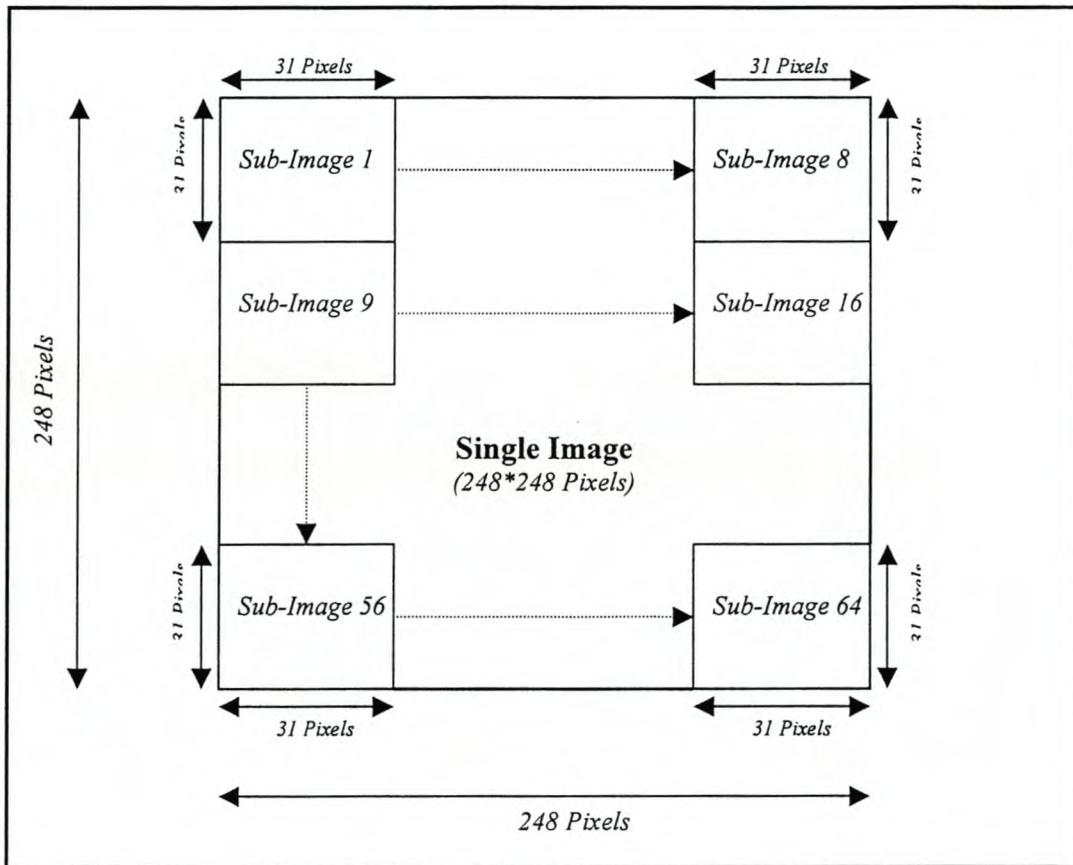
### 3.6 PRE-PROCESSING OF IMAGES TO BE USED AS INPUTS TO AN AUTOASSOCIATIVE NEURAL NETWORK

Images of the surface froth of a flotation process under differing conditions were used as the inputs into the AANN. The images were taken at different intervals during a flotation process. A digital camera was used to capture these images. The images were captured in a bitmap format (bmp). The sizes of all the images were 248 x 248 pixels. Therefore the total number of pixels in a single image was 61504 (248\*248 pixels). Initially each pixel in an image was considered to be a variable, thus the total number of variables obtained in a single image was 61504. Each of these 61504 variables were then to be used as inputs into an AANN. This was initially tried but such a large number of inputs to the AANN made it extremely difficult to train the network. Therefore, an alternative approach had to be followed. The approach followed was very similar to the one followed by Moolman (1995). It was based on an image being sampled via windows of dimension  $k*k$  pixels. The simplest way in which to verify that the correct window size has been chosen is to perform a visual inspection of the reconstructed images at different window sizes. If it is found that at specific window size the quality of the reconstructed image by the AANN is good, then the window size is incrementally increased until a point is reached where the reconstructed image is of a poorer quality. Sophisticated techniques to quantify the quality of the reconstructed image are not available and any method is subjective to a certain extent.

A simple technique that is often used, is to compute the correlation coefficient between the original image and the reconstructed image. In this way a value can be attached to the subjective result of a visual inspection of the quality of an image by setting a lower limit to the correlation coefficient of an image that is just acceptable in appearance. The value of the lower limit of the correlation coefficient was obtained by taking a number of different size images (Minimum size 4\*4 pixels and increased by 4\*4 pixels to a maximum size of 44\*44 pixels) and simulating them through an AANN. The correlation coefficient was computed for each reconstructed image at different window sizes. The

correlation coefficient together with the visual inspection of the reconstructed image led to the selection of the appropriate window size. It was found that an acceptable appearance of a reconstructed image could be obtained at a window size of  $31 \times 31$  pixels. Any window size larger than that led to a poor reconstruction of the original image.

The input layer size was governed by the quality of the reconstructed image. It was found that a neural network with an input layer of approximately 961 nodes (961 variables) would work well although it took much longer to train than images with a smaller window size. Thus a single image of size  $248 \times 248$  pixels is sampled into blocks of dimensions of  $31 \times 31$  pixels to give 64 sub-images (Refer to Figure 3.6 for a graphical description of the segmentation of a single image). Each image ( $248 \times 248$ ) was split into 64 sub-images using a software package called Blimp, which is a graphics package.



**Figure 3.6: Segmentation of a single image ( $248 \times 248$  Pixels) into 64 sub-images ( $31 \times 31$  pixels)**

### 3.6.1 VECTOR FORMATION

Vector formation is the process by which pixels of an image are regrouped in vectors using a certain criterion. There are two important steps in the vector formation process:

- The vector formation strategy and
- The choice of the vector size

The goal of the vector formation strategy is to exploit the dependencies among the elements of the input data set. These dependencies can be linear correlation's or non-linear (all the other dependencies) (Basso, A. & Kunt, M., 1992). Assuming to work in the domain of natural images, a common vector formation strategy consists in forming vectors out of blocks of contiguous pixels of an image, exploiting in this way the inherent pixel to pixel dependence. If the domain is different (subband, transform, vector quantization) usually the dependencies among the data are different, and as a consequence, the vector formation strategy is different. The vector formation strategy discussed above for natural images was followed in this work since the images worked with in this study fall in the domain of natural images. The vector formation strategy used was the process of rasterization. Rasterization is a process of the ordering of blocks of pixels of dimension  $k*k$  pixels of an image to form a vector  $X$  with  $k^2$  elements by columnwise raster scanning (i.e. the first column of pixels in the  $k*k$  block is placed first in the vector  $X$  followed by the second column of pixels then the third, etc.).

The second step in the vector formation process is the choice of the vector size. Vector size in this context defines the dimensionality of the number of inputs to the AANN. The vector size of the input feature space to the AANN was governed by the specifications of the computer being used to train the AANNs as was discussed in Section 3.6. A process of rasterization of the sub-images (dimensions 31\* 31 pixels) gave the input vector size of the AANN. Thus for the sub-images<sup>12</sup> used in this study the vector  $X$  had 961 (31\*31)

<sup>12</sup> From here on in this study sub-images (31\*31 pixels) will be referred to as images.

elements and the vector size of the number of inputs to the AANN corresponds to the number of elements in  $X$  which is 961. Figure 3.7 demonstrates the process of rasterization and the inputting of the rasterized image into the AANN.

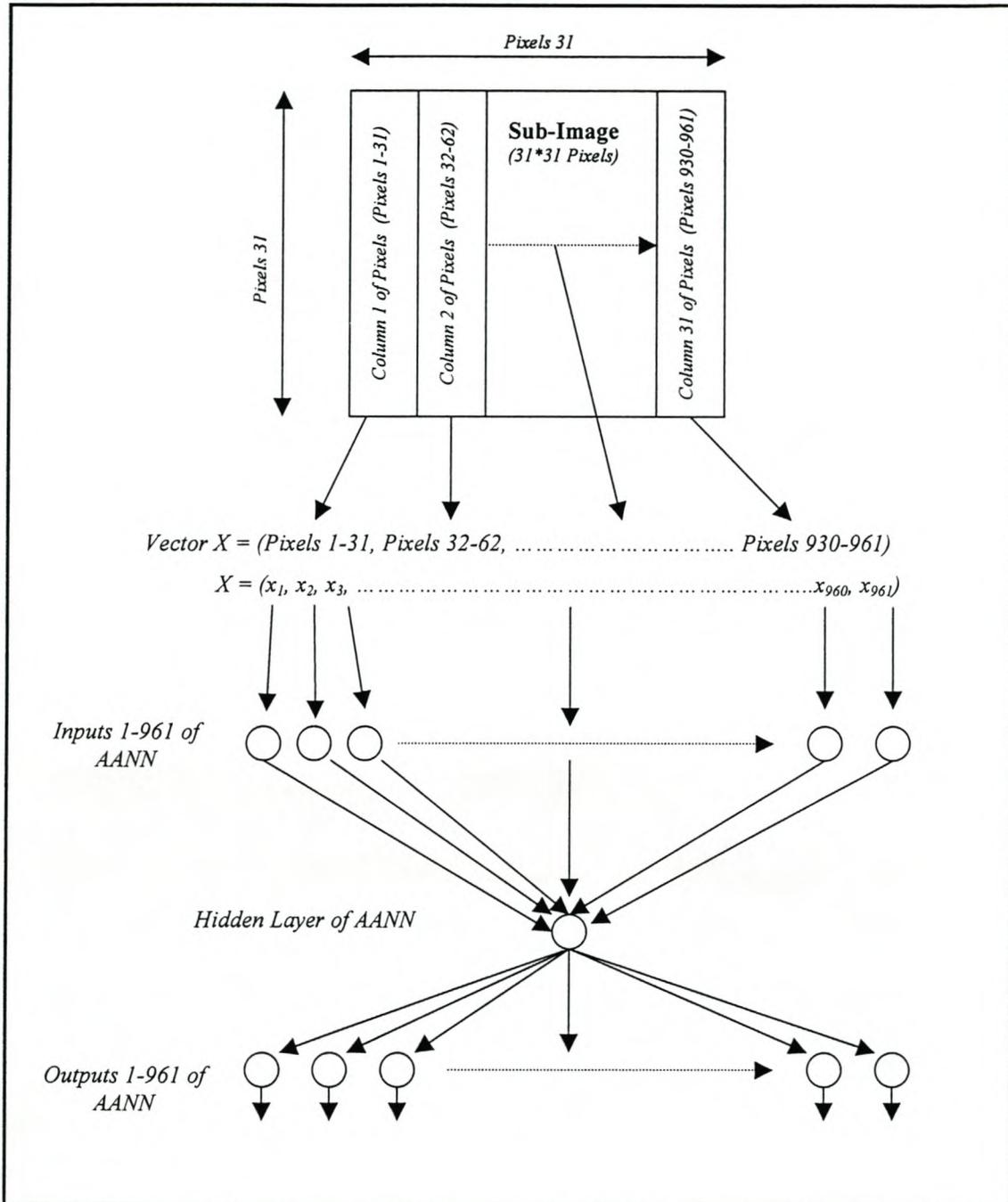


Figure 3.7: Rasterization of sub-images to form a vector  $X$  which is used as an input into the autoassociative neural network

### 3.7 FORMATION OF THE INPUT MATRIX FOR THE AUTOASSOCIATIVE NEURAL NETWORK

All images were pre-processed in the following way.

The mathematical representation of each image was captured using the `Imread` function in Matlabs Image Processing toolbox. The `Imread` function captured the mathematical representation of each image as a  $31 \times 31 \times 3$  matrix. The reason for a three dimensional matrix is because the images were taken in colour. If the images were greyscale intensity images then the `Imread` function would capture the image as a two dimensional matrix

This three dimensional matrix of each image had to be converted to a two dimensional matrix of greyscale intensity values. This was done by converting the colour image, using the `Rgb2gray` function in Matlab, to a greyscale intensity image. The `Rgb2gray` function converts a colour image to a greyscale image by eliminating the hue and saturation information while retaining the luminance. The `Rgb2gray` function converts a colour image to a two-dimensional intensity image with a greyscale colour map, with values ranging between 0 to 255. Now the mathematical representation of each image is in a two-dimensional intensity matrix. In the two dimensional matrix, each entry or element in the matrix represents a single pixel or variable. The elements in the two-dimensional intensity matrix represent various intensities, or grey levels, of the pixels where the intensity 0 usually represents black and the intensity 255 usually represents full intensity, or white.

Thus the two-dimensional intensity matrix ( $31 \times 31$ ) was converted to a vector with 961 ( $31 \times 31$ ) elements by the process of rasterization as discussed in Section 3.6.1. This was done using Matlabs `Reshape` function, which reshaped the  $31 \times 31$  intensity matrix to a  $1 \times 961$  vector. This process was followed for all the images to be used in this study. The

input matrix thus formed was  $m * 961$ <sup>13</sup> where  $m$  represents the number of images to be used in the input matrix, and each image was of size  $1 * 961$ . The  $m * 961$  matrix was then scaled from values between 0-255 to values between 0-1 by dividing each element in the matrix by the element with the largest value in the matrix. The reason for scaling the values from between 0-255 to 0-1 was so that all the inputs into the AANN were equally weighted and so that no one of the 961 variables would have a greater effect on the AANN than the other. This would also make training the AANN much more efficient and quicker.

The  $m * 961$  matrix was then randomised. The randomisation process only shuffled where each image ( $1*961$ ) was placed in the  $m$  rows of the  $m * 961$  matrix. The reason for the randomisation of the  $m * 961$  matrix was to remove any trends in the data.

The  $m * 961$  matrix was then spilt into a training and a test set. Eighty percent of the images were used for training the AANN and the remaining twenty- percent was used for testing the trained AANN model. The training and validation data (sub-images  $31*31$  pixels) for the AANN was selected from the original images ( $248*248$  pixels) due to the small number of images available. Therefore, it was not possible to train the neural network with an entire image (64 sub-images) being left for validation. If an entire image was not used in training, the AANN then the AANN would have being inadequately trained from two points of view. The first would be that the AANN would have not being trained on enough images that would represent a class, and the second being that there would not be enough training exemplars available to train the AANN to be able to generalise. It is important to remember that the greater the number of exemplars that the AANN is trained on the better will be its performance.

---

<sup>13</sup> Any reference to a  $m*n$  matrix refers to a matrix with  $m$  rows and  $n$  columns.

### 3.8 SUMMARY

This chapter starts with an introduction into what PCA is and is followed by a discussion on what is conventional LPCA. The non-linear extension of PCA, which is called NLPCA is discussed as a method for the extraction of non-linear features from data. Various techniques have been developed for NLPCA such as principal surfaces, self-organising maps, generalised linear model, AANNs, generative topographical mapping and probabilistic principal surfaces. AANNs was chosen to perform NLPCA due to its ease of implementation and because it is well suited to the processing of very large data sets. PCA via AANNs is discussed as an alternative method to that of conventional LPCA. The advantages of using AANNs in extracting features from data to that of conventional LPCA is that with the latter can extract linear and non-linear features from data.

Three different versions of AANNs are presented for the performing of PCA. The three layer AANN with a single node in the hidden layer with a linear activation function extracts linear features from data. The three and five layer AANNs with a single and three hidden layers respectively contain non-linear activation functions in the nodes of the hidden layers and extract non-linear features from data.

Two different approaches for performing PCA using an AANN were presented. The first approach discussed extracted the principal components simultaneously while the second approach extracted the non-linear principal components sequentially. The sequential method of principal component extraction was opted for due to this method been able to allow one to directly calculate the amount of variance that each principal component extracts from a data set. This was not possible with the simultaneous approach.

The chapter was concluded with a discussion on how an image of the froth captured during a flotation process is segmented into sub-images and converted to a mathematical representation. The mathematical representation of a sub-image is then ordered into a vector by the process of rasterization. Each vectored sub-image was finally grouped into a matrix, which could be used as an input into an AANN, which extracts the essential features of all the sub-images.

## CHAPTER 4

### EXPERIMENTAL RESULTS

Chapter 3 discussed the development of AANN models to perform LPCA and NLPCA as well as the pre-processing of images which is then converted to data which can be used as inputs to the AANN models. This chapter compares the use of the AANN models developed in Chapter 3 for linear and non-linear PCA with conventional LPCA in the extraction of features from digital images of the surface froth of a laboratory flotation process under differing conditions. The technique of Local PCA, which is based on cluster analysis, and conventional LPCA is also presented as an alternative means for the extraction of features from digital images. The different PCA techniques are compared on the basis of the amount of variance captured by the first three principal components of each technique. The reason for using three principal components and not four or five is due to the ability of human beings been able to visualise in three dimensions. The comparison of the different PCA techniques is based on the calculation of the correlation coefficient and the significance of difference between correlation coefficients, which is given by the z-statistic. The Fishers Z Transformation is used to calculate the z-statistic. The reason for using the Fishers Z Transformation to calculate the z-statistic is to determine whether two correlation coefficients,  $r_1$  and  $r_2$ , drawn from samples of sizes  $N_1$  and  $N_2$ , respectively, differ significantly from each other. Before the results of the

different AANN models are presented a brief review of the flotation process and how images of the surface froth of the flotation process were classified into classes is given.

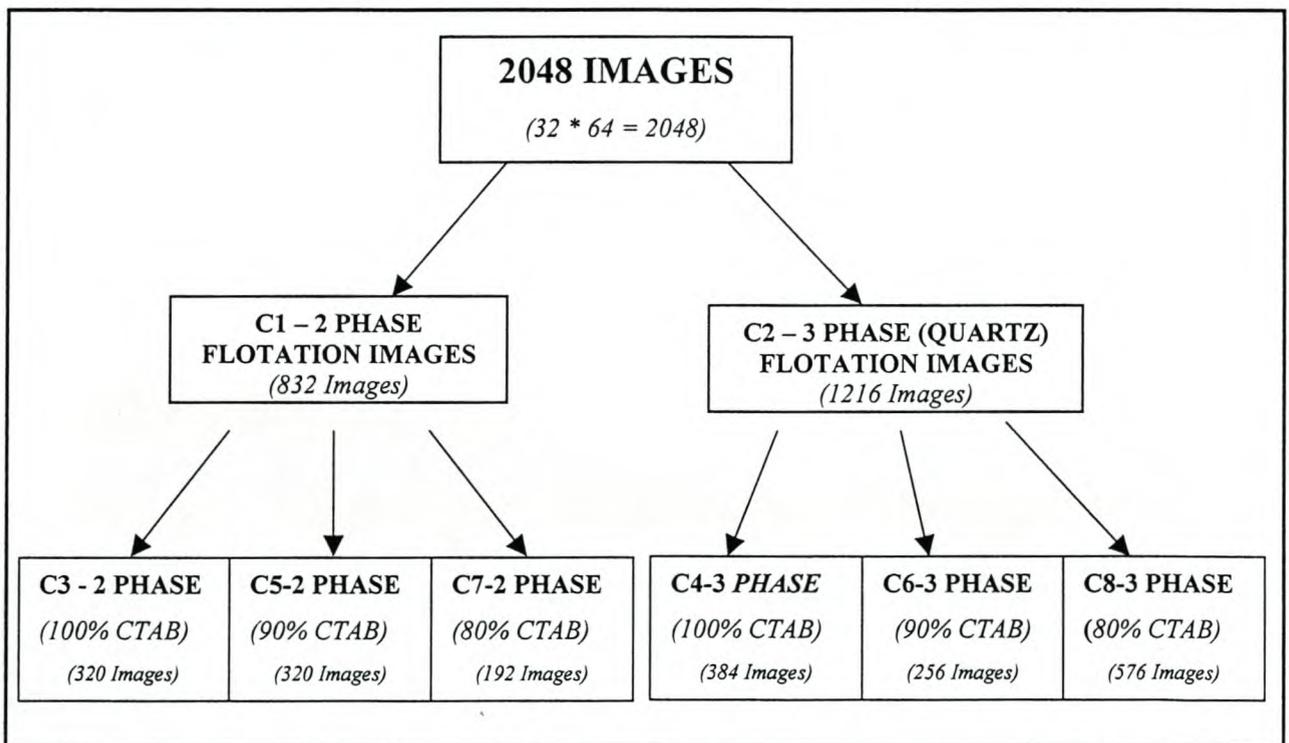
## 4.1 OVERVIEW OF THE FLOTATION PROCESS AND CLASSIFICATION OF SURFACE FROTH IMAGES

Digital images were obtained from the surface of a two and three phase laboratory froth flotation system (Ellis, 2001). For the two-phase study, the surfactants cetyltrimethylammonium bromide (CTAB), with 99% purity and octadecylamine (ODA), with 97% purity were used. The two surfactants were used in different ratios, namely 100%, 90% and 80% with CTAB being the dominant surfactant. Stock solutions of the two surfactants, CTAB and ODA were added in the three different ratios to distilled water. The first phase of the two-phase froth system comprised of liquid, which was a combination of distilled water, CTAB and ODA, and the second phase comprised of air (gaseous phase). Thus three sets of images of the two phase froth were obtained from the three different operating conditions of the laboratory flotation process, they were 100% CTAB two phase, 90% CTAB two phase and 80% CTAB two phase. Different concentrations of surfactants were used to study the effect it has on the bubble size, stability and recovery of the froth.

The three-phase froth system (solid/liquid/gas) was exactly the same as the two-phase froth system (liquid/gas), with the third phase being the solid phase. A fixed amount of quartz in the form of silica sand was used as the solid phase in the three phase froth system. Three sets of images of the three phase froth were obtained from the three different operating conditions of the flotation process, they were 100% CTAB quartz, 90% CTAB quartz and 80% CTAB quartz. The size distributions of the quartz used were  $-75\mu\text{m}$  and  $75-106\mu\text{m}$ . Each size distribution was used for the three operating conditions.

### 4.1.1 CLASSIFICATION OF IMAGES

Thirty-two digital images of the surface froth of the flotation process discussed above were used in this study. These images were captured at the six different operating conditions<sup>14</sup> of the flotation process with a digital camera. Each of the thirty-two images were split into 64 sub images as discussed in the section on Pre-processing of Images (Section 3.6). Thus a total of 2048 ( $32 * 64$ ) images were obtained. The thirty-two original images were categorised into eight different classes of images. Of the eight classes, six classes were obtained from the six different operating conditions of the flotation process. The remaining two classes of images were obtained by combining the images obtained from the three, two phase and three, three phase flotation operating conditions as can be seen in Figure 4.1.



*Figure 4.1: Flowsheet of how the images were split into different classes*

<sup>14</sup> The six different operating conditions are two phase froths, 100% CTAB , 90% CTAB, 80% CTAB and three phase froths, 100% CTAB quartz, 90% CTAB quartz and 80% CTAB quartz.

The 2048 images were categorised into eight different classes as can be seen in Figure 4.1. From here on each class of images will be referred to as the name given in the column headed class reference in Table 4.1.

**Table 4.1: Eight different classes of images with the number of images in each class**

Class Reference	Conditions of Class	Original Number Of Images Captured For Each Class Of Images	Number Of Images In Each Class After Splitting Each Original Image
<b>C1 - 2 Phase Froth</b>	This class is a combination of class C3, C5 and C7	13	832
<b>C2 - 3 Phase Froth</b>	This class is a combination of class C4, C6 and C8	19	1216
<b>C3 - 100% 2 Phase</b>	2 phase, 100% CTAB	5	320
<b>C4 - 100% 3 Phase</b>	3 phase, 100% CTAB, Quartz	6	384
<b>C5 - 90% 2 Phase</b>	2 phase, 90% CTAB	5	320
<b>C6 - 90% 3 Phase</b>	3 phase, 90% CTAB, Quartz	4	256
<b>C7 - 80% 2 Phase</b>	2 phase, 80% CTAB	3	192
<b>C8 - 80% 3 Phase</b>	3 phase, 80% CTAB, Quartz	9	576

Two major classes of images existed in the thirty-two original images. These two classes were class C1-2 phase froth and class C2-3 phase froth. Class C1-2 phase froth (832 images) is a combination of classes C3 (320 Images), C5 (320 Images) and C7 (192 Images) while class C2-3 phase froth (1216 Images) is a combination of classes C4 (384 Images), C6 (256 Images) and C8 (576 Images) as can be seen in Table 4.1. Images of the froth for classes C3-C8 can be seen in Appendix B.

## 4.1.2 NUMBER OF IMAGES USED FOR TRAINING AND TESTING THE AUTOASSOCIATIVE NEURAL NETWORK MODELS

Eighty percent of the images were used to train the AANN models while the remaining twenty percent were used to validate the AANN models. Table 4.2 gives a summary of the number of images used from each class of images for training and testing the AANN models. All the images in class C2-3 phase froth could not be used to train the AANN models due to the computational limitations of the computer been used. Therefore of the total 1216 images, 600 images were used for training and 232 images were used to validate the AANN models. The remaining 384 images were discarded for this study.

*Table 4.2: Number of images used from each class for training and testing the autoassociative neural network models*

Class	Number Of Images Used For Training	Number Of Images Used For Testing
C1	600	232
C2	600	232
C3	280	40
C4	330	54
C5	280	40
C6	215	41
C7	165	27
C8	490	86

## 4.2 CALCULATION OF THE CORRELATION COEFFICIENT AND THE Z-STATISTIC

The correlation coefficient is defined as the measure of strength of the linear relation between two sets of variables. In this study the correlation coefficient was used as a measure of how much correlation there is between the input into the AANN models and the output of the AANN models. The magnitude of the correlation coefficient is an indication of the strength of the linear relationship between the input and output of the AANN models. If the sign of the correlation coefficient is positive it indicates a positive linear relation between the inputs and outputs of the AANN model and if the sign is negative it indicates a negative linear relation between the inputs and outputs of the AANN models. Thus a correlation coefficient close to one represents a strong linear relation and a value close to zero means that the linear association is very weak (Johnson, R.A., Tsui, K.W., 1998).

Thus if the AANN model has a correlation coefficient close to one, it means that the AANN model has reconstructed the input data fairly accurately. This means that the bottleneck layer of the AANN model has captured most of the variance in the input data in a single principal component. If the correlation coefficient is close to zero it means that the bottleneck layer of the AANN model has captured very little of the variance in the input data in a single principal component. Sophisticated techniques to quantify the quality of the reconstructed data are not available and any method is subjective to a certain extent.

The correlation coefficient is also used to calculate the Fishers Z Transformation that will be used in a later stage in this chapter. The correlation coefficient (Aldrich, 1998-1999) is calculated by the following Equation:

$$R = \sqrt{1 - \left( \frac{\sum (x - \hat{x})^2}{\sum (x - \bar{x})^2} \right)} \quad (4.1)$$

where  $x = \text{input}$

$\hat{x} = \text{output}$

$\bar{x} = \text{mean of the input}$

And the Fishers Z transformation is given by:

$$Z = 1.5153 * \log_{10} \left( \frac{1+R}{1-R} \right) \quad (4.2)$$

Where Z is approximately normally distributed (Aldrich, 1998-1999)

### 4.3 CALCULATION OF THE AMOUNT OF VARIANCE CAPTURED BY EACH PRINCIPAL COMPONENT

In this study the correlation coefficient is used as a measure of how well the AANN model performs in recreating the input data into the model. The correlation coefficient explains the amount of variance captured by the principal component<sup>15</sup> extracted from the bottleneck layer of the AANN model. The first three principal components were extracted sequentially as explained in Section 3.4.2 (Sequential PCA via an AANN). Each principal component extracted required a separate AANN model. Thus if three principal components are to be extracted from a data set then three AANN models are required as can be seen in Figure 3.5 (P.55). The following methodology presented is based on sequential PCA using AANNs and is directly related to Figure 3.5.

The correlation between the input and output of the first AANN in Figure 3.5 is used to calculate the amount of variance captured by the first principal component. The

<sup>15</sup> Principal component here refers to either a linear principal component or non-linear principal component depending on the type of the AANN model.

correlation between the input and output of the AANN is calculated using the correlation coefficient. Then the variance captured by the first principal component is given by:

$$\text{Variance captured by the 1}^{\text{st}} \text{ PC} = (\text{CC for AANN 1}) * 100$$

The correlation coefficient is multiplied by 100 to explain the amount of variance captured by the first principal component in terms of a percentage. It is important to remember that the amount of correlation between the input and output of the AANN is directly related to the amount of variance captured by the single node of bottleneck layer which represents the first principal component. In terms of percentage, the total amount of variance in a data set is 100. Therefore, if the first principal component captured a certain amount of variance, then the remainder of the variance not accounted for by the first principal component is given by one hundred subtract the variance captured by the first principal component. The input to the second AANN in Figure 3.5 is calculated by subtracting the input from the first AANN from the output of the first AANN. The variance captured by the second principal component is based on the variance not accounted for by the first principal component. The correlation between the input and output of the second AANN (Measured by the correlation coefficient) multiplied by the variance not accounted for by the first gives the amount of variance captured by the second principal component. The amount of variance captured by the second principal component is given by:

$$\text{Variance captured by 2}^{\text{nd}} \text{ PC} = (100 - \text{Variance captured by 1}^{\text{st}} \text{ PC}) * \text{CC of AANN 2}$$

The amount of variance captured by the third principal component is calculated in a similar fashion as for the second principal component and is given by:

$$\text{Variance captured by 3}^{\text{rd}} \text{ PC} = (100 - \text{Variance captured by 1}^{\text{st}} \text{ PC} - \text{Variance captured by 2}^{\text{nd}} \text{ PC}) * \text{CC of AANN 3}$$

The above procedure for calculating the amount of variance captured by each principal component using sequential principal component extraction was derived by the author in conjunction with his supervisor. The above methodology will be used in the following sections to calculate the amount of variance captured by each principal component. The above procedure is based on the sequential extraction of principal components as illustrated in Section 3.4.2.

## 4.4 CONVENTIONAL LINEAR PRINCIPAL COMPONENT ANALYSIS

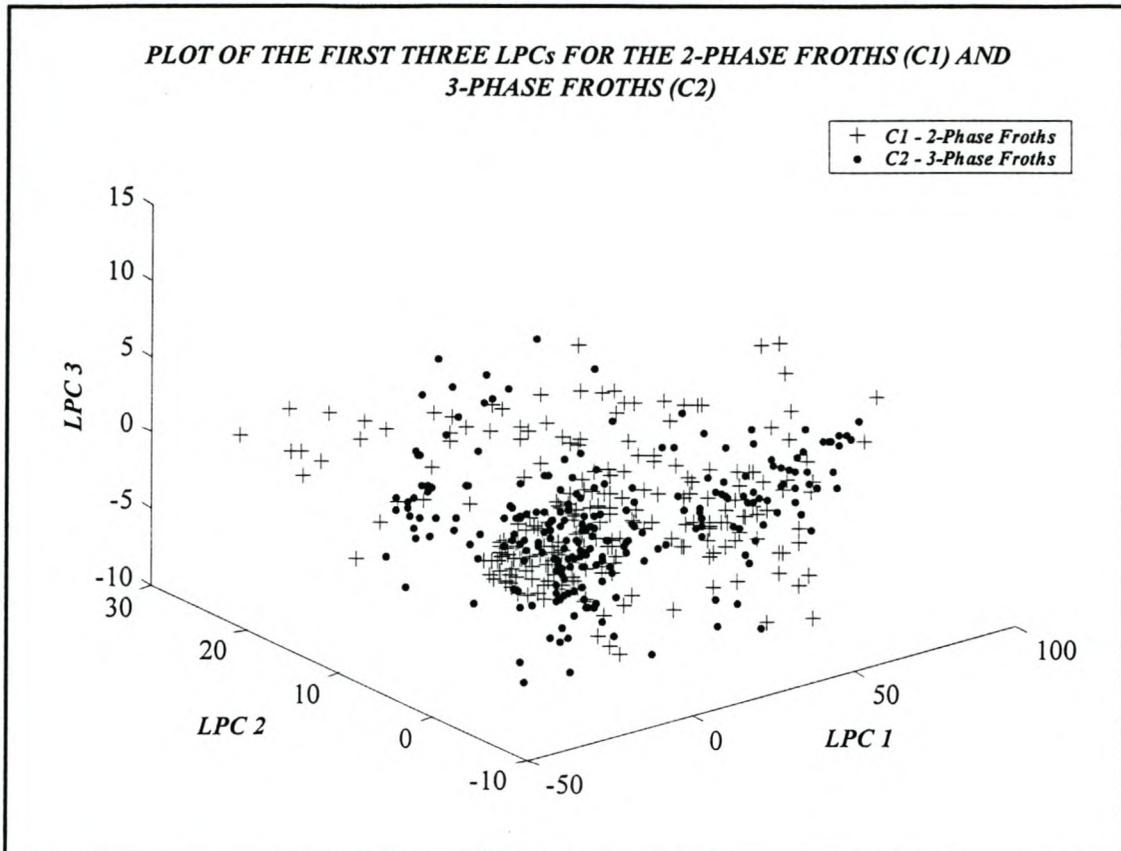
A conventional LPCA was performed in Matlab on the eight different classes of images as defined in Table 4.1. Eighty percent of the images of each class were used to develop the conventional linear principal component models and the remaining twenty percent were used to validate the models as was discussed in Section 4.1.2. The data used to develop the conventional linear principal component models was standardised. The reason for standardisation of the data is to ensure that all variables in the data set are equally weighted in the analysis. If this is not done then those variables which have a larger range would tend to dominate the analysis. The data used to validate the models was standardised by the mean and standard deviation of the data used to develop the conventional linear principal component models. This is done to ensure that the new data falls within the range of the data on which the model was developed.

Table 4.3 gives the amount of variance captured by each linear principal component and the total variance captured by the first three linear principal components. The methodology used to calculate the amount of variance captured by each linear principal component is the method discussed in Section 4.3, which is based on the correlation coefficient. The results in Table 4.3 are based on the validation data for each class of images.

**Table 4.3: Amount of variance captured by the first three linear principal components for each class of data**

Class	Variance Captured by LPC 1 (%)	Variance Captured by LPC 2 (%)	Variance Captured by LPC 3 (%)	Total Variance Captured by First 3 LPCs (%)
C1	90.710	2.950	0.650	94.310
C2	92.810	1.730	0.490	95.040
C3	84.230	3.230	2.260	89.730
C4	92.630	1.700	0.360	94.690
C5	79.960	3.870	1.670	85.500
C6	93.250	3.150	0.590	96.990
C7	61.090	2.360	2.750	66.200
C8	91.080	1.010	1.030	93.120

From Table 4.3 it can be seen that the first linear principal component for each class of images captures over 80% of the variance in the data with the exception of class C7. The second and third linear principal components capture very little of the remaining variance in the data. For class C7 and C8 the third linear principal component captures a little more variance in the data than the second linear principal component. This is possible due to the linear principal component models for these classes been able to reconstruct the test data better with three linear principal components rather than with two. This has not been observed on the data used to develop the models. It is interesting to note that the total variance captured by the first three linear principal components for classes C2, C4, C6 and C8, which all belong to the 3-phase froths are all over 90%,



*Figure 4.2: Plot of the first three linear principal components for images of class C1 and C2 obtained during simulation of the test set by the conventional linear principal component model*

Figure 4.2 is a plot of the scores of the first three linear principal components of images of classes C1 and C2. This plot shows that the first three linear principal components extracted by the conventional linear principal component models of classes C1 and C2 are not able to distinguish between the two classes of images.

## 4.5 PRINCIPAL COMPONENT ANALYSIS USING AN AUTOASSOCIATIVE NEURAL NETWORK

### 4.5.1 LINEAR PRINCIPAL COMPONENT ANALYSIS USING A ONE HIDDEN LAYER AUTOASSOCIATIVE NEURAL NETWORK

The eight different classes of images as defined in Table 4.1 were used to train the AANN model with one hidden layer with a linear activation function. Thus for each class of images a different model was obtained. The number of images from each class that were used for training and testing the AANN model can be seen in Table 4.2.

The AANN used in this section has only linear activation functions in the bottleneck and output layer. Therefore the features extracted from the bottleneck layer are not non-linear principal components but just linear principal components. The sequential method of PCA was opted for due to the reasons discussed in the section on Sequential PCA (Section 3.4.2, P.54). Each class of images were trained and tested using the sequential method of PCA.

Each of the eight AANN models was used to extract the first three linear principal components of each class of images. An indication of how well the eight AANN models were trained can be seen by how each of the eight AANN models performed when they were presented with their respective test sets<sup>16</sup>. The correlation coefficient as defined by Equation 4.1 gives an indication of the performance of the eight AANN models on their respective test sets. Table 4.4 shows the correlation coefficient (R-value) and the corresponding Fishers Z Transformation (Z-value) for the AANN models for the extraction of the first linear principal component for images of class C1– C8. The R-values and Z-values given in the following tables in this chapter are all based on the test sets.

<sup>16</sup> Test sets are data sets which the AANN model has not be trained on and is used to validate the model

**Table 4.4: R and Z values of images of classes C1 to C8 for the extraction of the first linear principal component for each class of images**

Class	R-Value	Z-Value
C1	0.837	1.210
C2	0.913	1.544
C3	0.808	1.120
C4	0.904	1.496
C5	0.681	0.831
C6	0.909	1.522
C7	0.504	0.555
C8	0.900	1.471

The R-values obtained for all eight test sets in Table 4.4 are all above 0.7 with the exception of classes C5 and C7 where a value of 0.5041 and 0.6811 were obtained respectively. The possible reason for class C7 having such a low R-value is due to the fact that the data set on which the AANN model was trained was very small, in fact it was the smallest data set of all 8 classes with just 166 observations used for training. The other 7 classes have reasonably high R-values indicating that the AANN models trained for each of these classes has captured most of the variance in the data and predicts unseen data well.

It is interesting to note how images of the 3-phase froths (classes C2, C4, C6 and C8) all have R-values of approximately 90% while images of the 2-phase froths (classes C1, C3, C5, C7) do not have as high and consistent R-values as those of the 3-phase froth as can be seen in Table 4.4. Table 4.5 shows the correlation coefficient and the corresponding Fishers Z Transformation for the AANN models for the extraction of the second linear principal component for images of class C1-C8.

**Table 4.5: R and Z values of images of classes C1 to C8 for the extraction of the second linear principal component for each class of images**

Class	R-Value	Z-Value
C1	0.714	0.895
C2	0.535	0.597
C3	0.797	1.089
C4	0.444	0.477
C5	0.641	0.760
C6	0.707	0.880
C7	0.303	0.313
C8	0.188	0.190

The R-values in Table 4.5 are not as consistent as in Table 4.4. The AANN models trained for the extraction of the second linear principal component from images of classes C2, C4, C7 and C8 perform very poorly when presented with the residual data obtained from the extraction of the first linear principal component.

Table 4.6 shows the correlation coefficient and the corresponding Fishers Z Transformation for the AANN models for the extraction of the third linear principal component for images of class C1-C8.

**Table 4.7: Amount of variance captured by the first three linear principal components**

Class	Variance Captured by LPC 1 (%)	Variance Captured by LPC 2 (%)	Variance Captured by LPC 3 (%)	Total Variance Captured by all 3 LPCs (%)
C1	83.657	11.669	2.452	97.778
C2	91.282	4.660	0.983	96.926
C3	80.760	15.325	1.028	97.112
C4	90.439	4.243	1.524	96.207
C5	68.115	20.439	5.337	93.891
C6	90.904	6.428	0.345	97.677
C7	50.407	15.029	28.357	93.793
C8	89.979	1.880	2.438	94.296

For classes C2, C4, C6 and C8 (Images of the 3-phase froth) the first linear principal component captures approximately 90% of the variance in the data with the remaining two linear principal components capturing less than 10% of the remaining variance in the data. The first linear principal component for classes C1, C3, C5 and C7 (Images of the 2-phase froth) does not capture as much of the variance in the data compared to that of classes C2, C4, C6 and C8. For these classes the second linear principal component captures a large amount of the variance not accounted for by the first linear principal component. The third linear principal component for classes C7 and C8 captures more variance in the data than the second linear principal component. This could be due to the second AANN models for these classes having been poorly trained. It should be noted that the AANNs trained for the extraction of the second and third linear principal components were not trained as rigorously as the AANNs for the extraction of the first linear principal component for all classes of images.

Three linear principal components were captured for each of the eight classes of images. The first reason for extracting three linear principal components for each class of images was for easy visualisation. The second reason was because a neural network classifier

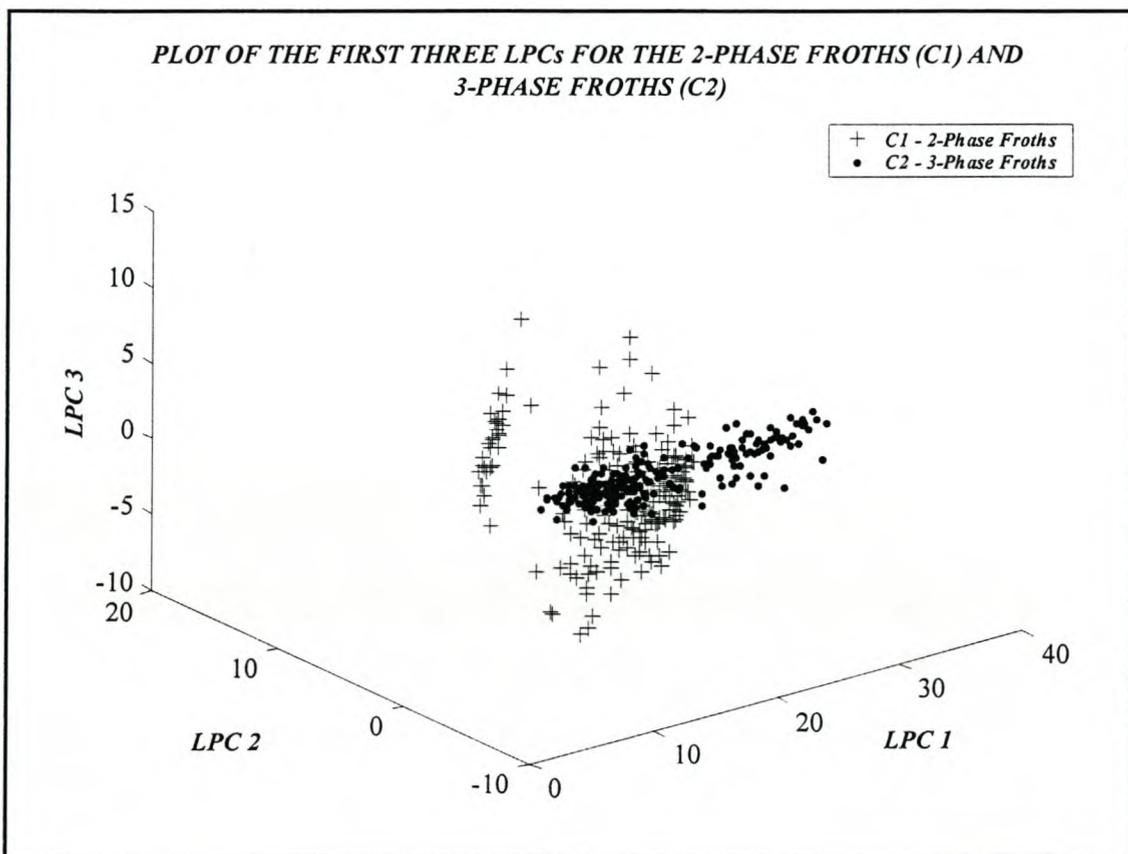
**Table 4.6: R and Z values of images of classes C1 to C8 for the extraction of the third linear principal component for each class of images**

Class	R-Value	Z-Value
C1	0.525	0.583
C2	0.242	0.247
C3	0.263	0.269
C4	0.287	0.295
C5	0.466	0.505
C6	0.129	0.130
C7	0.820	1.158
C8	0.299	0.309

The AANN models trained to extract the third linear principal component all performed poorly except for the AANN model for images of class C7. Table 4.6 shows that six of the AANN models have R-values of less than 50% while C1 had a R-value of just over 50% and C7 a R-value of over 80%. The AANN model for images of class C7 has done poorly for the extraction of the first two linear principal components. This is reflected in their low R-values of 0.50 and 0.3 respectively. The possible reason for this is due to the small number of observations available for the training of the AANN model for images of class C7. More linear principal components need to be extracted to explain the remaining variance in the data.

The amount of variance captured by each of the three linear principal components was calculated using the methodology as discussed in Section 4.3. Table 4.7 gives the amount of variance captured by each of the three linear principal components as well as the total variance captured by all three linear principal components for each class of images.

using the first three linear principal components would be able to classify images more accurately than a neural network classifier just using one or two linear principal components. The reason for not using more than three principal components for classification is because the first three principal components capture most of the variance in the data and due to it being easier to train the classifier with three inputs rather than with four or five. Plots of the first three linear principal components of only the two major classes of images namely the 2-phase froths (Class C1) and the 3-phase froths (Class C2) were plotted. Figure 4.3 is a plot of the first three linear principal components obtained during testing of the AANN models for images of class C1-2-phase froths and C2-3-phase froths.



*Figure 4.3: Plot of the first three linear principal components for images of class C1 and C2 obtained during simulation of the test set by the autoassociative neural network with one hidden layer (Linear activation function)*

Figure 4.3 shows how the linear principal components extracted by the AANN model for class C1 forms a cluster which is different to the cluster formed by the linear principal components extracted by the AANN model for class C2. The linear principal components plotted in this figure for class C1 and C2 are based on the test sets. If the three-dimensional plot was rotated, it would be possible to see the clusters more clearly.

#### 4.5.2 NON-LINEAR PRINCIPAL COMPONENT ANALYSIS USING A ONE HIDDEN LAYER AUTOASSOCIATIVE NEURAL NETWORK

The eight different classes of images as defined by Table 4.1 were used to train the AANN with one hidden layer with a non-linear activation function. Thus for each class of images a different AANN model was obtained. Table 4.2 gives the number of images from each class that were used for training and testing the AANN models.

*Table 4.8: R and Z values of images of classes C1 to C8 for the extraction of the first non-linear principal component for each class of images*

Class	R-Value	Z-Value
C1	0.847	1.245
C2	0.906	1.505
C3	0.774	1.029
C4	0.885	1.397
C5	0.694	0.855
C6	0.906	1.502
C7	0.507	0.558
C8	0.887	1.409

The R-values in Table 4.8 are very similar to those of Table 4.4. Thus the AANN with one hidden layer (linear activation function) and the AANN with one hidden layer (non-linear activation function) perform equally well when extracting the first principal component. Images of the 3-phase froth (classes C2, C4, C6 and C8) all have R-values of approximately 90% while those of the 2-phase froth (classes C1, C3, C5, C7) are not as high and consistent as can be seen in Table 4.8. This trend is consistent with that observed for the AANN models with one hidden layer (linear activation function) and the conventional linear principal component models.

**Table 4.9: R and Z values of images of classes C1 to C8 for the extraction of the second non-linear principal component for each class of images**

Class	R-Value	Z-Value
C1	0.693	0.853
C2	0.412	0.438
C3	0.359	0.376
C4	0.292	0.301
C5	0.452	0.487
C6	0.448	0.482
C7	0.394	0.416
C8	0.103	0.103

The R-value of the AANN model for the extraction of the second non-linear principal component for class C1 is approximately 70% indicating that the second non-linear principal component captures most of the remaining variance in the data not captured by the first non-linear principal component. The R-values for the remaining seven AANN models indicate that the second non-linear principal component captures little of the remaining variance in the data not captured by the first non-linear principal component.

**Table 4.10: R and Z values of images of classes C1 to C8 for the extraction of the third non-linear principal component for each class of images**

Class	R-Value	Z-Value
C1	0.376	0.396
C2	0.203	0.206
C3	0.120	0.121
C4	0.186	0.188
C5	0.550	0.618
C6	0.927	1.636
C7	0.302	0.312
C8	0.762	1.000

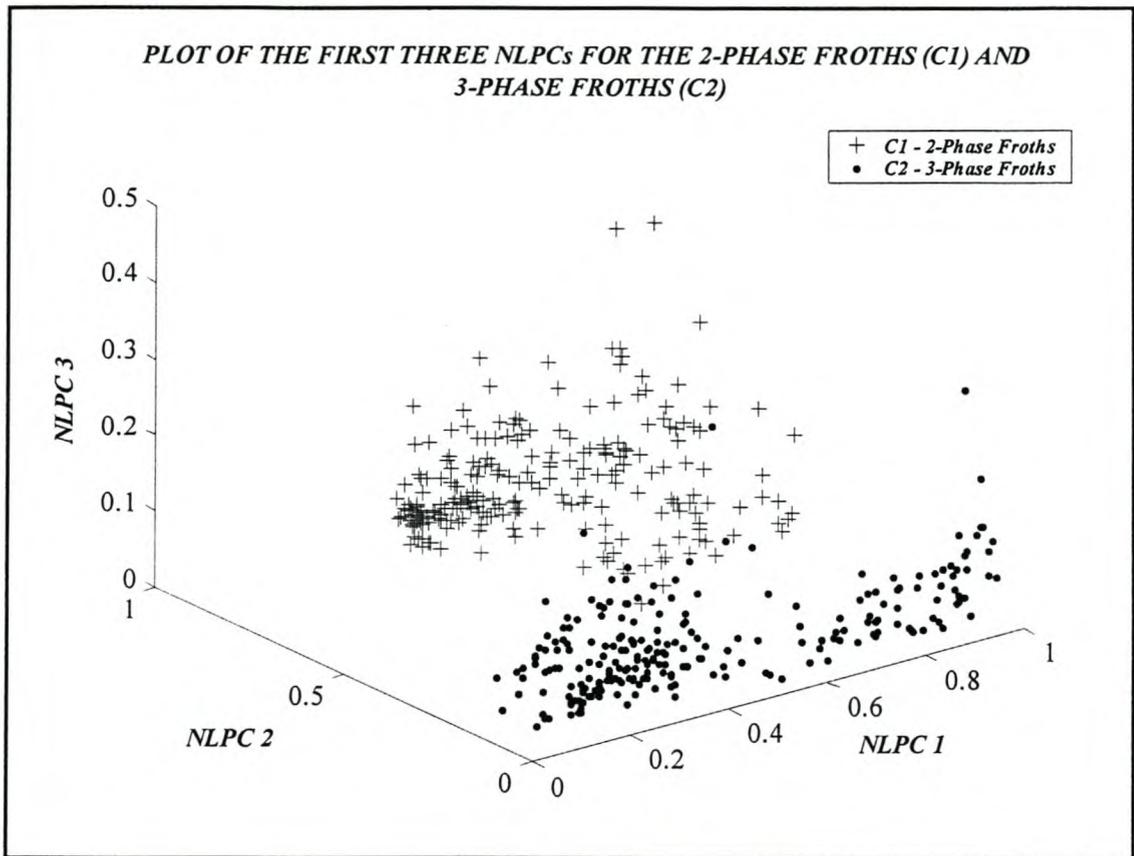
The third non-linear principal component for each class of images captures very little of the remaining variance in the data not captured by the first two non-linear principal components with the exception of classes C6 and C8 as can be seen in Table 4.10. The amount of variance captured by each of the three non-linear principal components was calculated using the methodology as discussed in Section 4.3. Table 4.11 gives the amount of variance captured by each of the three non-linear principal components as well as the total variance captured by all three-non-linear principal components for each class of images.

**Table 4.11: Amount of variance captured by the first three non-linear principal components for each class of data**

Class	Variance Captured by NLPC 1 (%)	Variance Captured by NLPC 2 (%)	Variance Captured by NLPC 3 (%)	Total Variance Captured by all 3 NLPCs (%)
C1	84.672	10.618	1.773	97.063
C2	90.606	3.867	1.124	95.597
C3	77.349	8.138	1.744	87.232
C4	88.463	3.367	1.517	93.347
C5	69.366	13.852	9.229	92.447
C6	90.546	4.232	4.840	99.618
C7	50.663	19.421	9.043	79.128
C8	88.727	1.160	7.703	97.590

Table 4.11 has similar trends to those of Table 4.7 and Table 4.3. The amount of variance captured by the first non-linear principal component for classes C2, C4, C6 and C8 (Images of the 3-phase froth) is approximately 90%. The remaining two non-linear principal components captures less than 10% of the variance in the data for these classes. The first non-linear principal component for classes C1, C3, C5 and C7 (Images of the 2-phase froth) does not capture as much of the variance in the data compared to that of classes C2, C4, C6 and C8. For these classes the second non-linear principal component captures a large amount of the variance not accounted for by the first non-linear principal component. The third non-linear principal component for classes C6 and C8 captures more variance in the data than the second non-linear principal component. The possible reason for this is due to the poor training of these networks as mentioned in the previous section. For class C8 there seems to be a trend in that the third non-linear principal component extracts more variance than the second non-linear principal component. This can be seen in Table 4.3, Table 4.7 and Table 4.11.

Plots of the first three non-linear principal components of only the two major classes of images namely the 2-phase froth (Class C2) and 3-phase froth (Class C1) have been plotted.



*Figure 4.4: Plot of the first three non-linear principal components for images of class C1 and C2 obtained during simulation of the test set by the autoassociative neural network with one hidden layer (Non-linear activation function)*

Figure 4.4 shows how the first three non-linear principal components extracted by the two AANN models for classes C1 and C2 clusters each class of data into almost distinct clusters. Slight changes in the co-ordinates of just a few points would result in perfect separation of the two clusters. From this plot it can be concluded that the three non-linear principal components extracted via the two AANN models characterises each data set uniquely.

The amount of variance captured by the first three principal components of the AANN models with one hidden layer (Non-linear activation function) and one hidden layer (Linear activation function), for classes C1 and C2 are approximately the same. Yet when one views Figure 4.4 and Figure 4.3 it can be seen that the AANN models with one hidden layer (Non-linear activation function) are able to extract principal components which form more distinct clusters for classes C1 and C2. The AANN with one hidden layer (Linear activation function) extracts linear principal components, which create clusters for classes C1 and C2, which overlap. Although the performance of the two AANNs may seem the same from the point of the amount of variance captured by the first three principal components, they are not from the point of separating the clusters distinctly.

#### **4.5.3 NON-LINEAR PRINCIPAL COMPONENT ANALYSIS USING A THREE HIDDEN LAYER AUTOASSOCIATIVE NEURAL NETWORK**

The eight different classes of images as defined by Table 4.1 were used to train the AANN with three hidden layers with non-linear activation functions (sigmoidal activation functions). Thus for each class of images a different model was obtained. Table 4.2 gives the number of images from each class that were used for training and testing the AANN models.

**Table 4.12: R and Z values of images of classes C1 to C8 for the extraction of the first non-linear principal component for each class of images**

Class	R-Value	Z-Value
C1	0.862	1.302
C2	0.909	1.523
C3	0.926	1.626
C4	0.902	1.484
C5	0.722	0.915
C6	0.909	1.524
C7	0.518	0.574
C8	0.896	1.453

The R-values in Table 4.12 are all approximately 90% with the exception of class C5 and C7, which are approximately 72% and 51% respectively. Similar R-values have been obtained for classes C5 and C7 using the AANN model with one hidden layer (Linear activation function) and the AANN model with one hidden layer (Non-linear activation function) as can be seen in Table 4.4 and Table 4.8 respectively.

**Table 4.13: R and Z values of images of classes C1 to C8 for the extraction of the second non-linear principal component for each class of images**

Class	R-Value	Z-Value
C1	0.626	0.734
C2	0.044	0.044
C3	0.230	0.234
C4	0.0260	0.026
C5	0.598	0.690
C6	0.053	0.053
C7	0.424	0.453
C8	0.311	0.322

The AANN models for extracting the second non-linear principal component for images of class C1 to C8 perform poorly when presented with the residuals of the test data from the extraction of the first non-linear principal component. This is indicated by the low R-values in Table 4.13.

**Table 4.14: R and Z values of images of classes C1 to C8 for the extraction of the third non-linear principal component for each class of images**

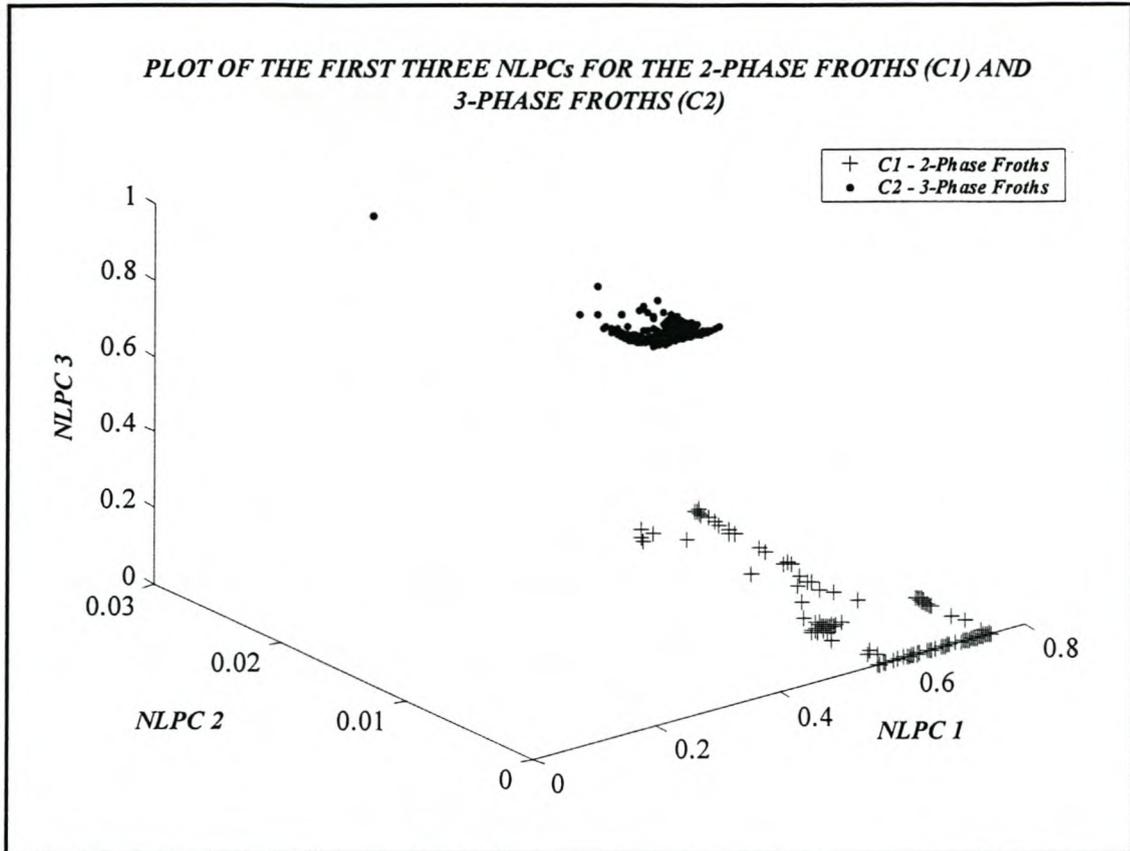
Class	R-Value	Z-Value
C1	0.254	0.259
C2	0.052	0.052
C3	0.771	1.022
C4	0.014	0.014
C5	0.537	0.600
C6	0.054	0.054
C7	0.156	0.157
C8	0.047	0.047

The R-values obtained in Table 4.14 are all low with the exception of class C3. These low R-values indicate that that the first two non-linear principal components have captured most of the variance in the data. Table 4.15 gives the amount of variance captured by each of the three non-linear principal components as well as the total variance captured by all three non-linear principal components for each class of data.

**Table 4.15: Amount of variance captured by the first three non-linear principal components for each class of images**

Class	Variance Captured by NLPC 1 (%)	Variance Captured by NLPC 2 (%)	Variance Captured by NLPC 3 (%)	Total Variance Captured by all 3 NLPCs (%)
C1	86.228	8.615	1.308	96.152
C2	90.925	0.399	0.453	91.777
C3	92.554	1.711	4.419	98.684
C4	90.223	0.253	0.129	90.605
C5	72.238	16.603	5.989	94.830
C6	90.934	0.476	0.463	91.874
C7	51.843	20.428	4.318	76.588
C8	89.618	3.231	0.338	93.187

The trends observed in Table 4.15 are similar to those observed in Table 4.7 and Table 4.11. For classes C2, C4, C6 and C8 (Images of the 3-phase froth) the first non-linear principal component captures approximately 90% of the variance in the data with the remaining two non-linear principal components capturing less than 10% of the variance in the data. The first non-linear principal component for classes C1, C5 and C7 (Images of the 2-phase froth) does not capture as much of the variance in the data compared to that of classes C2, C6 and C8 (3-phase froth) only class C3 (2-phase froth) captures more variance than C4 (3-phase froth) with the first non-linear principal component. For these classes the second non-linear principal component captures a large amount of the variance not accounted for by the first non-linear principal component. A plot of the first three non-linear principal components of only class C1 (2-phase froth) and C2 (3-phase froth) were plotted. Figure 4.5 is a plot of the first three non-linear principal components obtained during testing of the AANN models for images of class C1 and C2.



**Figure 4.5:** Plot of the first three non-linear principal components for images of class C1 and C2 obtained during simulation of the test set by the autoassociative neural network with three hidden layers (Non-linear activation function)

When the AANN models were presented with test data, the non-linear principal components extracted by the models characterise each class of images uniquely. Figure 4.5 shows that the non-linear principal components defining each of the two classes of images forms two distinct clusters which are totally separated. If Figure 4.5 is compared with Figure 4.3 and Figure 4.4 one realises that the AANN with three hidden layers (Non-linear activation function) performs better in extracting non-linear principal components which define classes C1 and C2 uniquely.

## 4.6 LOCAL PRINCIPAL COMPONENT ANALYSIS

Local principal component analysis was performed on each of the eight classes of images. Each class of images was partitioned into clusters using k-means clustering. The number of clusters each class of images was in real time into depended on the number of exemplars in the class. The larger classes of data such as classes C1 and C2 were in real time into eight clusters. While smaller classes of less than 300 images were in real time into three clusters. Classes with more than 300 images but less than 600 images were in real time into four clusters. The number of images belonging to each cluster for each class of images can be seen in Table 4.16.

*Table 4.16: Number of images belonging to each cluster for each class of images*

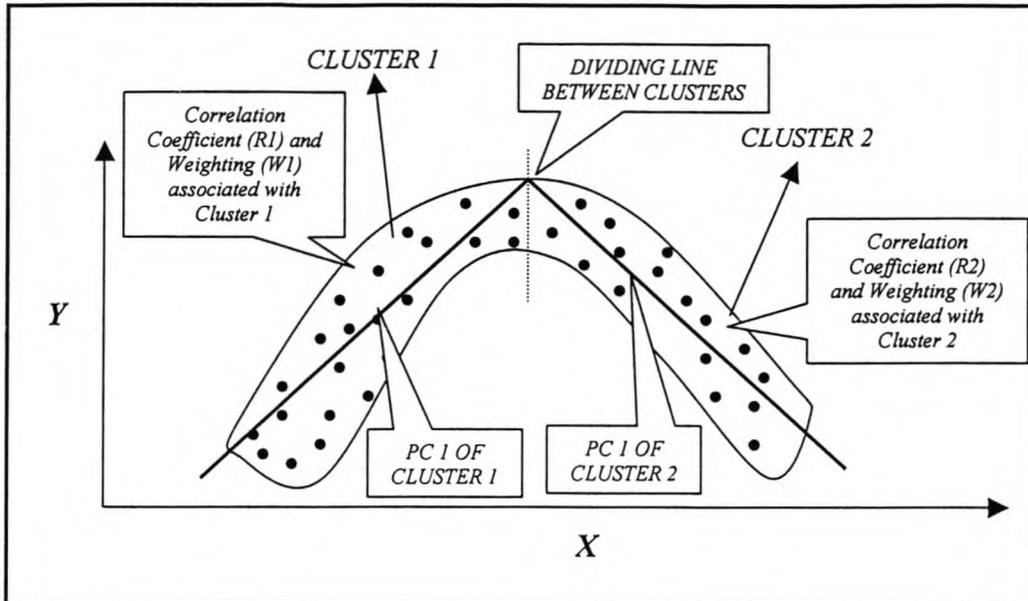
CLASS	C1	C2	C3	C4	C5	C6	C7	C8
CLUSTER 1	154	173	121	117	100	107	83	181
CLUSTER 2	145	154	110	105	82	75	97	94
CLUSTER 3	142	130	49	88	77	74	12	76
CLUSTER 4	118	126	40	74	61	0	0	120
CLUSTER 5	105	80	0	0	0	0	0	105
CLUSTER 6	81	72	0	0	0	0	0	0
CLUSTER 7	49	71	0	0	0	0	0	0
CLUSTER 8	38	26	0	0	0	0	0	0
TOTAL	832	832	320	384	320	256	192	576

Conventional LPCA was performed on each cluster of images using Matlab's principal component function in the statistics toolbox. Eighty percent of the images in each cluster were used to obtain a conventional linear principal component model while the remaining twenty percent were used to test the models. The data used to develop the conventional linear principal component models for each cluster were standardised. The data used to validate the conventional linear principal component models for each cluster was standardised by the mean and standard deviation of the data used to develop the conventional linear principal component models of each cluster. The correlation

coefficient as defined by Equation 4.1 was used as a measure of how well the conventional linear principal component models of each cluster recreated the test data of that cluster.

The amount of variance captured by the first piecewise linear principal component for each class of images was calculated by weighting the correlation coefficient obtained from the reconstruction of the test data of each cluster in a class using the first linear principal component and summing the weighted correlation coefficient of all the clusters in each class. The weighting of the clusters in each class of images was calculated by taking the number of images in a cluster and dividing it by the total number of images in its class.

The amount of variance captured by the second piecewise linear principal component of each class of images was calculated by reconstructing the test data of each cluster using the first two linear principal components and summing the weighted correlation coefficient and subtracting it from the correlation coefficient obtained for the first piecewise linear principal component. The amount of variance captured by the third piecewise principal component is calculated in a similar fashion. The methodology is similar to one presented in Section 4.3 with the difference been that the correlation coefficient referred to in this section, is obtained by summing the weighted correlation coefficient of each cluster in a class. The above procedure is illustrated in the figure to follow.



**Figure 4.6: Illustration of piecewise linear principal component analysis**

Figure 4.6 is an illustration of piecewise linear principal component analysis. The diagram illustrates how a data set is clustered into two clusters. Principal component analysis is performed on each standardised cluster. The principal component analysis yields the linear principal components of each cluster. The first linear principal component of each cluster can then be joined in a piecewise fashion to yield the first principal component of the overall data set as can be seen in the figure above. In a similar fashion, the second principal component of the entire data set can be obtained. Larger data sets can be in real time in a number of clusters and principal component analysis can be performed on each cluster. Then the first linear principal component of each cluster can be joined in a piecewise fashion to give the first principal component of the entire data set.

The correlation coefficient ( $R$ ) associated with the reconstruction of the entire data set is obtained by summing the weighted correlation coefficient ( $R1$ ) for the reconstruction of Cluster 1 and the weighted correlation coefficient ( $R2$ ) for the reconstruction of Cluster 2. The weighting of each cluster is obtained by dividing the number of observations in a

cluster by the total number of observations in the data set. Therefore the correlation coefficient associated with the entire data set is given by:

$$R = W1*R1 + W2*R2 \quad (4.3)$$

$$W1 = \frac{\text{Number of observations in Cluster 1}}{\text{Total number of observations in the data set}} \quad (4.4)$$

$$W2 = \frac{\text{Number of observations in Cluster 2}}{\text{Total number of observations in the data set}} \quad (4.5)$$

In a similar fashion the correlation coefficient for a data set with any number of clusters can be calculated. Table 4.17 gives the weightings of each cluster in their respective classes.

**Table 4.17: Weightings of each cluster for each class of images**

CLASS	C1	C2	C3	C4	C5	C6	C7	C8
CLUSTER 1	0.185	0.208	0.378	0.305	0.313	0.418	0.432	0.314
CLUSTER 2	0.174	0.185	0.344	0.273	0.256	0.293	0.505	0.163
CLUSTER 3	0.171	0.156	0.153	0.229	0.241	0.289	0.063	0.132
CLUSTER 4	0.142	0.151	0.125	0.193	0.191	0.000	0.000	0.208
CLUSTER 5	0.126	0.096	0.000	0.000	0.000	0.000	0.000	0.182
CLUSTER 6	0.097	0.087	0.000	0.000	0.000	0.000	0.000	0.000
CLUSTER 7	0.059	0.085	0.000	0.000	0.000	0.000	0.000	0.000
CLUSTER 8	0.046	0.031	0.000	0.000	0.000	0.000	0.000	0.000
TOTAL	1	1	1	1	1	1	1	1

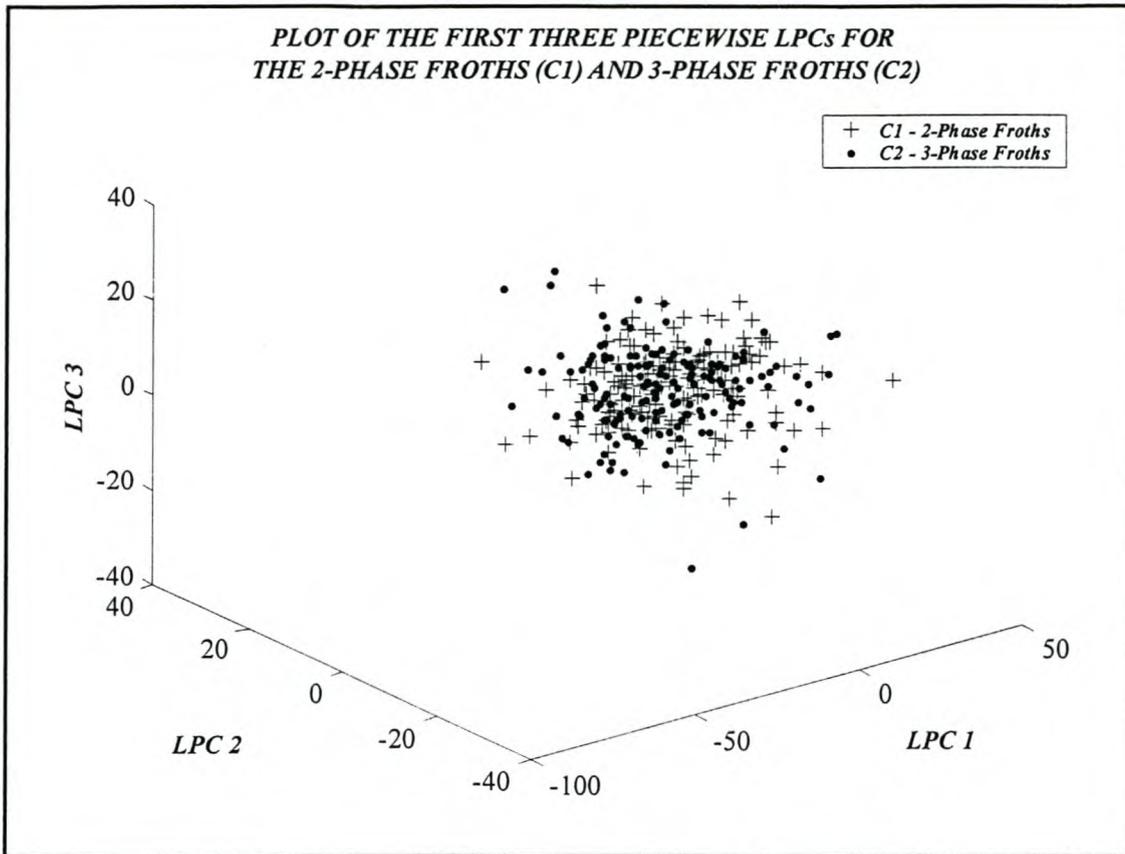
The amount of variance captured by the first three piecewise linear principal components for each class of images is given in Table 4.18. From the table it can be seen that the local principal component models for each class of images do not reconstruct the test data well. The models based on the first, first two and first three piecewise linear principal components do not capture more than 60% of the variance in the test data for any class of

images. For each class of images the first piecewise linear principal component is the most dominant principal component.

*Table 4.18: Variance captured by the first three piecewise linear principal components of each class of images*

Class	Variance Captured by Piecewise LPC 1 (%)	Variance Captured by Piecewise LPC 2 (%)	Variance Captured by Piecewise LPC 3 (%)	Total Variance Captured by all 3 Piecewise LPCs (%)
C1	41.380	7.640	5.150	54.170
C2	42.870	7.660	5.480	56.010
C3	44.170	7.480	3.060	54.710
C4	47.770	7.350	2.310	57.440
C5	36.550	6.700	2.980	46.220
C6	43.530	9.360	2.780	55.680
C7	43.900	4.500	3.130	51.540
C8	40.540	6.390	3.740	50.670

Figure 4.7 is a plot of the scores of the first three piecewise linear principal components of classes C1 (2-phase froths) and C2 (3-phase froths). The scores plotted are of the eight clusters for classes C1 and C2. From the plot, it can be seen that the first three piecewise linear principal components of each of the eight clusters for classes C1 and C2 occupy the same regions in the plot. From this plot it can be seen that local PCA is not able to extract features which distinguish between the two classes of images as was the case with conventional LPCA.



**Figure 4.7: Plot of the first three piecewise linear principal components for the 2 and 3 phase froths**

Theoretically, local PCA is able to capture more of the variance in data sets, which have non-linear relationships than conventional LPCA. If local PCA is applied to a linear data set it would capture as much or more of the variance in the data than conventional LPCA. Therefore the amount of variance captured by the first three piecewise linear principal components for local PCA, for the 2 and 3 phase froths, should be more or, at minimum, equal to the variance captured by the first three linear principal components of conventional LPCA. This has not been the case as can be seen when Table 4.3 and Table 4.18 are compared. Conventional LPCA captures more of the variance in the data for all eight classes of data. A possible reason for the results not being in accordance with what was expected could be due to the clustering of the data. The data sets used in this study consisted of 961 dimensions, which are far too many dimensions for any clustering

technique to be able to effectively cluster the data into meaningful clusters. Data sets of approximately twenty-five dimensions and less can be effectively clustered by present clustering techniques to yield meaningful clusters. Therefore, due to the poor clustering of the data, local PCA was not able to capture as much of the variance in the data compared to that of conventional LPCA.

A non-linear two-dimensional set of data will be used to illustrate how local PCA is able to capture more of the variance in non-linear data than conventional LPCA in the example to follow.

### **Example: Reconstruction of Two Dimensional Data Using Local PCA and Conventional LPCA**

Non-linear data of the form shown in Figure 4.6 is used in this example to illustrate how local PCA is able to extract more variance from non-linear data than conventional LPCA. The non-linear data was generated by combining data obtained from the following two equations:

$$\text{Equation 1: } Y = X$$

$$\text{Equation 2: } Y = 10 - X$$

Random noise with a normal distribution was added on to the data to make the problem slightly more difficult. The non-linear data set consisted of 1002 observations. The data set was partitioned into two clusters by use of k-means cluster analysis. Cluster 1 consisted of 476 observations and Cluster 2 consisted of 526 observations. Eighty percent of the observations in each cluster were used to model each cluster while the remaining twenty-percent was used to validate the models. Conventional LPCA was performed on each cluster as well as on the entire data set of 1002 observations. The amount of variance captured by local PCA with the first piecewise linear principal component for the validation data was calculated according to Equations 4.3. The weightings  $W_1$  and

W2 were calculated according to Equations 4.4 and 4.5 and are based on the validation data.

$$W1 = \frac{95}{95 + 105} = 0.475$$

$$W2 = \frac{105}{95 + 105} = 0.525$$

The correlation coefficients for the reconstruction of the validation data for Cluster 1 and Cluster 2 using the first linear principal component was 0.987 (R1) and 0.985 (R2) respectively. The correlation coefficients for the reconstruction of the validation data for both Cluster 1 and Cluster 2 using the first two linear principal components was approximately 1. The amount of variance (R) captured by the first piecewise linear principal component for local PCA was calculated as follows:

$$R = 0.475 * 0.987 + 0.525 * 0.985 = 0.986$$

The amount of variance (R) captured by the first two piecewise linear principal components for local PCA was approximately 1. Table 4.19 gives a comparison of the performance of local PCA compared to that of conventional LPCA. The one-dimensional representation of the data using local PCA captures approximately 99% of the variance of the data whereas the one-dimensional representation of the data using conventional LPCA captures only approximately 72% of the variance of the data. The two-dimensional representation of the data for local PCA and conventional LPCA is the same as can be seen in Table 4.19. From the above example it can be seen that local PCA can model non-linear data more accurately than conventional LPCA.

*Table 4.19: Comparison of the amount of variance captured by local PCA and conventional LPCA*

<b>Number Of LPCs Used In Reconstruction Of Data</b>	<b>Correlation Coefficient for Local PCA (Cluster 1 and Cluster 2 Modelled Separately)</b>	<b>Correlation Coefficient for Conventional LPCA (Cluster 1 and Cluster 2 combined and Modelled )</b>
First LPC	0.986	0.718
First two LPCs	$\cong 1$	$\cong 1$

## 4.7 COMPARISON OF TECHNIQUES FOR PRINCIPAL COMPONENT ANALYSIS

Five techniques have been presented for performing PCA. The techniques are conventional LPCA, LPCA using an AANN with one hidden layer (Linear activation function), NLPCA using an AANN with one hidden layer (Non-linear activation function), NLPCA using an AANN with three hidden layers (Non-linear activation functions) and local principal component analysis. These techniques have been used to extract the first three principal components from the eight classes of data described in Table 4.1. This section compares the amount of variance captured for the first, second and third principal components and the total variance captured by all three principal components based on the five PCA techniques. The variances presented in the following four tables are based on percentages.

### 4.7.1 FIRST PRINCIPAL COMPONENT

The first principal component captured the most amount of variance for all five PCA techniques. Conventional LPCA captured the most amount of variance in the extraction of the first principal component for all the test sets of the eight classes of images except for that of class C3 as can be seen in Table 4.20. For classes C5 and C7, conventional LPCA captured an extra 10% of the variance in the data than any other technique. For classes C2, C4, C6 and C8 (Classes belonging to the 3-phase froths) the variance captured by the first principal component using the first four techniques in Table 4.20 are all approximately 90%. For classes C1, C3, C5 and C7 (Classes belonging to the 2-phase froths) the variance captured by the first principal component using the first four techniques in Table 4.20 are not so consistent. All techniques perform poorly on class C7. This could be due to the small number of images in that class. Local PCA extracts the least amount of variance for all eight classes for the first principal component.

*Table 4.20: Comparison of the amount of variance captured by the first principal component using the five principal component analysis techniques for each class of images*

Class	Conventional LPCA	LPCA using an AANN with 1 hidden layer	NLPCA using an AANN with 1 hidden layer	NLPCA using an AANN with 3 hidden layers	Local Principal Component Analysis
C1	90.710	83.657	84.672	86.228	41.380
C2	92.810	91.282	90.606	90.925	42.870
C3	84.230	80.760	77.349	92.554	44.170
C4	92.630	90.439	88.463	90.223	47.770
C5	79.960	68.115	69.366	72.238	36.550
C6	93.250	90.904	90.546	90.934	43.530
C7	61.090	50.407	50.663	51.843	43.900
C8	91.080	89.979	88.727	89.618	40.540

## 4.7.2 SECOND PRINCIPAL COMPONENT

For classes C1, C3, C5 and C7, LPCA using an AANN with 1 hidden layer captured the most amount of variance with the second principal component as can be seen in Table 4.21. For classes C2, C4, C6, and C8, local PCA captured the most amount of variance with the second principal component. From Table 4.21 it can be seen that conventional LPCA captures very little of the variance in the second principal component for all eight classes. The amount of variance captured by the second principal component using the AANNs for classes C1, C5 and C7 are all approximately greater than 10% indicating that for these classes the second principal component captures a fair amount of the remaining variance in the data.

**Table 4.21: Comparison of the amount of variance captured by the second principal component using the five principal component analysis techniques for each class of images**

Class	Conventional LPCA	LPCA using an AANN with 1 hidden layer	NLPCA using an AANN with 1 hidden layer	NLPCA using an AANN with 3 hidden layers	Local Principal Component Analysis
C1	2.950	11.669	10.618	8.615	7.640
C2	1.730	4.660	3.867	0.399	7.660
C3	3.230	15.325	8.138	1.711	7.480
C4	1.700	4.243	3.367	0.253	7.350
C5	3.870	20.439	13.852	16.603	6.700
C6	3.150	6.428	4.232	0.476	9.360
C7	2.360	15.029	19.421	20.428	4.500
C8	1.010	1.880	1.160	3.231	6.390

### 4.7.3 THIRD PRINCIPAL COMPONENT

For classes C1, C2, and C4, local PCA captured the most amount of variance with the third principal component as can be seen in Table 4.22. This is not surprising since the first and second principal component using this technique captured very little of the variance in the data for these classes. For the remaining classes, the AANNs captures the highest amount of variance for the third principal component. It can be seen from Table 4.22 that most of the PCA techniques captures very little variance in the data of each class with the exception of class C7 where the AANN with one hidden layer captures approximately 28% of the remaining variance in the data.

**Table 4.22: Comparison of the amount of variance captured by the third principal component using the five principal component analysis techniques for each class of images**

Class	Conventional LPCA	LPCA using an AANN with 1 hidden layer	NLPCA using an AANN with 1 hidden layer	NLPCA using an AANN with 3 hidden layers	Local Principal Component Analysis
C1	0.650	2.452	1.773	1.308	5.150
C2	0.490	0.983	1.124	0.453	5.480
C3	2.260	1.028	1.744	4.419	3.060
C4	0.360	1.524	1.517	0.129	2.310
C5	1.670	5.337	9.229	5.989	2.980
C6	0.590	0.345	4.840	0.463	2.780
C7	2.750	28.357	9.043	4.318	3.130
C8	1.030	2.438	7.703	0.338	3.740

#### 4.7.4 TOTAL VARIANCE CAPTURED BY THE FIRST THREE PRINCIPAL COMPONENTS

For classes C1, C2, C4 and C7, LPCA using an AANN captures the most amount of variance in the data with the first three principal components as can be seen in Table 4.23. For classes C3 and C5, the AANN with three hidden layers captures the most amount of variance in the data. While for classes C6 and C8, NLPCA using an AANN with one hidden layer captures the most amount of variance in the data. From the results in Table 4.23 it can be seen that PCA via AANNs is able to capture more of the variance in the data than conventional LPCA and local PCA.

**Table 4.23: Comparison of the total amount of variance captured by the first three principal components using the five principal component analysis techniques for each class of images**

Class	Conventional Linear Principal Component Analysis	LPCA using an AANN with 1 hidden layer	NLPCA using an AANN with 1 hidden layer	NLPCA using an AANN with 3 hidden layers	Local Principal Component Analysis
C1	94.310	97.778	97.063	96.152	54.170
C2	95.040	96.926	95.597	91.777	56.010
C3	89.730	97.112	87.232	98.684	54.710
C4	94.690	96.207	93.347	90.605	57.440
C5	85.500	93.891	92.447	94.830	46.220
C6	96.990	97.677	99.618	91.874	55.680
C7	66.200	93.793	79.128	76.588	51.540
C8	93.120	94.296	97.590	93.187	50.670

If the results in Table 4.3 are compared with those of Table 4.7, Table 4.11 and Table 4.15 one would see that the total variance captured by the first three PCs in Table 4.7, Table 4.11 and Table 4.15 are generally higher than those in Table 4.3. These results show that PCA via an AANN gives better results than conventionally used PCA techniques.

## 4.8 SIGNIFICANCE OF A DIFFERENCE BETWEEN CORRELATION COEFFICIENTS OF THE THREE AUTOASSOCIATIVE NEURAL NETWORKS

The significance of a difference between two correlation coefficients is computed by using the z-statistic. The z-statistic is given by

$$z = \frac{Z_1 - Z_2 - \mu_{z_1 - z_2}}{\sigma_{z_1 - z_2}} \quad (4.6)$$

Where  $\mu_{z_1 - z_2} = \mu_{z_1} - \mu_{z_2} = 0$  (4.7)

$$\sigma_{z_1 - z_2} = \sqrt{\left( \frac{1}{n_1 - 3} + \frac{1}{n_2 - 3} \right)} \quad (4.8)$$

And Z is the Fishers Z transformation which is defined by Equation 4.2. Equations 4.6-4.8 are obtained from “Introduction to Exploratory Data Analysis and Empirical Modelling” by Aldrich (1998-1999).

$\mu$  = Mean

$\sigma$  = Standard deviation

$n$  = Number of observations (This value is obtained from Table 4.2 and is based on the test set)

A gauge of the performance of the three AANN models developed in Section 4.5 is given by the correlation coefficient associated with the AANN models. From the correlation coefficient the Fishers Z transformation associated with each AANN model was calculated according to Equation 4.2. The z statistic is calculated between the Fisher Z transformation values associated with any two AANN models. The three different AANN

models were compared relative to one another using the z-statistic at the 0.05 level to determine if there was any significant difference between the performance any two of the AANN models used in this study.

The critical values of z at the 0.05 level were obtained by using a two-tailed test of the normal distribution. The critical values of z at the 0.05 level are 1.96 and -1.96 and at the 0.01 level are 2.25 and -2.25. If the z statistic between any two AANNs is greater than 1.96 or less than -1.96 it means that there is a significant difference between the performance of the two AANN models. If the z statistic is less than 1.96 or greater than -1.96 it means that there is no significant difference between the performance of the two AANN models. The three different AANN models are:

AANN *A* - One hidden layer (Linear Activation Function)

AANN *B* - One hidden layer (Non-Linear Activation Function)

AANN *C* - Three hidden layers (Non-Linear Activation Functions in all three hidden layers)

The three AANN models performance are compared for the extraction of the first principal component, second principal component and the third principal component for each of the eight classes of data. Values in bold in Tables 4.24-4.26 indicates that the z-statistic is greater than the critical values, indicating that there is a significant difference between the performance of the AANN models been compared.

#### **4.8.1 SIGNIFICANCE OF DIFFERENCES BASED ON THE FIRST PRINCIPAL COMPONENT**

Table 4.24 gives the z-statistic between AANNs *A*, *B*, and *C* for the extraction of the first principal component from images of class C1 to C8. Values of the Fishers *Z* transformation used in the calculation of the z-statistic in Table 4.24 are obtained from Tables 4.4, 4.8, and 4.12. The discussion to follow is with reference to Table 4.24.

**Table 4.24: z-Statistics between autoassociative neural networks A, B and C for the extraction of the first principal component from images of classes C1 to C8**

Class	z-Statistic between AANN A and B	z-Statistic between AANN A and C	z-Statistic between AANN B and C
C1	-0.373	-0.990	-0.618
C2	0.419	0.225	-0.194
C3	0.392	<b>-2.178</b>	<b>-2.570</b>
C4	0.501	0.060	-0.441
C5	-1.020	-0.350	-0.248
C6	0.088	-0.008	-0.096
C7	-0.012	-0.067	-0.055
C8	0.400	0.120	-0.280

For images of class C1, C2, C4, C5, C6, C7, C8 there is no significant difference between the performance of the AANNs used for the extraction of the first principal component as can be seen in Table 4.24. For images of class C3 there is a significant difference between the performance of the AANNs A and C and AANNs B and C. This is due to the AANN C having an R-value<sup>17</sup> of 0.92 whereas the AANNs A and B have an R-value of approximately 0.8.

#### 4.8.2 SIGNIFICANCE OF DIFFERENCES BASED ON THE SECOND PRINCIPAL COMPONENT

Table 4.25 gives z-statistic between AANNs A, B, and C for the extraction of the second principal component from images of class C1 to C8. The Fishers Z transformation values used in the calculation of the z-statistic in Table 4.25 is obtained from Tables 4.5, 4.9, and 4.13.

<sup>17</sup> The R-values used in this section are obtained from Section 4.5.

**Table 4.25: z-Statistics between autoassociative neural networks A, B and C for the extraction of the second principal component from images of classes C1 to C8**

Class	z-Statistic between AANN A and B	z-Statistic between AANN A and C	z-Statistic between AANN B and C
C1	0.450	1.725	1.274
C2	1.701	5.193	4.212
C3	3.066	3.678	0.611
C4	0.891	2.278	1.387
C5	1.172	0.300	-0.872
C6	1.738	3.069	1.871
C7	-0.358	-0.485	-0.127
C8	0.558	-0.851	-1.408

For images of class C1, C5, C7, and C8 there is no significant difference between the performance of any of the AANNs as can be seen in Table 4.25. For images of class C2 there is a significant difference between the performance of the AANNs A and C and AANNs B and C. The reason for this is due to AANN C having a low R-value of 0.044 whereas the AANNs A and B have R-values of 0.53 and 0.41 respectively.

For images of class C3 there is a significant difference between the performance of the AANNs A and B and the AANNs A and C. The reason for this is due to the AANN A having a high R-value of approximately 0.8 whereas the AANNs B and C have R-values of 0.36 and 0.23 respectively.

For images of class, C4 there is a significant difference between the performance of the AANNs A and C. AANN C has a low R-value of 0.026 whereas the AANN A has an R-value of 0.46. Therefore, there is a significant difference between AANN models A and C for class C4.

For images of class C6 there is a significant difference between the performance of AANNs *A* and *C*. AANN *C* has a low R-value of 0.056 whereas the AANN *A* has an R-value of 0.71, therefore there is a significant difference between the performance of the AANNs *A* and *C* for class C6.

### 4.8.3 SIGNIFICANCE OF DIFFERENCES BASED ON THE THIRD PRINCIPAL COMPONENT

Table 4.26 gives z-statistic between AANNs *A*, *B*, and *C* for the extraction of the third principal component from images of class C1 to C8. The Fishers Z transformation values used in the calculation of the z-statistic in Table 4.26 are obtained from Tables 4.6, 4.10, and 4.14. The discussion to follow is with reference to Table 4.26.

*Table 4.26: z-Statistics between autoassociative neural networks A, B and C for the extraction of the third principal component from images of classes C1 to C8*

Class	z-Statistic between AANN A and B	z-Statistic between AANN A and C	z-Statistic between AANN B and C
C1	1.998	3.459	1.461
C2	0.439	2.086	1.647
C3	0.637	-3.237	-3.874
C4	0.540	1.421	0.880
C5	-0.486	-0.405	0.081
C6	-6.563	0.332	6.895
C7	2.931	3.468	0.537
C8	-4.454	1.685	6.140

For images of class C1 there is a significant difference between the performance of the AANNs *A* and *B* and *B* and *C*. The AANNs *C* and *B* have low R-values of 0.025 and 0.38 respectively whereas the AANN *A* has an R-value of 0.52. Therefore, there is a significant difference between the performance of the AANNs *A* and *C* and *A* and *B* for class C1.

For images of class, C2 there is a significant difference between the performance of the AANNs *A* and *C*. AANN *C* has a low R-value of 0.052 whereas the AANN *A* has an R-value of 0.24. Thus, there is a significant difference between the performance of the AANNs *A* and *C* for class C2.

For images of class C3 there is a significant difference between the performance of the AANNs *A* and *C* and *B* and *C*. The AANN *C* has a high R-value of 0.77 whereas the AANNs *A* and *B* have low R-values of 0.24 and 0.20 respectively. Therefore there is a significant difference between the performance of the AANNs *A* and *C* and *B* and *C* for class C3. For images of class C4 and C5, there is no significant difference between the performance of any of the AANNs.

For images of class C6 there is a significant difference between the performance of the AANNs *A* and *B* and *B* and *C*. The AANN *B* has a high R-value of 0.937 whereas the AANNs *A* and *C* have low R-values of 0.13 and 0.054 respectively. Therefore there is a significant difference between the performance of the AANNs *A* and *B* and *B* and *C* for class C6.

For images of class C7 there is a significant difference between the performance of the AANNs *A*, *B*, and *A* and *C*. The AANN *A* has a high R-value of 0.82 whereas the AANNs *B* and *C* have low R-values of 0.30 and 0.16 respectively. Therefore, there is a significant difference between the performance of the AANNs *A* and *B* and *A* and *C* for class C7.

For images of class C8 there is a significant difference between the performance of the AANNs *A* and *B* and *B* and *C*. The AANN *B* has a high R-value of 0.76 whereas the

AANNs *A* and *C* have low R-values of 0.30 and 0.05 respectively, thus there is a significant difference between the performance of the AANNs *A* and *B* and *B* and *C* for class C8.

## 4.9 CLASSIFICATION OF FROTH IMAGES

The classification of froth structures in flotation plants is a difficult task due to the flotation process being complex and poorly understood. The classification of froth structures has been investigated by a number of researches such as Woodburn et al. (1989), Symonds and De Jager (1992), Moolman (1995) and Botha (1999) and many more.

Moolman (1995) classified copper and platinum flotation froth images on the basis of statistical features extracted by means of digital image processing techniques. Textural features were extracted by the spatial grey level dependence matrix and the neighbouring grey level dependence matrix from froth images and used for classification by two neural network architectures. The two neural network architectures used for classification were a learning vector quantization neural network and a self-organising mapping neural network.

Moolman (1995) went further on to show that the application of neural networks for image compression is relatively simple and does not require an extensive background on image compression techniques. Moolman (1995) proposed a technique for image compression by a neural network that reduces the size of the feature space and results in classification afterwards. Moolman (1995) has shown that a classification system, which consists of backpropagation and self-organising mapping neural networks, has superior classification ability compared to a system that includes conventional image processing techniques.

The approach used in this study for classification of froth images is similar in methodology to the approach presented by Moolman (1995) which used a

backpropagation AANN and self organising mapping neural network. The main difference in the approach used in this study with that of Moolman (1995) is that:

1. Moolman (1995) used an AANN with three layers to extract features from froth images whereas in this study an AANN with three as well as five layers are used to extract features from froth images.
2. Moolman (1995) extracted four features from images (256\*256 pixels) by simultaneously extracting the features whereas the approach used in this study extracts three features from images (31\*31 pixels) sequentially (Refer to Section 3.4.1 and 3.4.2 for a discussion on the extraction of features simultaneously and sequentially).
3. Moolman (1995) classified features extracted from images by using a self-organising mapping neural network followed by single hidden layer backpropagation neural network. In this study features extracted from images are classified using two types of classifiers. They are:
  - Probabilistic neural network classifier and a
  - Feedforward backpropagation neural network classifier

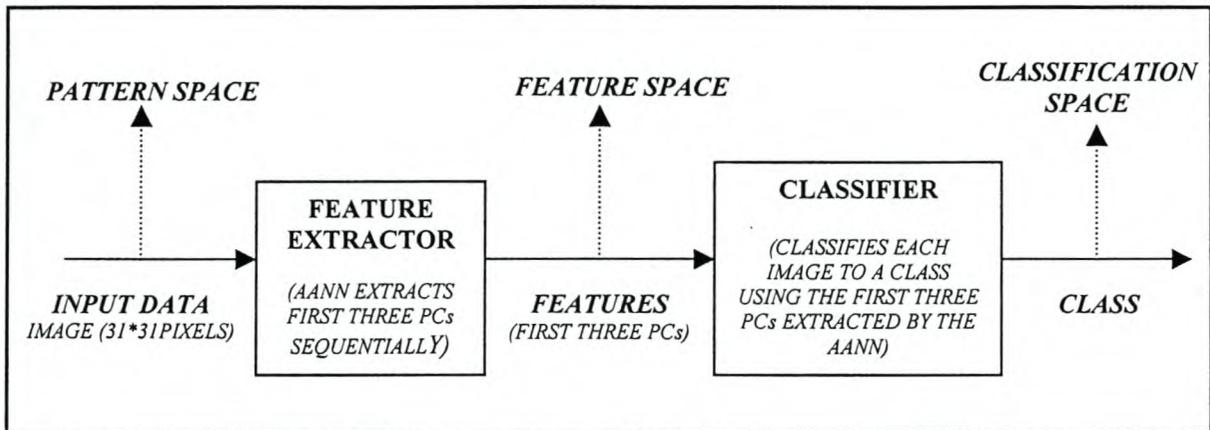
The reason for using two types of classifiers is to evaluate which neural network classifier architecture (feedforward backpropagation or radial basis neural networks) is more powerful in classifying features from images of the surface froth of a laboratory flotation cell.

Features extracted from froth flotation images, which are described in Section 4.1, are used in this study to classify to which class of froth an image belongs. Images described in Section 4.1 were categorised into eight classes. The methodology of sequential PCA via an AANN was used to extract the first three principal components from each image.

Two types of neural network classifiers have been developed in this section. The first type of classifier is based on radial basis networks and is called a probabilistic neural

network classifier. The second type of classifier is developed via a feedforward backpropagation neural network and will be called a feedforward backpropagation neural network classifier.

Figure 4.8 illustrates the process for classification of an image. Three principal components (Features) are extracted from an image using a sequential autoassociative neural network. The three principal components are then used as inputs to the neural network classifier<sup>18</sup> which classifies to which class of images an image belongs.



*Figure 4.8: Process for an image to be classified to a class*

#### 4.9.1 INPUT AND TARGET DATA USED FOR TRAINING AND TESTING THE CLASSIFIERS

The essential features of each image were extracted via a sequential AANN as discussed in Section 4.5. The features extracted from each image were captured in three principal components. Table 4.1 gives the number of images used for training and testing the sequential AANN models for each class of images. The input matrix to train the classifier was composed of an  $m*q$  matrix where  $m$  is the number of observations in the matrix and

<sup>18</sup> The neural network classifier could either be a probabilistic neural network classifier or a feedforward backpropagation neural network classifier

$q$  is the number of variables. The value of  $q$  is three, which corresponds to the number of principal components extracted from each image. The value of  $m$  is 2960. This value is obtained from Table 4.27 and is the sum of all the images used for training the AANN models for the eight classes of images. Therefore, the total number of images used for training the classifiers is 2960. The input matrix was of size  $(2960*3)$ . After the classifiers were trained, they were validated on the total number of images used to validate the AANN models for each class of images. A test set of size 752 images was used to validate the classifier models as can be seen in Table 4.27.

*Table 4.27: Number of images used to train and test the classifiers*

Class	Number Of Images Used For Training The AANN Models	Number Of Images Used For Testing The AANN Models	Total Number Of Images In Each Class
C1	600	232	832
C2	600	232	832
C3	280	40	320
C4	330	54	384
C5	280	40	320
C6	215	41	256
C7	165	27	192
C8	490	86	576
Total	2960	752	3712

*NB: Figures to follow in this chapter are plots based on images used for testing the AANN models*

The classifiers had eight outputs. Each output represented one class of images. Output one represents images of class C1 and output two images of class C2 etc. During training of the classifier, each exemplar<sup>19</sup>, which is of size  $(1*3)$ , has a corresponding target. If the

<sup>19</sup> Exemplar here refers to the three principal components which represent a single image  $(1*961)$  pixels).

input into the classifier is an image of class C1 then output one of the classifier is given a one while the remaining outputs are given an output of zero. If the input into the classifier is an image of class C3 then output three is given a one and the remaining outputs are given a zero etc. In other words, each image is coded using zeros and ones. Table 4.28 describes what has just being discussed above.

**Table 4.28: Coding of each class of images to be used as the target output during training of the classifier model**

Outputs Of Classifier	Target Of An Image From Class C1	Target Of An Image From Class C2	Target Of An Image From Class C3	Target Of An Image From Class C4	Target Of An Image From Class C5	Target Of An Image From Class C6	Target Of An Image From Class C7	Target Of An Image From Class C8
Output 1	1	0	0	0	0	0	0	0
Output 2	0	1	0	0	0	0	0	0
Output 3	0	0	1	0	0	0	0	0
Output 4	0	0	0	1	0	0	0	0
Output 5	0	0	0	0	1	0	0	0
Output 6	0	0	0	0	0	1	0	0
Output 7	0	0	0	0	0	0	1	0
Output 8	0	0	0	0	0	0	0	1

Principal components were extracted for all eight classes of images via the three types of AANNs in Section 4.5. Three different probabilistic neural network classifiers as well as three different feedforward backpropagation neural network classifiers were designed for each set of principal components extracted via the three AANN models. In total six classifiers have being developed as can be seen in Table 4.29.

*Table 4.29: Number of classifiers developed*

Principal Components Extracted with	AANN With 3 Hidden Layers (Non-Linear Activation function )	AANN With 1 Hidden Layer (Non-Linear Activation function )	AANN With 1 Hidden Layer (Linear Activation function)
PNN Classifiers	Classifier 1	Classifier 2	Classifier 3
Feedforward Backpropagation NN Classifiers	Classifier 4	Classifier 5	Classifier 6

## 4.9.2 PROBABILISTIC NEURAL NETWORK CLASSIFIER

Radial basis neural networks may require more neurons than standard feedforward backpropagation neural networks, but often they can be designed in a fraction of the time it takes to train standard feedforward neural networks. They work best when many training vectors are available (Chen, Cowan, Grant, 1991).

A probabilistic neural network is one variant of radial basis neural networks and can be used for a number of different classification problems. When an input is presented to a probabilistic neural network, the first layer computes the distances from the input vector to the training input vectors, and produces a vector whose elements indicate how close the input is to a training input. The second layer sums these contributions for each class of inputs to produce as its net output a vector of probabilities. Finally, a compete activation function on the output of the second layer picks the maximum of these probabilities, and produces a one for that class and a zero for the other classes.

### 4.9.2.1 Classifier 1-Classifies Non-Linear Principal Components extracted by the AANN model with Three Hidden Layers

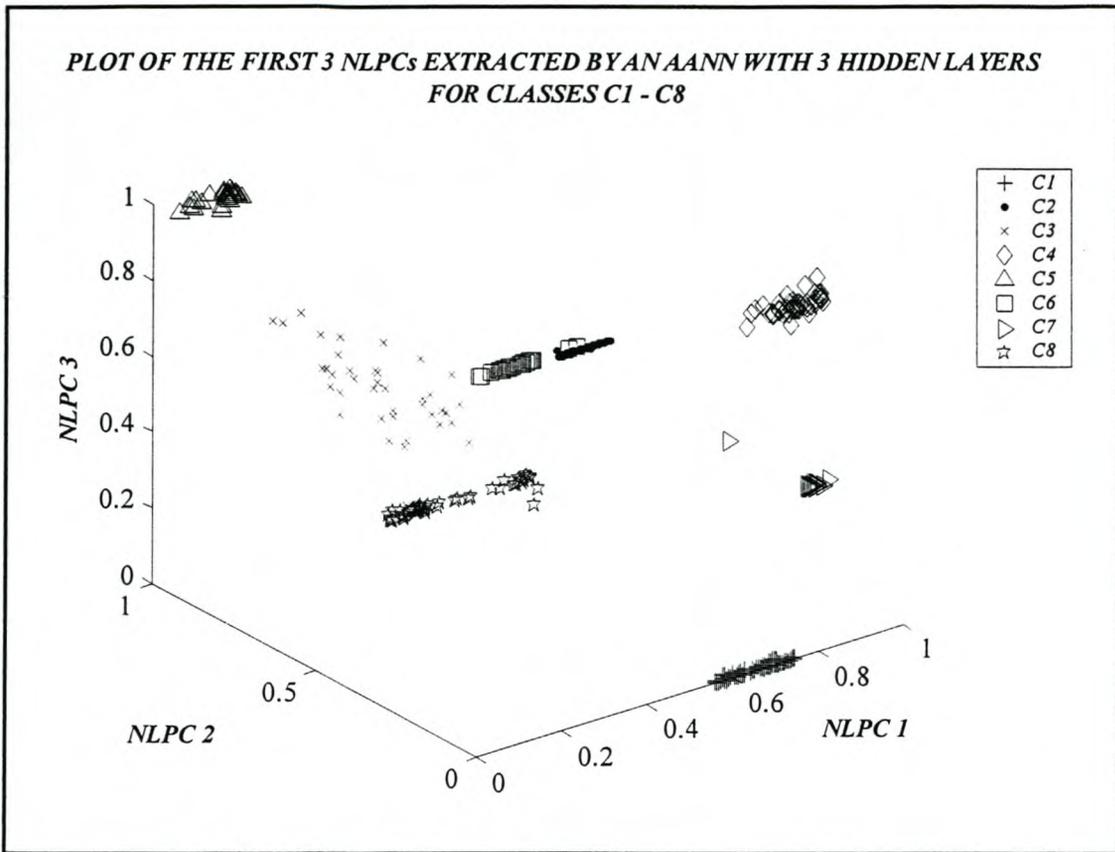
The first probabilistic neural network classifier was trained with the non-linear principal components obtained from the AANN with three hidden layers, which used non-linear

activation functions in the nodes of the hidden layers. The probabilistic neural network was trained on a Pentium 2, 400 MHz computer with 128 KB of ram. The total number of training exemplars available for training the probabilistic neural network classifier was 2960 as can be seen in Table 4.27. Of the 2960 only 2100 exemplars were used for training due to the large computational requirements required by the probabilistic neural network classifier and the limitation of the random access memory available on the computer used to train the network. Of the 2100 exemplars used to train the probabilistic neural network classifier, 30 exemplars were incorrectly classified. The percentage of correctly classified images during training was approximately 99% as can be seen in Table 4.30. When the probabilistic neural network classifier was presented with unseen data such as the test set which comprised of 752 exemplars it incorrectly classified 10 exemplars thus giving a correct classification rate of approximately 99% as can be seen in Table 4.30.

*Table 4.30: Percentage of correctly classified non-linear principal components of the training and test data sets by the probabilistic neural network classifier 1*

<b>Data Sets</b>	<b>Total Number Of Exemplars</b>	<b>Number Of Correctly Classified Exemplars</b>	<b>Number Of Incorrectly Classified Exemplars</b>	<b>Correctly Classified Exemplars (%)</b>
<b>Training</b>	2100.00	2070.00	30.00	98.57
<b>Test</b>	752.00	742.00	10.00	98.67

A possible reason for the probabilistic neural network classifier having such a high percentage of correctly classified exemplars during training and testing could be attributed to the first three non-linear principal components of each class of images. The plot of the first three non-linear principal components of each class of images clearly distinguishes each class of images from one another as can be seen in Figure 4.9 with the exception of class C2 and C6 where two exemplars of class C6 lies in the cluster of C2.



*Figure 4.9: Plot of the first three non-linear principal components, for images from the test set, for classes C1-C8 which were obtained from an AANN with three hidden layers (non-linear activation functions)*

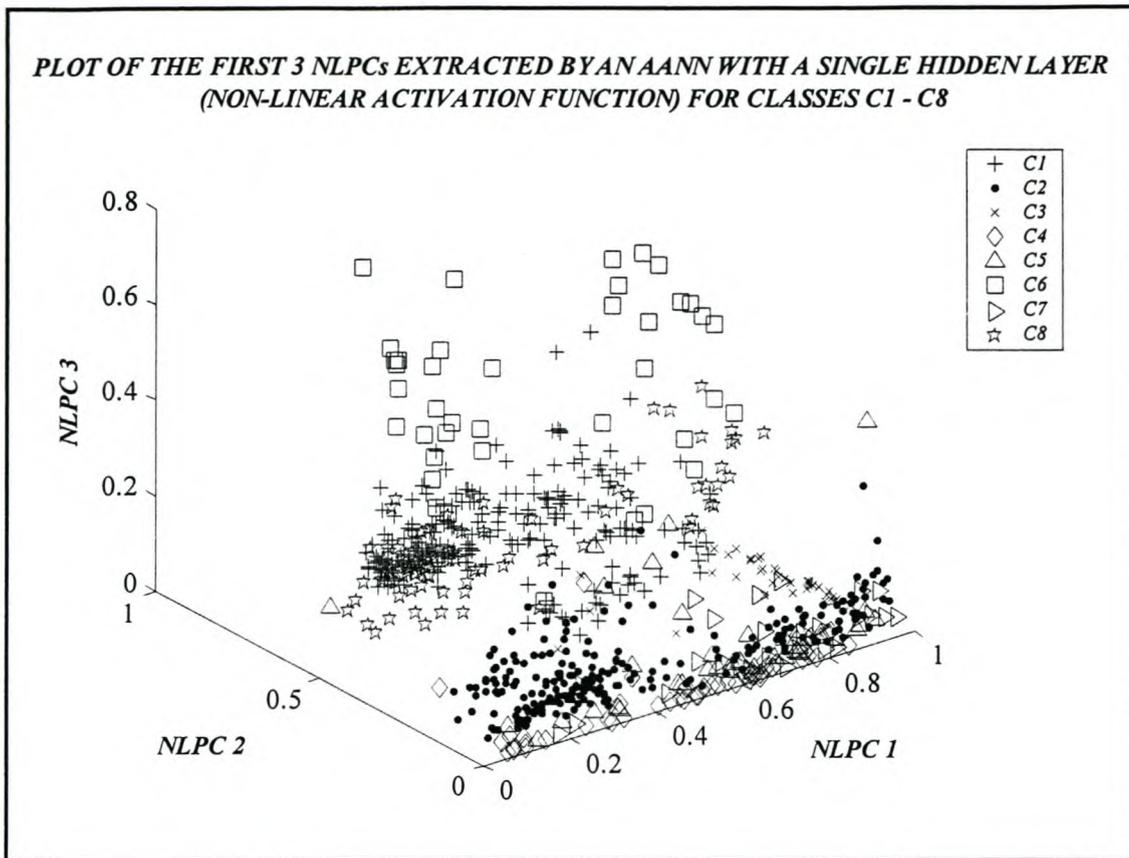
#### 4.9.2.2 Classifier 2 - Classifies Non-Linear Principal Components extracted by the AANN model with One Hidden Layer

The second probabilistic neural network classifier was trained with the non-linear principal components obtained from the AANN with one hidden layer, which used a non-linear activation function in the single node of the hidden layer. 2100 exemplars were used to train the classifier of which 460 exemplars were incorrectly classified by the classifier. The percentage of correctly classified exemplars was approximately 78% as can be seen in Table 4.31.

*Table 4.31: Percentage of correctly classified non-linear principal components of the training and test data sets by the probabilistic neural network classifier 2*

Data Sets	Total Number Of Exemplars	Number Of Correctly Classified Exemplars	Number Of Incorrectly Classified Exemplars	Correctly Classified Exemplars (%)
Training	2100.00	1640.00	460.00	78.10
Test	752.00	596.00	156.00	79.26

When the test set which comprised of 752 exemplars was presented to the classifier it incorrectly classified 156 exemplars of the 752. 79% of the exemplars presented to the classifier were correctly classified as can be seen in Table 4.31. The possible reason for classifier two not performing as well as classifier one could be attributed to the non-linear principal components extracted by the AANN with one hidden layer (Non-linear activation function). The non-linear principal components extracted by this network does not separate each class of images into distinct clusters, as was the case with the non-linear principal components extracted via the AANN with three hidden layers (Non-linear activation function). This can be seen when Figure 4.9 and Figure 4.10 are compared. In Figure 4.9 all classes of images are distinctly separated into clusters besides that of classes C6 and C2 which seem to slightly overlap with one another. In Figure 4.10 on the other hand the clusters are not clearly separated but more fuzzy with classes C1 and C8 overlapping, as well as classes C2, C4, C5, C7 all occupying the same area in the three dimensional plot. Thus classifier twos performance has decreased by approximately 20% when compared to that of classifier one which gave an approximately 99% correct classification rate.



*Figure 4.10: Plot of the first three non-linear principal components, for images from the test set, for classes C1-C8 which were obtained from an AANN with a single hidden layer (non-linear activation function)*

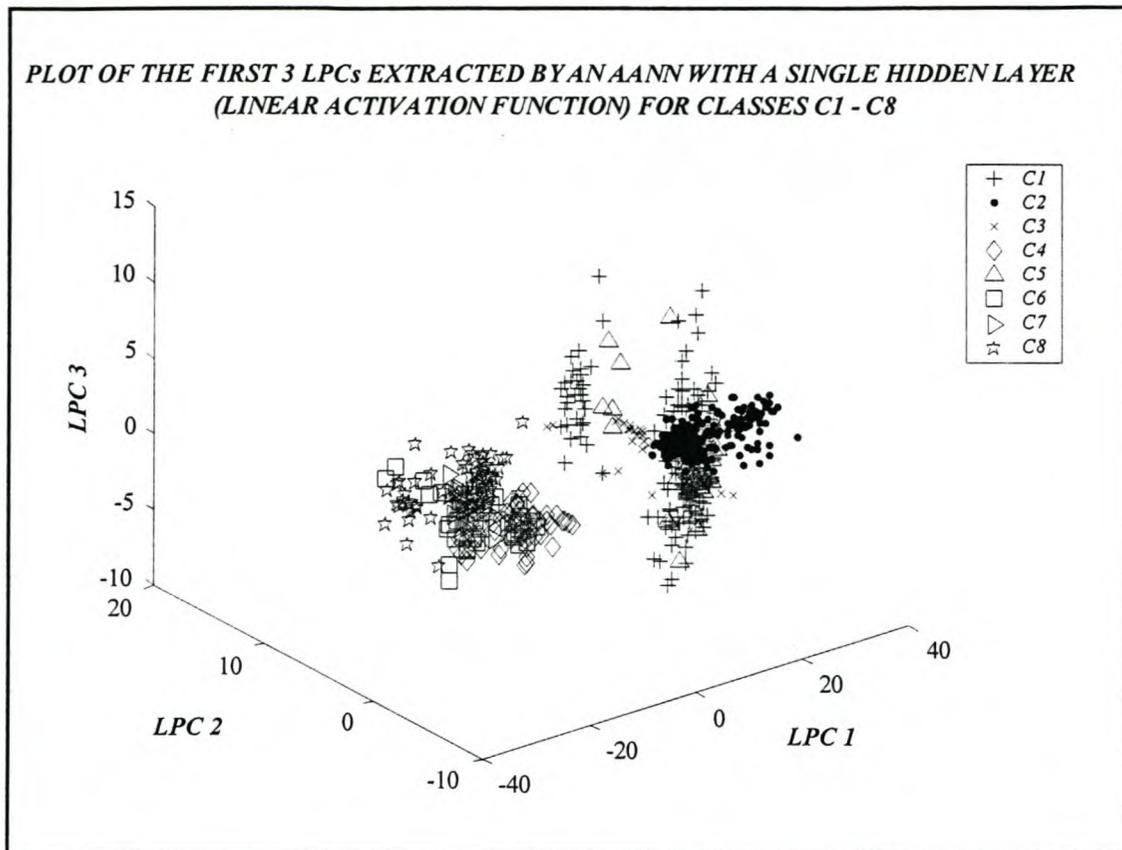
#### 4.9.2.3 Classifier 3 - Classifies Linear Principal Components Extracted by the AANN model with One Hidden Layer

The third probabilistic neural network classifier was trained with the linear principal components obtained from the AANN with one hidden layer, which used a linear activation function in the single neuron of the hidden layer. 2100 exemplars were used to train the classifier of which 197 exemplars were incorrectly classified by the classifier. The percentage of correctly classified exemplars was approximately 91% as can be seen in Table 4.32.

**Table 4.32: Percentage of correctly classified linear principal components of the training and test data sets by the probabilistic neural network classifier 3**

<b>Data Sets</b>	<b>Total Number Of Exemplars</b>	<b>Number Of Correctly Classified Exemplars</b>	<b>Number Of Incorrectly Classified Exemplars</b>	<b>Correctly Classified Exemplars (%)</b>
<b>Training</b>	2100.00	1903.00	197.00	90.62
<b>Test</b>	752.00	744.00	8.00	98.94

When the test set which was compromised of 752 exemplars was presented to the classifier it incorrectly classified 8 exemplars of the 752. 99% of the exemplars presented to the classifier were correctly classified as can be seen in Table 4.32.



**Figure 4.11: Plot of the first three linear principal components for images from the test set for classes C1-C8 which were obtained from an AANN with a single hidden layer (Linear activation function)**

The reason for classifier three performing as well as classifier one could be due to the linear principal components extracted by the AANN with one hidden layer (Linear activation function). The linear principal components extracted by this AANN does not separate each class of images into distinct clusters as can be seen in Figure 4.11 but rather the linear principal components extracted for each class of images are very sharply clustered together. This is contrary to the way the AANN with one hidden layer (non-linear activation function) extracted the non-linear principal components as can be seen in Figure 4.10. The reason why Figure 4.11 differs from Figure 4.9 is that the principal components describing each class of images in Figure 4.11 are not as tightly clustered as the principal components describing each class of images in Figure 4.9. Even though

there seems to be overlapping of classes in Figure 4.11 the classifier is able to distinguish between the different classes of images. This could be a possible reason why classifier 1 and classifier 3 perform better than classifier 2.

### **4.9.3 FEEDFORWARD BACKPROPAGATION NEURAL NETWORK CLASSIFIER**

The first three classifiers developed are based on probabilistic neural networks, which is a variant of radial basis neural networks. The following section develops a classifier using feedforward backpropagation neural networks. The architecture used to design the feedforward backpropagation neural network classifier is simple. The classifier consisted of three inputs and eight outputs. The three inputs being the three principal components extracted for each image in Sections 4.5 and the eight outputs being the eight different classes of images. The network consisted of a single hidden layer and an output layer. The hidden layer and the output layer were connected to biases. The number of nodes in the hidden layer was varied between ten and twenty nodes. The nodes in the output and hidden layer consisted of non-linear activation functions. The highly optimised Levenberg-Marquardt algorithm could not be used to train the classifier due to the large memory requirements needed when training the classifier with a large data set. The next best alternative used to train the classifier was the resilient backpropagation algorithm.

The number of exemplars used to train the classifier was 2960, as can be seen in Table 4.27 and the number of exemplars used to validate the feedforward backpropagation neural network classifier model was 752. When developing the classifiers only 2100 exemplars of the total 2960 exemplars were used to train the probabilistic neural network classifier. This was due to the large computational requirements required to train the probabilistic neural network classifier and the limitation of the amount of RAM available. A similar problem was experienced when trying to train the feedforward backpropagation neural network classifier with the Levenberg-Marquardt algorithm but this was overcome by using the resilient backpropagation algorithm which allowed the use of the full 2960

exemplars for training. The outputs during training were coded as in Table 4.28. Three feedforward backpropagation neural network classifiers were developed for each different set of principal components extracted via the three different AANN models.

#### 4.9.3.1 Classifier 4 - Classifies Non-Linear Principal Components extracted by the AANN model with Three Hidden Layers

The fourth classifier, which is a feedforward backpropagation neural network classifier, was trained with the non-linear principal components obtained from the AANN with three hidden layers. The classifier was trained with 2960 exemplars of which it incorrectly classified five. The percentage of correctly classified exemplars was approximately 100%. When the classifier was presented with the test set it correctly classified all exemplars in the test set as can be seen in Table 4.33.

*Table 4.33: Percentage of correctly classified non-linear principal components of the training and test data sets by the feedforward backpropagation neural network classifier 4*

Data Sets	Total Number Of Exemplars	Number Of Correctly Classified Exemplars	Number Of Incorrectly Classified Exemplars	Correctly Classified Exemplars (%)
Training	2960.00	2955.00	5.00	99.83
Test	752.00	752.00	0.00	100.00

The reasons for such good results is due to the first three non-linear principal components of each class of images forming distinct clusters as was discussed in Section 4.9.2.1 and as can be seen in Figure 4.9.

#### 4.9.3.2 Classifier 5 - Classifies Non-Linear Principal Components extracted by the AANN model with One Hidden Layer

The fifth classifier, which is a feedforward backpropagation neural network classifier, was trained with the non-linear principal components obtained from the AANN with one hidden layer (non-linear activation function). 2960 exemplars were used to train the classifier of which 249 exemplars were incorrectly classified by the classifier. The percentage of correctly classified exemplars was approximately 92% as can be seen in Table 4.34.

*Table 4.34: Percentage of correctly classified non-linear principal components of the training and test data sets by the feedforward backpropagation neural network classifier 5*

Data Sets	Total Number Of Exemplars	Number Of Correctly Classified Exemplars	Number Of Incorrectly Classified Exemplars	Correctly Classified Exemplars (%)
Training	2960.00	2711.00	249.00	91.60
Test	752.00	556.00	196.00	74.00

The classifier did not perform as well when presented with the test set. Of the 752 test exemplars presented to the classifier it incorrectly classified 196 of the exemplars giving an overall correct classification of 74%. The reason for the poorer performance of classifier 5 has been explained in the Section 4.9.2.2.

### 4.9.3.3 Classifier 6 - Classifies Linear Principal Components extracted by the AANN model with One Hidden Layer

The sixth classifier, which is a feedforward backpropagation neural network classifier, was trained with the linear principal components obtained from the AANN with one hidden layer (Linear activation function). 2900 exemplars were used to train the classifier of which 1074 exemplars were incorrectly classified by the classifier. The percentage of correctly classified exemplars was approximately 64% as can be seen in

Table 4.35. When the classifier was presented with the test set it, achieve an overall correct classification rate of 63%.

*Table 4.35: Percentage of correctly classified linear principal components of the training and test data sets by the feedforward backpropagation neural network classifier 6*

Data Sets	Total Number Of Exemplars	Number Of Correctly Classified Exemplars	Number Of Incorrectly Classified Exemplars	Correctly Classified Exemplars (%)
Training	2960.00	1886.00	1074.00	64.00
Test	752.00	473.00	279.00	63.00

#### 4.9.4 PROBABILISTIC NEURAL NETWORK CLASSIFIERS VS FEEDFORWARD BACKPROPAGATION NEURAL NETWORK CLASSIFIERS

A probabilistic neural network is guaranteed to converge to a bayesian classifier providing it is given enough training data. These networks generalise well. The probabilistic neural networks have many advantages but suffers from one major disadvantage. They are slower to operate because they use more computation than other kinds of networks to do their classification (Chen et al., 1991).

The results of the two different types of neural network architectures used for classification are presented in Table 4.36. The heading in Table 4.36 for example classifier 1, 3 hidden layers refers to the classifier number and 3 hidden layers refers to which AANN model which was used to obtain the principal components which were used as inputs into the classifier.

*Table 4.36: Comparison in the performance of probabilistic neural network classifiers and feedforward backpropagation neural network classifiers*

Data Sets	Probabilistic Neural Network Classifier			Feedforward Neural Network Classifier		
	Classifier 1 NLPCs from AANN with 3 Hidden Layers	Classifier 2 NLPCs from AANN with 1 Hidden Layer	Classifier 3 LPCs from AANN with 1 Hidden Layer	Classifier 4 NLPCs from AANN with 3 Hidden Layers	Classifier 5 NLPCs from AANN with 1 Hidden Layer	Classifier 6 LPCs from AANN with 1 Hidden Layer
	Correctly Classified Exemplars (%)	Correctly Classified Exemplars (%)	Correctly Classified Exemplars (%)	Correctly Classified Exemplars (%)	Correctly Classified Exemplars (%)	Correctly Classified Exemplars (%)
Training	98.57	78.10	90.62	99.83	92.00	64.00
Test	98.67	79.26	98.94	100.00	74.00	63.00

As can be seen in Table 4.36 classifier 1 and classifier 4 performs equally well on the training and test data sets. Classifier 5 outperforms classifier 2 with regard to the training data sets as classifier 5 obtains a 92% correct classification when compared with the 72% correct classification by classifier 2. But classifier 2 performs better than classifier 5 when presented with the test data set with classifier 2 having a 79% correct classification and classifier 5 having a 72% correct classification. Classifier 3 outperforms classifier 6 with regard to both the training and test data sets as can be seen in Table 4.36.

From the above results it seems as though the probabilistic neural network classifiers are more robust and able to generalise better than the feedforward backpropagation neural network classifiers. The feedforward backpropagation neural network classifier performs just as well as the probabilistic neural network classifier when clusters are distinctly separated from one another as is the case in Figure 4.9 but when clusters are close to or overlap with one another as in Figure 4.10 and Figure 4.11 their performance begins to deteriorate as can be seen in Table 4.36.

It is therefore recommended that the probabilistic neural network classifiers be used to distinguish between different classes of images obtained from flotation processes.

## 4.10 SUMMARY

The technique of PCA was illustrated on image data using conventional LPCA, local PCA and PCA using AANNs. Conventional LPCA was able to extract slightly more of the variance in the image data with the first linear principal component than the other PCA techniques used in this study. The AANNs on the other hand were able to extract more of the variance in the image data with the first three principal components than conventional LPCA and local principal component analysis. The AANN with one hidden layer (Linear activation function) captured the most amount of variance with the first three linear principal components for classes C1, C2, C4 and C7 while the AANN with one hidden layer (Non linear activation function) captured the most amount of variance with the first three non-linear principal components for classes C6 and C8. The AANN with three hidden layers (Non-linear activation function) captured the most amount of variance with the first three non-linear principal components for classes C3 and C5.

Local PCA performed poorly in the extraction of piecewise linear principal components from the eight classes of image data relative to the other techniques. This was due to the data sets been of too high a dimensionality. The high dimensionality (approximately 1000 dimensions) of the data sets has led to them been poorly clustered. Present clustering techniques can create meaningfully clusters for data sets up to approximately 25 dimensions. For data sets with higher dimensionality clustering techniques do not yield meaningful results. Therefore the poor clustering of the data has led to local PCA not being able to capture as much of the variance in the data compared to the other PCA techniques. It was shown via a simple example how local PCA is able to capture more of the variance in data than conventional LPCA.

The total amount of variance captured for each class of images by the first three piecewise linear principal components were all below 60% using local PCA due to the reasons explained above. The other techniques on average captured over 90% of the variance in the data with the first three principal components for each class of images.

From these quantitative results, it is difficult to say whether conventional LPCA or PCA using AANNs is better in extracting features from digital images of the froth.

To better understand which PCA technique performs better in extracting features from froth flotation images one has to look at the qualitative results which are the three dimensional plots of the first three principal components of each class of images. From these plots, it was concluded that the AANN with three hidden layers extracted non-linear features from each class of images, which defined them as unique clusters in three-dimensional space. This was not the case with the AANNs with one hidden layer and conventional PCA. These techniques were not able to extract features which characterised each class of images as unique clusters but rather as clusters which were difficult to differentiate between in three dimensional space. Therefore, though the AANN with three hidden layers captures one or two percent less variance in the data than the other techniques it is able to extract non-linear features which characterise each class of images uniquely. This is due to the mapping and demapping layers associated with an AANN with three hidden layers which is able to map non-linear features more accurately than the other PCA techniques.

One of the drawbacks of using an AANN with three hidden layers to extract features from froth images is that it is difficult to train these networks. This is a possible reason for the AANNs not being able to extract as much of the variance in the data as conventional linear principal component analysis. The AANNs were trained using gradient descent backpropagation training algorithms. These algorithms tend to get caught in local minima in the error surface therefore not been able to reach the global minima in the error surface resulting in the AANN not been optimally trained. The availability of over 2 GHz CPUs and the optimisation of backpropagation algorithms will not totally solve the problem of these algorithms been caught in local minima in the error surface. One of the solutions to the above-mentioned problem is to use optimisation algorithms which are non-gradient based such as genetic algorithms. Another way of overcoming the above mentioned problem has been suggested by Tan & Mavrovouniotis (1995). They have suggested the use of an input training neural network. An input

training neural network can perform all the functions of an AANN and in addition is more efficient in reducing data dimensionality than training an AANN. In light of the above it is still recommended that an AANN with three hidden layers be used for the extraction of features from froth images due to them extracting features which are more unique than the other techniques.

The features extracted by the AANNs were used to classify to which class of froth an image belonged. Two neural network classifier architectures were used for the classification of features, they were the probabilistic neural network classifier and feedforward backpropagation neural network classifier. Both classifiers achieved approximately 100% correct classification on the training and test sets of features extracted by the AANN with three hidden layers. For features extracted by the AANN with one hidden layer (Non-linear activation function) and one hidden layer (Linear activation function) a correct classification of approximately 80% and 98% was achieved respectively based on the test sets by the probabilistic neural network classifier whereas the feedforward backpropagation neural network classifier only achieved 74% and 63% respectively. The performance of these classifiers further indicate that the AANN with three hidden layers extracts non-linear features which are unique and can be accurately classified. Although it is more difficult to train, the probabilistic neural network classifier from a computational viewpoint it is still recommended to be used to classify froth images based on the non-linear features extracted by the AANN with three hidden layers.

There are a number of practical implications from the findings presented in this chapter. The results have shown that AANN is an attractive alternative to feature extraction compared to conventional methods. This has been proved on a laboratory scale and now warrants the investigation to be moved onto a higher level. The advantage of using a neural network approach is that no assumptions regarding the structure of the relationship of the parameters need to be made. Feature extraction using AANNs can be implemented in an adaptive manner, so that changes in the characteristics of processes are taken into account, which is of importance in the flotation industry. These AANN models can be used in conjunction with a number of analysis techniques to analyse the process that is

modelled, thereby turning the available data into knowledge. The powerful classification ability of probabilistic neural networks can be used to detect different classes of froth thereby assisting the operator in understanding the process.

## CHAPTER 5

# MULTIVARIATE STATISTICAL PROCESS CONTROL

The last twenty years have seen considerable shifts in world economics. One of the repercussions in the manufacturing sector has been increased competitiveness. The need to achieve consistent production at lower costs has never been greater to ensure retention of existing market share. Achievement of such an objective has resulted in a significant increase in the industrial application of statistical quality assurance techniques.

One area of rapidly growing interest for the monitoring of manufacturing processes is that of MSPC. The technique of MSPC is equally applicable to continuous and batch processes and has been investigated by a number of researchers such as Lane (2001), Martin (1996), Macgregor (1995), Mason (1998), Riach (1996), Yoon (2000) and Nomikos (1994). The primary objectives of MSPC are to provide early warnings in changes in process behaviour and the assured manufacture of consistent production.

Additionally through a deeper understanding of the process a reduction in raw materials usage and levels of rework and waste are achievable, and as a consequence a minimisation of the impact of the process on the environment. These objectives can be

realised through the monitoring of the performance of the manufacturing process over time and ensuring that it remains in a “state of statistical control”. When the process moves outside the desired operating region, it is regarded as being “not in a state of statistical process control” and it is necessary to warn plant operational personnel that a change in process behaviour has occurred, as well as provide them with appreciate diagnostic information about the process variables causing, or reflecting, the problem, thus enabling corrective action to be implemented.

On line monitoring of chemical process performance is extremely important for plant safety and good product quality. When analytical sensors are unavailable, a mathematical representation of a process can be used for the monitoring task. The representation of a process can take two forms:

- A first principle model or an
- Empirical model based on historical data.

Many factors such as high process non-linearity, high dimensionality, and the complexity of a process, can make the development of a first principle model very difficult. As an alternative, empirical modelling approaches that are basically data-driven multivariate statistical methods have attracted much interest by chemical engineers. These approaches are based on the theory of statistical process control, under which the behaviour of a process is modelled using data obtained when the process is operating well and in a state of control.

Future unusual events are detected by referencing the measured process behaviour against this model. Kresta et al. (1991) has proposed a detailed procedure for multivariate monitoring, where PCA is used to model the normal process data. Because of the favourable features of PCA, their method can handle high dimensional and correlated process variables. The key point to their approach is to use PCA to compress normal process data and extract information by projecting the data onto a low dimensional score space. New MSPC charts, which are similar to conventional Shewart charts have been

developed for monitoring in the low dimensional space. Simulation results have shown that multivariate methods are simple and powerful.

The objectives of MSPC are directly related to the control of industrial flotation plants, where these processes are controlled suboptimally owing to high personnel turnover, lack of fundamental understanding of plant dynamics, in-accuracy or unreliability of manual control systems, etc. As a result, optimal control of flotation plants is not usually maintained especially where incipient erratic behaviour in the plant is difficult to detect. The implementation of MSPC in flotation plants will overcome the above mentioned problems by been able to detect process deviations in real-time. Thus, the use of MSPC will have a significant impact on the improvement of the monitoring of the plant. This will therefore assure consistent production at lower costs.

Two types of models are developed for MSPC of a flotation process. The first type of model is developed using conventional LPCA. The second type of model is developed using LPCA via AANNs and NLPCA via AANNs. The two models are developed using historical data of a flotation process, which represents the normal operating conditions of the process. Future behaviour of the flotation process can be referenced against these “in-control” models.

Two sets of flotation data will be used in this study to illustrate the usefulness of MSPC in flotation processes. The first set of data used to illustrate the technique of MSPC is the data described in Section 4.1.1 and will be referred to as Case 1. The second set of data used to illustrate the technique of MSPC will be referred to as Case 2.

## 5.1 THEORY BEHIND THE DEVELOPMENT OF MULTIVARIATE STATISTICAL PROCESS CONTROL CHARTS BASED ON PRINCIPAL COMPONENT MODELS

### 5.1.1 CONVENTIONAL LINEAR PRINCIPAL COMPONENT MODELS

PCA is a tool for data compression and information extraction. Conventional LPCA finds linear combinations of variables that describe major trends in a data set. Conventional LPCA can be used to build low dimensional models for the analysis, monitoring of process operations is now well established, and many industrial applications exist. Provided with historical data  $Y$  collected during normal process operation, most of the common cause variations in the process can be expressed in terms of a small number of linear principal components. Below is the methodology of how a conventional linear principal component model is developed for MSPC. Conventional LPCA is scaled dependent, therefore before a conventional LPCA can be performed on a data set, it has to be scaled in some meaningful way. The most common form of scaling a data set for conventional LPCA is standardisation.

Let the original standardised data set =  $Y (m * q)$

Where  $m$  is the number of observations and  $q$  the number of variables.

Then conventional LPCA decomposes the observation matrix  $Y$  as:

$$Y = T * P^T = \sum_{i=1}^q t_i * p_i^T \quad (5.1)$$

Where  $t$  is the scores,  $p$  is the linear principal components, which are column vectors, and  $q$  is the number of variables in the original data set. Then  $Y$  can be approximated by using any number of linear principal components. It is common practice to limit the number of

linear principal components to no more than three, for the convenience of visual inspection. For this study, the first two linear principal components will be used to approximate  $Y$  by the following equation:

$$\hat{Y} = \sum_{i=1}^2 t_i * p_i^T \quad (5.2)$$

Where  $Y$  approximated by the first two linear principal components is given by  $\hat{Y}$ .

**N.B.**  $\hat{Y}$  could be obtained by using any number of linear principal components from 1 till  $q$ . Usually the number of linear principal components that are retained are those which capture most of the relevant variations in the data.

*A conventional linear principal component model based on historical data has now been established.* New multivariate observations can be projected onto the plane defined by the linear principal component loading vectors to obtain their scores. The new scores are obtained by the following equation:

$$t_{i,new} = y_{new} * p_i \quad (5.3)$$

where  $y_{new}$  is the new multivariate observation,  $p_i$  the linear principal component loadings from the model and  $t_{i,new}$  is the new scores obtained from the model.

**N.B.** New multivariate observations have to be standardised by the mean and standard deviation of the data set, which defines the normal operating condition<sup>20</sup> (NOC) of the process. Then the reconstructed data from the new scores is given by:

$$\hat{y}_{new} = t_{a,new} * p_a^T \quad (5.4)$$

<sup>20</sup> The data set which defines the normal operating conditions of the process is also known as historical data.

Where  $\hat{y}_{new}$  is the approximated new multivariate observation using the first two linear principal components (i.e.  $a = 2$ ),  $p_a$  the linear principal component loadings from the model and  $t_{a,new}$  is obtained from Equation 5.3. The above equations (5.1-5.4) have been taken from Yoon & Macgregor (2000).

## 5.1.2 MULTIVARIATE STATISTICAL PROCESS CONTROL CHARTS

An advantage of multivariate process control based on PCA is that only two control charts are needed. They are the so-called  $T^2$ -chart and the squared prediction error (SPE) chart. The  $T^2$ -chart is based on the significant principal components which monitors the multivariate distance of a new measurement from the target value in the reduced principal component space and a SPE chart represents the error of the model based on the remaining principal components which monitors the deviation from the principal component model (Nijhuis, De Jong & Vandeginste, 1997). These charts with their control limits are equally applicable to linear principal component and non-linear principal component models. The control limits for the SPE control chart and the  $T^2$ -control chart are calculated from the historical data  $Y$ , collected during normal process operation, used to develop the principal component model.

### SPE-Control Chart

A MSPC chart based on the SPE of the residuals of new observations as defined by Yoon & Macgregor (2000) is given by:

$$SPE = \sum_{i=1}^q (y_{new,i} - \hat{y}_{new,i})^2 \quad (5.5)$$

The SPE is the squared difference between the observed and the predicted values from the nominal or reference model for each new observation and is used for early warning of

non-conforming operation. The SPE statistic represents unstructured residuals that cannot be accounted for by the principal component model. The upper control limits of the SPE control chart is computed based on historical data and is defined by Aldrich (1998-1999) as:

$$CL(SPE) = \overline{SPE} + \frac{S}{\sqrt{n}} * t_{\alpha/2} \quad (5.6)$$

$CL$  - Control Limit

$\overline{SPE}$  - Mean SPE of the data set, which defines the NOC <sup>21</sup>.

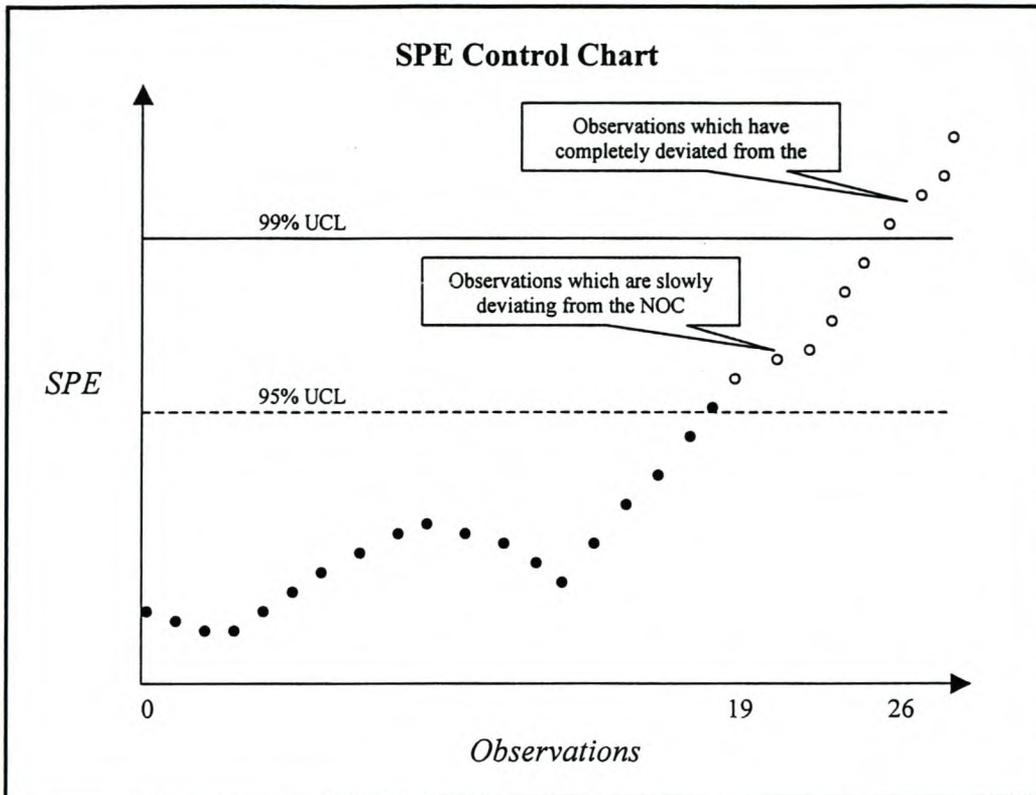
$S$  - Standard deviation of the data set which defines the NOC of the process.

$n$  - Is the number of samples in the data set which defines the NOC of the process.

$t_{\alpha/2}$  - Is the upper  $\alpha/2$  point of the T-distribution with  $n-1$  degrees of freedom.

For this study the 95 and 99 % upper control limits were used in the SPE control chart. The 95% upper control limit is computed by setting  $\alpha = 0.05$  and the 99% upper control limit is computed by setting  $\alpha = 0.01$  in Equation 5.6.

<sup>21</sup> NOC refers to the historical data set used to develop the principal component model.



*Figure 5.1: SPE control chart indicating a process deviation*

Figure 5.1 is an example of a SPE control chart. Observations 1-19 (black dots) indicate that the process is in control but slowly beginning to deviate from the NOC of the process. The slow deviation is indicated by observations 20-26 (circles) lying outside the 95% upper control limit. Observations 27-30 indicates that the process has completely deviated from the NOC of the process as these observations lie outside the 99% upper control limit.

### **T<sup>2</sup>-Control Chart**

A multivariate control chart based on Hotellings  $T^2$  statistic can be plotted based on the first  $a$  (where  $a = 2$ ) principal components, where the  $T^2$  statistic is defined by Yoon & Macgregor (2000) as:

$$T_a^2 = \sum_{i=1}^a \frac{t_i^2}{s_{t_i}^2} \quad (5.7)$$

$s_{t_i}^2$  is the estimated variance of  $t_i$  according to the principal component model based on historical data. The upper control limit of the  $T^2$  control chart is computed based on historical data and is defined by Nijhuis et al. (1997) as:

$$T_{uct}^2 = \frac{(n-1) * (n+1) * a}{n(n * a)} * F_{\alpha}(a, n-a) \quad (5.8)$$

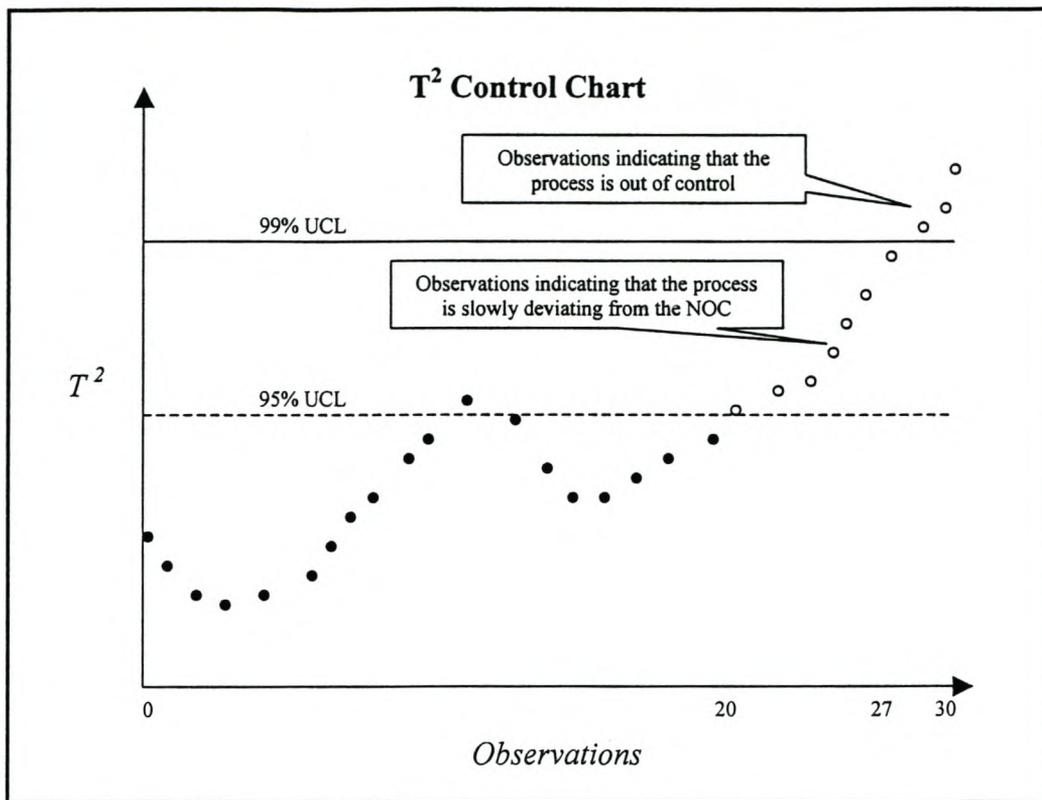
$n$  - Is the number of samples in the data set which defines the NOC of the process

$a$  - Is the number of principal components retained in the principal component model (i.e.

$a = 2$ )

$F_{\alpha}(a, n-a)$  - Is the upper  $\alpha$  point of the F-distribution with  $a$  and  $n-a$  degrees of freedom.

Equation 5.8 makes use of the assumption that the  $T^2$  statistic can be approximated by an F-Distribution. For this study the 95 and 99 % upper control limits were used in the  $T^2$  control chart. The 95% upper control limit is computed by setting  $\alpha = 0.05$  and the 99% upper control limit is computed by setting  $\alpha = 0.01$  in Equation 5.8.



*Figure 5.2:  $T^2$  control chart indicating a process deviation*

Figure 5.2 is an example of a  $T^2$  control chart. Observations 1-20 (black dots) indicate that the process is in control but slowly beginning to deviate from the NOC of the process. The slow deviation is indicated by observations 20-27 (circles) lying outside the 95% upper control limit. Observations 27-30 indicates that the process has completely deviated from the NOC of the process as these observations lie outside the 99% upper control limit.

### 5.1.3 PRINCIPAL COMPONENT MODELS BASED ON AUTOASSOCIATIVE NEURAL NETWORKS

It has thus far being illustrated that conventional linear principal component models can be used as a tool to monitor multivariate processes. The shortcoming of conventional linear principal component models in multivariate statistical process is that it is a linear technique and may not perform well when used to control a process, which is highly non-linear. For highly non-linear processes, this form of monitoring may not be efficient since the process dimensionality cannot be represented by a small number of linear principal components. Non linearly correlated process variables can be reduced to a set of non-linear principal components, through the application of NLPCA. Efficient process monitoring can then be performed in a low dimensional non-linear principal component space (Zhang, Martin & Morris, 1997).

The following section discusses the use of LPCA and NLPCA via AANNs in MSPC. Several approaches for NLPCA have been proposed, with most techniques being based upon artificial neural networks. In this study, the AANN approach prospered by Kramer (1991) has been adopted to monitor process operation. Sequential PCA as described in Section 3.4.2 was used to extract the first three principal components via an AANN. Three different AANN models have been presented in this work thus far. They are:

- AANN with one hidden layer using a linear activation function in the single node of the hidden layer.
- AANN with one hidden layer using a non-linear activation function in the single node of the hidden layer.
- AANN with three hidden layers using non-linear activation functions in the nodes of all three hidden layers.

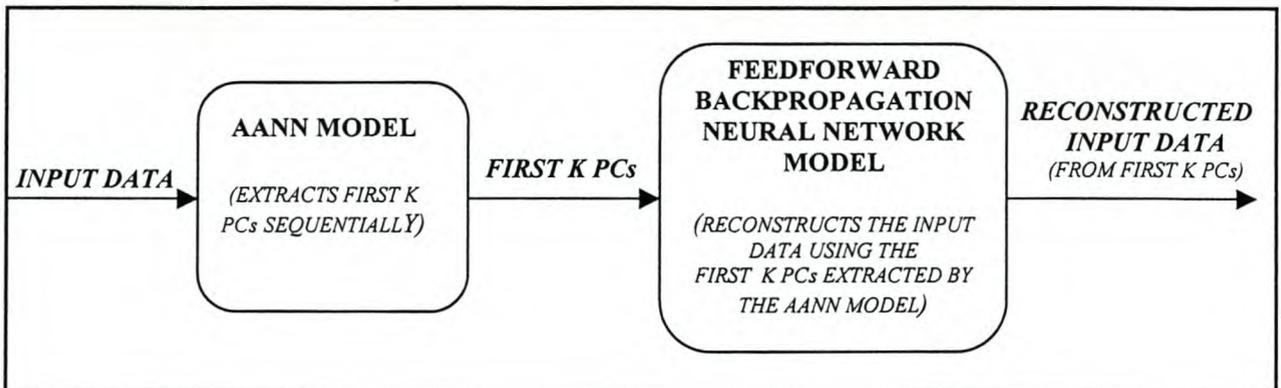
For each of the eight classes of data described in Section 4.1.1 an AANN model was designed to extract the first three principal components using each of the three AANN

models described above. Thus for each class of data an AANN model has been designed where each model extracts the first three principal components via a sequential AANN.

### 5.1.3.1 METHODOLOGY USED FOR DEFINING THE NOC OF THE PROCESS

An AANN model of a process is obtained by training the AANN on historical data collected during normal process operation. Once the AANN is trained on the historical data and is validated on a test set that gives acceptable results, a model of the process has been acquired. Two of the three principal components obtained during training of the AANN model on historical data partly define the NOC of the process.

A feedforward backpropagation neural network is then trained to reconstruct the original historical data from the first two principal components obtained from the AANN model. The sum squared difference between the reconstructed historical data using the first two principal components and the original historical data gives the SPE associated with the NOC of the process. The SPE associated with the NOC of the process along with the first two principal components obtained from the historical data define the NOC of the process. Figure 5.3 illustrates how a MSPC model can be designed using an AANN and a feedforward backpropagation neural network model.



*Figure 5.3: Multivariate statistical process control via an autoassociative neural network and feedforward backpropagation neural network*

### 5.1.3.2 METHODOLOGY USED FOR THE PROJECTION OF NEW OBSERVATIONS ONTO THE PLOT DEFINING THE NOC OF THE PROCESS

A new observation is inputted into the AANN model to give the first three principal components<sup>22</sup>. The first two principal components are then used as inputs into the feedforward backpropagation neural network model, which gives as its output the reconstructed new observation (Refer to Figure 5.3). The SPE (As defined by Equation 5.5) associated with the new observation is given by the sum-squared difference between the new observation and the reconstructed new observation obtained from the feedforward backpropagation neural network model. The new observation can be then plotted on the control charts, which defines the NOC of the process.

The  $T^2$  and SPE controls charts discussed in the development of the conventional linear principal component models can also be developed for principal component models via AANNs. The procedures for defining the SPE and the  $T^2$  statistic is exactly the same as that discussed in the conventional linear principal component models (Section 5.1.1). The variance used to calculate the  $T^2$  statistic is obtained from the correlation coefficient associated with the AANN used to extract the principal components. If two principal components are used in the model, then the correlation coefficient associated with the AANN model used to extract the first principal component as well as the correlation coefficient associated with the AANN model used to extract the second principal component, is used as the variances to calculate the  $T^2$  statistic based on two principal components.

The following two case studies use conventional linear principal component models as well as principal component models via AANNs to monitor the behaviour of two flotation processes. Case study one uses the image data described in Section 4.1 while case study two uses flotation data based on measured variables of the flotation process.

<sup>22</sup> The principal components referred to here are the scores of the AANN model.

## 5.2 CASE STUDY ONE

### 5.2.1 DEFINING THE NORMAL OPERATING CONDITIONS

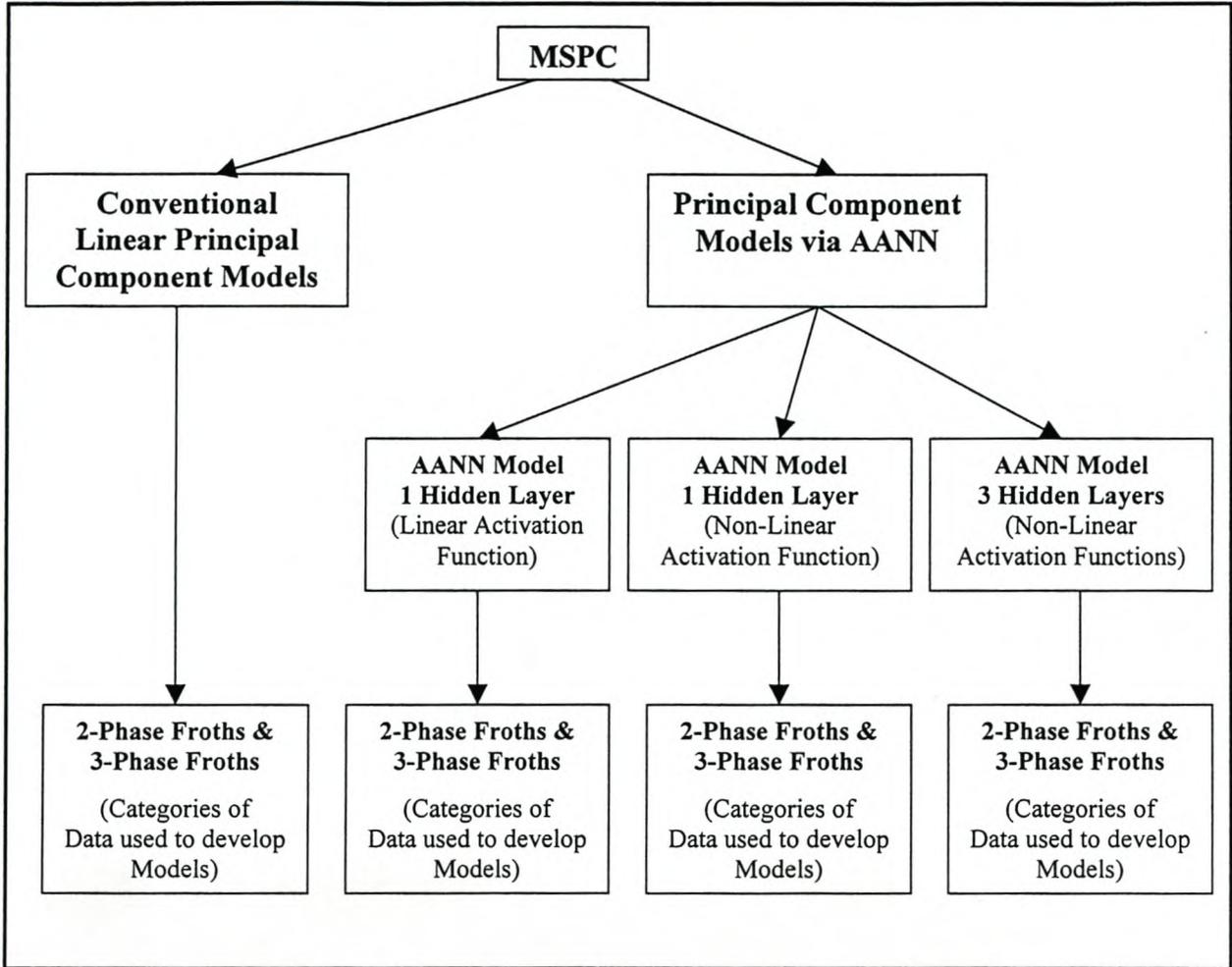
The flotation data described in Section 4.1.1 was used in developing the MSPC models for the control of a flotation process. The data was classified into eight classes with two classes (Classes C1 and C2) being a combination of the remaining six base classes. Classes C1 and C2 were not used in this part of the study but only the six base classes were used. The six base classes were split evenly with three classes been 2-phase froths and the remaining three classes been 3-phase froths as can be seen in Table 5.1.

*Table 5.1: Categories to which each class of images belongs*

Categories	2-Phase Froth	3-Phase Froth
Classes	C3	C4
	C5	C6
	C7	C8

MSPC charts were developed using conventional linear principal component models as well as principal component models via AANNs to model the two categories of data as can be seen in Figure 5.4. For the 2-phase froths, data from class C3 was chosen to represent the NOC (Historical data) which was used to develop the conventional linear principal component models and principal component models via AANNs. While classes C5 and C7 were used as situations which deviated from the NOC (C3) and was simulated by the conventional linear principal component models and principal component models via AANN. For the 3-phase froths data from class C4 was chosen to represent the NOC (Historical data) which was used to develop the conventional linear principal component models and principal component models via AANNs. Classes C6 and C8 were used as

situations which deviated from the NOC (C4) and was simulated by the conventional linear principal component models and principal component models via AANN.



*Figure 5.4: MSPC models developed using conventional linear principal component models and non-linear principal component models*

## 5.2.2 STANDARDISATION OF DATA USED FOR CONVENTIONAL LINEAR PRINCIPAL COMPONENT MODELS

Conventional LPCA is scale dependent, therefore the data has to be scaled in some meaningful way. The data from class C3 and class C4 which defined the NOC of the process for the 2-phase froths and 3-phase froths respectively were scaled via standardisation (i.e. the data was centred). Data from class C5 and C7 were standardised using the mean and standard deviation obtained from the standardisation of data from class C3 while data from class C6 and C8 were standardised using the mean and standard deviation obtained from the standardisation of data from class C4.

The following section will discuss the development of the conventional linear principal component models, based on the theory explained in Section 5.1.1, for the 2-phase froths and 3 Phase Froths.

## 5.2.3 CONVENTIONAL LINEAR PRINCIPAL COMPONENT MODELS

### 5.2.3.1 2-Phase Froths

A conventional LPCA was performed on the standardised data set from class C3 using the princomp function in Matlabs statistics toolbox. The data set from class C3 represents the normal operating Condition (NOC) of the process. The conventional LPCA on C3 yielded the scores<sup>23</sup>, latent variables<sup>24</sup> and the principal components<sup>25</sup>. The first 3 linear principal components accounted for most of the variation in the data as can be seen in Table 5.2.

<sup>23</sup> Is the data in the new co-ordinate system defined by the LPCs.

<sup>24</sup> Is the variance explained by each LPC.

<sup>25</sup> These are the linear combinations of the original variables that generate the new variables.

**Table 5.2: The percentage of the total variability explained by the first five linear principal components**

<i>Linear Principal Components</i>	<i>Percentage of Variability (%)</i>
<i>LPC 1</i>	<i>67</i>
<i>LPC 2</i>	<i>6</i>
<i>LPC 3</i>	<i>5</i>
<i>LPC 4</i>	<i>2</i>
<i>LPC 5</i>	<i>1</i>
<i>Total</i>	<i>81</i>

The first three linear principal components explain roughly 78% of the variability in the data and the first five linear principal components explain roughly 81% of the variability in the data. The first three linear principal components account for most of the variance in the data set as can be seen in Table 5.2.

**N.B.** The percent of the total variability explained by each linear principal component is given by:

$$\% \text{ Variance explained by } LPC_i = \frac{\text{Latent}(i)}{\sum_{i=1}^q \text{Latent}(i)} \quad (5.9)$$

Where  $q$  is the number of variables in the data set and latent is a vector of size  $(q*1)$  which is obtained from the conventional LPCA performed in Matlab.

C3 was reconstructed using the first score and the first linear principal component, first two scores and first two linear principal components and the first three scores and first three according to Equation 5.2. A measure of how well the model reconstructed the original data (C3) using the first score and the first linear principal component, first two

and first two linear principal components and the first three scores and first three linear principal components can be obtained by the calculation of the correlation coefficient between the original data (C3), and the reconstructed data using the conventional linear principal component model.

From Table 5.3 above it can be seen that the correlation coefficient between the original data (C3) and reconstructed data using the first score and the first linear principal component is 0.82 indicating that the first linear principal component accounts for most of the variance in the data, as indicated in Table 5.2. If the correlation coefficient of 0.82 is squared, it gives a value of 0.67, which is the value seen in Table 5.2. Thus, the correlation coefficient is an accurate measure of the amount of variance accounted for by a linear principal component.

The reconstruction of C3 using the second score and second linear principal component accounts for only 0.035 ( $0.855 - 0.82$ ) of the variance in the original data and the reconstruction using the third score and third linear principal component accounts for only 0.03 ( $0.88 - 0.855$ ) of the variance in the original data. The first two linear principal components explain approximately 85% of the variability in the original data. It is thus justified to use the first two linear principal components in the model since they accounted for more than 80% of the variation in the data.

**Table 5.3: Correlation Coefficients between original data and reconstructed data using the conventional linear principal component model based on C3**

<b>Number Of LPCs And Scores Used In Reconstruction of Data</b>	<b>Correlation Coefficient For NOC</b>	<b>Correlation Coefficient For Condition 1</b>	<b>Correlation Coefficient For Condition 2</b>
First score and first LPC	0.82	0.87	0.81
First two scores and first two LPCs	0.855	0.88	0.82
First three scores and first three LPCs	0.88	0.89	0.84

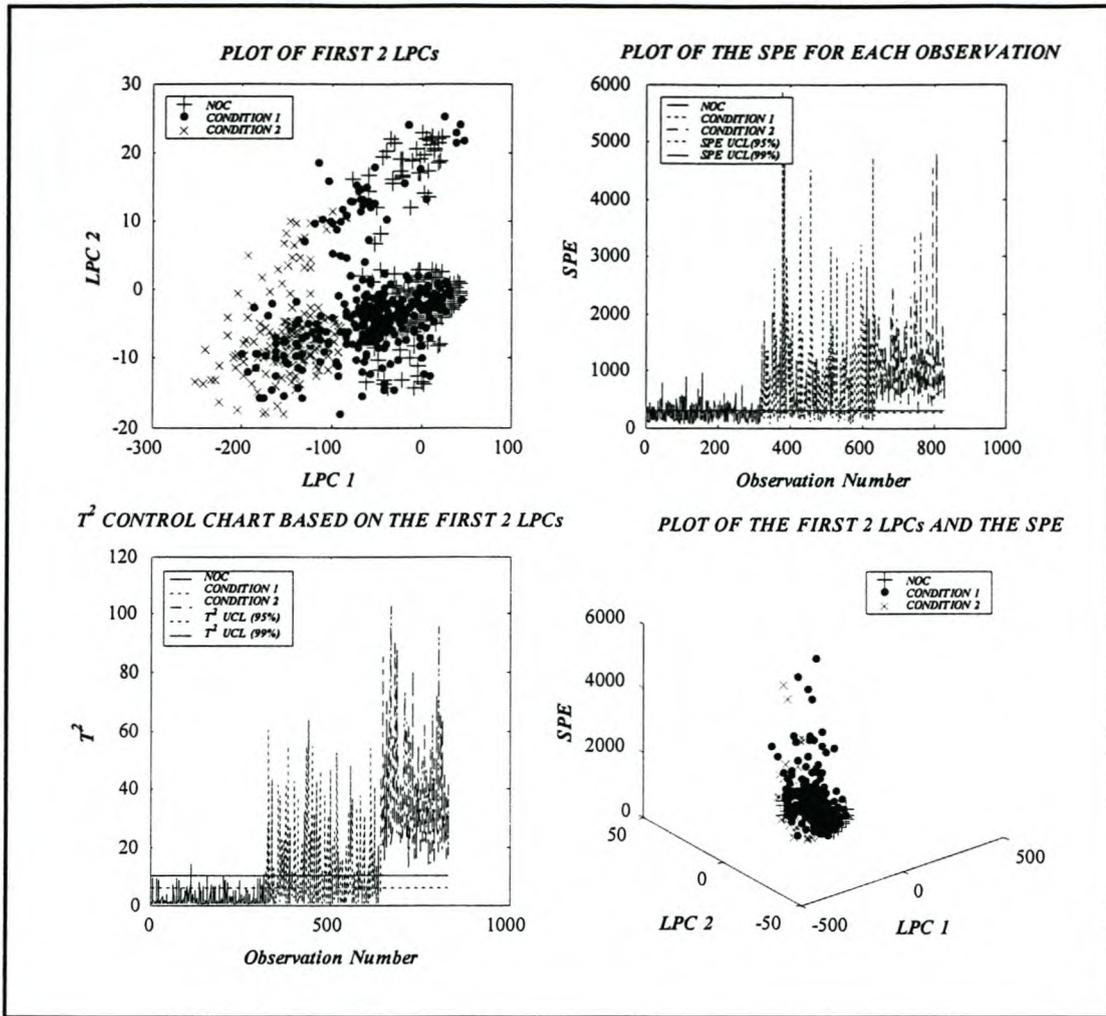
Having established a conventional linear principal component model based on historical data (C3) collected when only common cause variation was present future behaviour can be referenced against this “in control model”.

The model obtained for the NOC of the process was presented with two data sets, which have deviated, from the NOC of the process. The first data set was C5, which represents Condition 1 and the second data set, was C7, which represents Condition 2. The scores of the new data sets were obtained by using Equation 5.3. The data was reconstructed according to Equation 5.4 using the first, second and third linear principal components as was done for the NOC, which is defined by the data set (C3). The correlation coefficient was calculated between the original data set C5 and the reconstructed data set of C5 using the first three linear principal components and similarly for C7. The results can be seen in Table 5.3 above.

The SPE was calculated according to Equation 5.5, which is the squared difference between the original data and the reconstructed data, by the model using the first two linear principal components. Table 5.4 gives the mean of the SPE for each data set. From Table 5.4 it can be seen that the mean SPE increases from the NOC to Condition 1 and Condition 2. Indicating that the process is slowly deviating from the NOC of the process.

**Table 5.4: Mean SPE between original data and reconstructed data of the model**

<b>Data Sets</b>	<b>Number Of Observations</b>	<b>Mean SPE</b>
NOC (C3)	320	261
Condition 1 (C5)	320	830
Condition 2 (C7)	192	1045



**Figure 5.5: Plot of the control charts based on the conventional linear principal component model of class C3**

The plot of the first two linear principal components shows how Condition 1 and Condition 2 deviate from the NOC of the process. The upper control limits for the SPE chart are calculated according to Equation 5.6. The solid line represents the 95% upper control limit, and the dashed line represents the 99% upper control limit. In all the SPE charts to follow in this chapter, the control limits are drawn the same as in Figure 5.5.

The SPE control chart shows the SPE associated with each observation for the NOC, Condition 1 and Condition 2. The NOC (Observations 1-320) has a very low SPE when compared to that of Condition 1 (Observations 321-640) and Condition 2 (Observations

641-832). Many more of the observations of Condition 1 lie above the 95 and 99% upper control limit compared to that of the NOC of the process. This indicates that the process is deviating from the NOC. Condition 2 has completely deviated from the NOC of the process as all observations lie above the 95 and 99% upper control limit when compared to that of the NOC. This is further verified by the  $T^2$  control chart which shows how the  $T^2$  statistic of many observations of Condition 1 lie above the 95 and 99 % upper control limit and all the observations of Condition 2 lie outside the 95 and 99 % upper control limit indicating that Condition 2 has totally deviated from the NOC of the process.

A plot of the scores of the first two linear principal components along with the SPE is given in the bottom right hand corner of Figure 5.5. The SPE of Condition 1 and Condition 2 have much higher values than the NOC. This indicates that Condition 1 and Condition 2 have deviated from the NOC with regard to the scores and the SPE. It can be seen clearly in this plot how Condition 1 slowly deviates from the NOC of the process while Condition 2 has completely deviated from the NOC.

### 5.2.3.2 3-Phase Froths

A conventional LPCA was performed on the standardised data set from class C4 using Matlabs statistics toolbox. The data set from class C4 represents the NOCs of the process. The conventional LPCA on C4 yielded the scores, latent variables and the linear principal components. The first three linear principal components accounted for most of the variation in the data as can be seen in Table 5.5.

**Table 5.5: The percentage of the total variability explained by the first five linear principal components**

<i>Linear Principal Components</i>	<i>Percentage of Variability (%)</i>
<i>LPC 1</i>	88
<i>LPC 2</i>	3
<i>LPC 3</i>	0.7
<i>LPC 4</i>	0.6
<i>LPC 5</i>	0.4
<i>Total</i>	92.5

The first three linear principal components explain approximately 91% of the variability in the data and the first five linear principal components explain approximately 92.5% of the variability in the data. The percent of the total variability explained by each principal component is given by Equation 5.9. From Table 5.5 it can be seen that the first two linear principal components are the most dominant.

C4 was reconstructed using the first score and the first linear principal component, first two scores and first two linear principal components and the first three scores and first three linear principal components according to Equation 5.4. A measure of well the model reconstructed the original data (C4) using the first score and the first linear principal component, first two scores and first two linear principal components and the first three scores and first three linear principal components can be obtained by the calculation of the correlation coefficient between the original data (C4) and the reconstructed data by the model.

From Table 5.6 above it can be seen that the correlation coefficient between the original data (C4) and reconstructed data using the first score and the first LPC is 0.94 indicating that the first linear principal component accounts for most of the variance in the data as

indicated in Table 5.6. The second score and second linear principal component accounts for only approximately 0.01 (0.95 – 0.94) of the variance in the original data and the third score and third linear principal component accounts for only approximately 0.01 (0.96 – 0.95) of the variance in the original data. The first two linear principal components explain approximately 95% of the variability. Having established a conventional linear principal component model based on historical data (C4) collected when only common cause variation was present, future behaviour can be referenced against this in control model.

**Table 5.6: Correlation Coefficients between original data (C4) and reconstructed data**

<b>Number of LPCs And Scores Used In Reconstruction of Data</b>	<b>Correlation Coefficient For NOC</b>	<b>Correlation Coefficient For Condition 1</b>	<b>Correlation Coefficient For Condition 2</b>
First score and first LPC	0.937	0.943	0.890
First two scores and first two LPCs	0.952	0.960	0.912
First three scores and first three LPCs	0.956	0.963	0.920

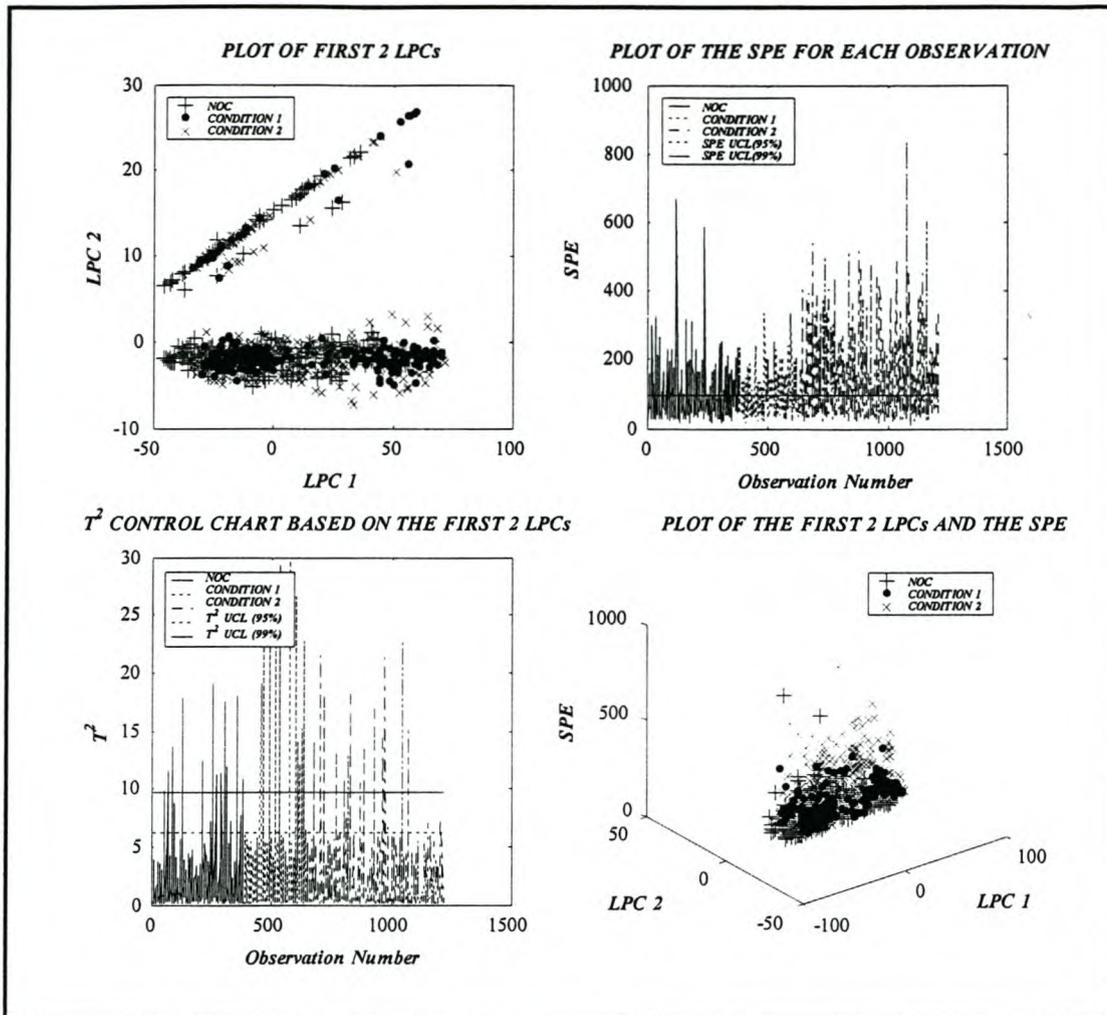
The model obtained above was presented with two data sets, which have deviated, from the NOC of the process. The first data set is C6, which represents Condition 1, and the second data set is C8, which represents Condition 2. The scores of the new data sets were obtained using Equation 5.3. The data was reconstructed according to Equation 5.4 using the first, second and third linear principal components as was done for the NOC defined by the data set (C4). The correlation coefficient was calculated between the original data set C6 and the reconstructed data set C6 using the first three linear principal components and similarly for C8. From the results of Table 5.6 it can be seen that model recreates data from Condition 1 very accurately as indicated by the high values of the correlation coefficient indicating that Condition two does not deviate from the NOC of the process. Condition 2 slowly begins to deviate from the NOC. This is evident from a decrease in

the value of the correlation coefficient of the reconstruction of the data. Table 5.7 gives the mean of the SPE for each data set.

*Table 5.7: Mean SPE between original data and reconstructed data of the model*

Data Sets	Number Of Observations	Mean SPE
NOC (C4)	384	89
Condition 1 (C6)	256	97
Condition 2 (C8)	576	127

An indication of whether Condition 1 and Condition 2 deviate from the NOC of the process was obtained from the mean SPE associated with Condition 1 and Condition 2. The mean SPE of Condition 1 indicates that the process has slightly deviated from the NOC while the mean SPE of Condition 2 indicates that it has deviated much more than Condition 1.



**Figure 5.6: Plot of the control charts based on the conventional linear principal component model of class C4**

From the plot of the first two scores it can be seen that the scores of Condition 1 and Condition 2 lie within the scores of the NOC. This chart does not detect the deviation of Condition 1 and Condition 2 from the NOC of the process. The SPE associated with each observation for the NOC (Observations 1-384), Condition 1 (Observations 385-640) and Condition 2 (Observations 641-1216) is given in the SPE control chart. From the SPE control chart, it is not clear whether Condition 1 and Condition 2 have deviated from the NOC of the process. The  $T^2$  control chart further verifies this. The control chart based on the scores of the first two linear principal components and the SPE indicates that Condition 2 is deviating from the NOC of the process. The SPE and the scores of

Condition 1 lies within the NOC of the process whereas Condition 2 begins to move away from the NOC with observations having much higher SPE values than that of the NOC. The scores of Condition 2 remain within that of the scores of the NOC indicating that the process is beginning to deviate from the NOC. This event could not be clearly detected from the previous control charts.

## **5.2.4 PRINCIPAL COMPONENT MODELS USING AUTOASSOCIATIVE NEURAL NETWORKS**

The classes of data that were defined as the NOC of the process for the 2-phase froths and the 3-phase froths that were used in developing the conventional linear principal component models for MSPC will again be used in this section for the development of principal component models using AANNs. Data from class C3 represents the NOC for the 2-phase froths and data from class C4 represents the NOC for the 3-phase froths. Data from classes C5 and C7 are used as conditions that deviate from the NOC for the 2-phase froths and classes C6 and C8 are used as conditions that deviate from the NOC for the 3-phase froths.

In Section 4.5 it has been shown that LPCA and NLPCA can be performed using AANNs. For the 2-phase and 3-phase froths linear principal component and non-linear principal component models using AANNs was used to perform MSPC.

### **5.2.4.1 2-Phase Froths**

Each of the three AANNs presented in Section 4.5 used data from class C3 as historical data, which defines the NOC, and data from class C5 and C7 as conditions, which have deviated, from the NOC. Data from class C5 will be referred to as Condition 1 and data from class C7 will be referred to as Condition 2.

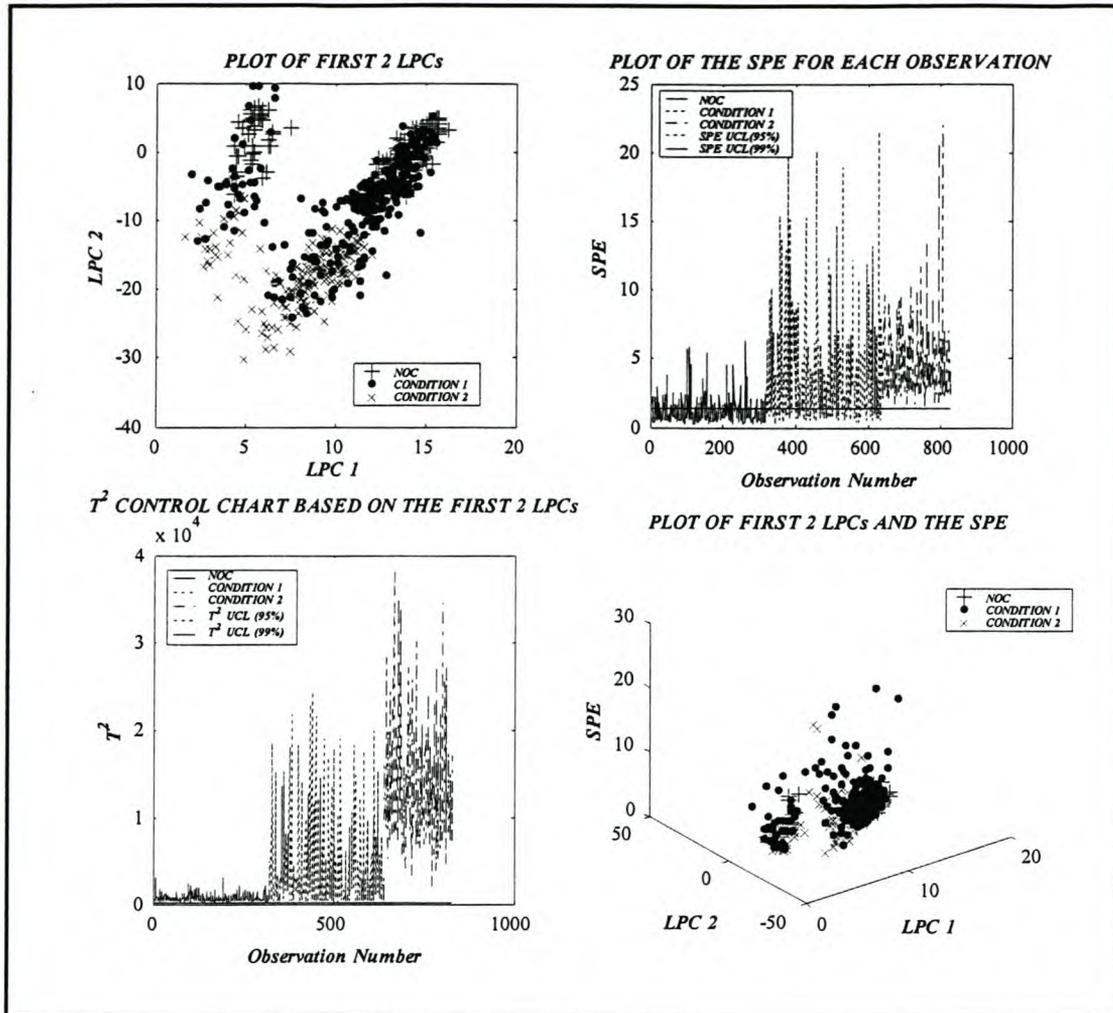
### 5.2.4.1.1 LINEAR PRINCIPAL COMPONENT MODEL-AANN WITH ONE HIDDEN LAYER (LINEAR ACTIVATION FUNCTION)

The linear principal component model developed from data of class C3 in Section 4.5.1 represents the model of the NOC of the process. The model yielded the first three linear principal components. The first two linear principal components obtained from the model was used as inputs to train a feedforward backpropagation NN model to reconstruct the original NOC data. The correlation coefficient was used as a measure of how well the model reconstructed the original data. Table 5.8 gives the correlation coefficients for the reconstruction of the data using the first linear principal component and the first two linear principal components for the NOC, Condition 1 and Condition 2.

*Table 5.8: Correlation coefficients between original data and reconstructed data*

Number Of LPCs Used In Reconstruction Of Data	Correlation Coefficient For NOC	Correlation Coefficient For Condition 1	Correlation Coefficient For Condition 2
First LPC	0.80	0.52	0.35
First two LPCs	0.92	0.89	0.74

From Table 5.8, it can be seen that the first two linear principal components of the NOC captures over 92% of the variance in the data. The first two linear principal components of Condition 1 and Condition 2 on the other hand only captures 89% and 74% respectively indicating that these conditions may be deviating from the NOC.



**Figure 5.7: Plot of the four control charts for the normal operating condition, Condition 1 and Condition 2**

The plot of the first two linear principal components in Figure 5.7 indicates how Condition 1 begins to drift out of the NOC defined by the first two linear principal components. Condition 2 is the onset of a new event in the process as no observations of Condition 2 lie in the NOC of the process defined by the first two linear principal components. Table 5.9 gives the mean SPE for each data set, which is the average SPE for each observation in the data set. As the process begins to deviate from the NOC the SPE increases as can be seen in the SPE control chart. Many of the observations of Condition 1 lie outside the 95 and 99% upper control limits of the SPE chart, indicating that the process is drifting from the NOC. Condition 2 on the other hand has totally

deviated from the NOC, with no observations lying within the 95 and 99% upper control limits of the SPE chart.

**Table 5.9: Mean SPE between original data and reconstructed data of the model**

Data Sets	Number Of Observations	Mean SPE
NOC (C3)	320	1.22
Condition 1 (C5)	320	3.70
Condition 2 (C7)	192	4.47

The  $T^2$  control chart, which further verifies that Condition 1 and Condition 2 have deviated from the NOC of the process. From the plot of the scores of the first two linear principal components and the SPE it is clear that Condition 1 and Condition 2 are deviating from the NOC.

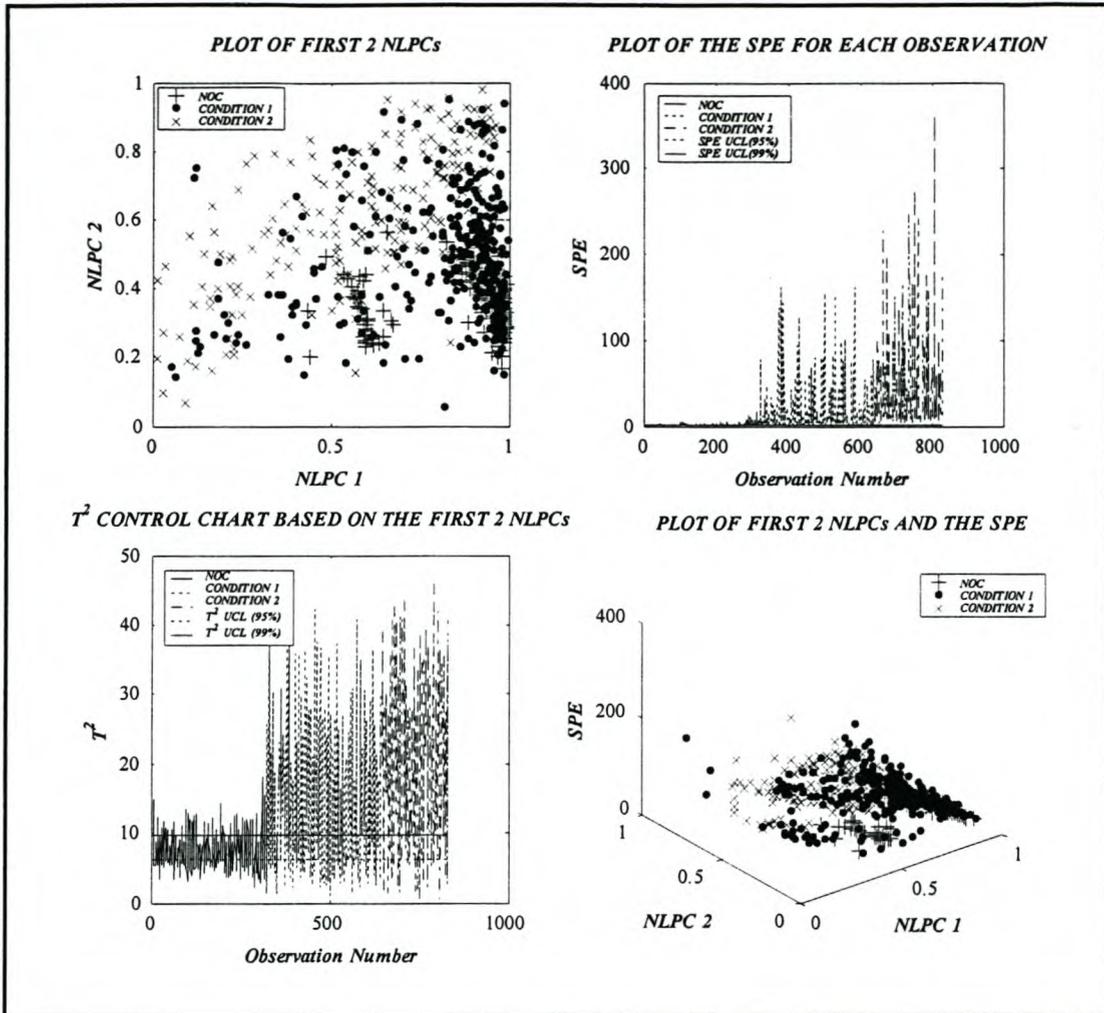
#### 5.2.4.1.2 NON-LINEAR PRINCIPAL COMPONENT MODEL-AANN WITH ONE HIDDEN LAYER (NON-LINEAR ACTIVATION FUNCTION )

The non-linear principal component model developed from data of class C3 in Section 4.5.2 represents the model of the NOC of the process. The model yielded the first three non-linear principal components. The first two non-linear principal components obtained from the model was used as inputs to train a feedforward backpropagation NN model to reconstruct the original NOC data. The correlation coefficient was used as a measure of how well the model reconstructed the original data. Table 5.10 gives the correlation coefficients for the reconstruction of the data using the first non-linear principal component and the first two non-linear principal components for the NOC, Condition 1 and Condition 2.

**Table 5.10: Correlation coefficients between original data and reconstructed data**

<b>Number Of NLPCs Used In Reconstruction Of Data</b>	<b>Correlation Coefficient For NOC</b>	<b>Correlation Coefficient For Condition 1</b>	<b>Correlation Coefficient For Condition 2</b>
First NLPC	0.80	0.48	0.34
First two NLPCs	0.91	0.60	0.37

From Table 5.10 it can be seen that the first two non-linear principal components of the NOC captures over 90% of the variance in the data. The first two non-linear principal components of Condition 1 and Condition 2 on the other hand only captures 60% and 37% of the variance in the data respectively, indicating how these conditions deviate from the NOC.



**Figure 5.8:** Plot of the four control charts for the normal operating condition, Condition 1 and Condition 2

The plot of the first two linear principal components indicates how Condition 1 begins to drift out of the NOC and Condition 2 is the onset of a new event in the process as no observations of Condition 2 lie in the NOC of the process. Table 5.11 gives the mean SPE for each data set, which is the average SPE for each observation in the data set. From the table it can be seen that the mean SPE associated with Condition 1 and Condition 2 are much higher than that of the normal operating condition.

**Table 5.11: Mean SPE between original data and reconstructed data of the model**

<b>Data Sets</b>	<b>Number Of Observations</b>	<b>Mean SPE</b>
NOC (C3)	320	1.56
Condition 1 (C5)	320	17.00
Condition 2 (C7)	192	44.04

The SPE control chart shows that several of the observations of Condition 1 lie outside the 95 and 99% upper control limits of the SPE chart. While all the observations of Condition 2 lie outside the 95 and 99% upper control limits This clearly indicates that the SPE chart can detect a change in process behaviour from that of the NOC of the process. As the process begins to deviate from the NOC the  $T^2$  statistic increases as can be seen in  $T^2$  control chart. The plot of the scores of the first two non-linear principal components and the SPE verifies that the process has deviated from the NOC as was detected in the other control charts.

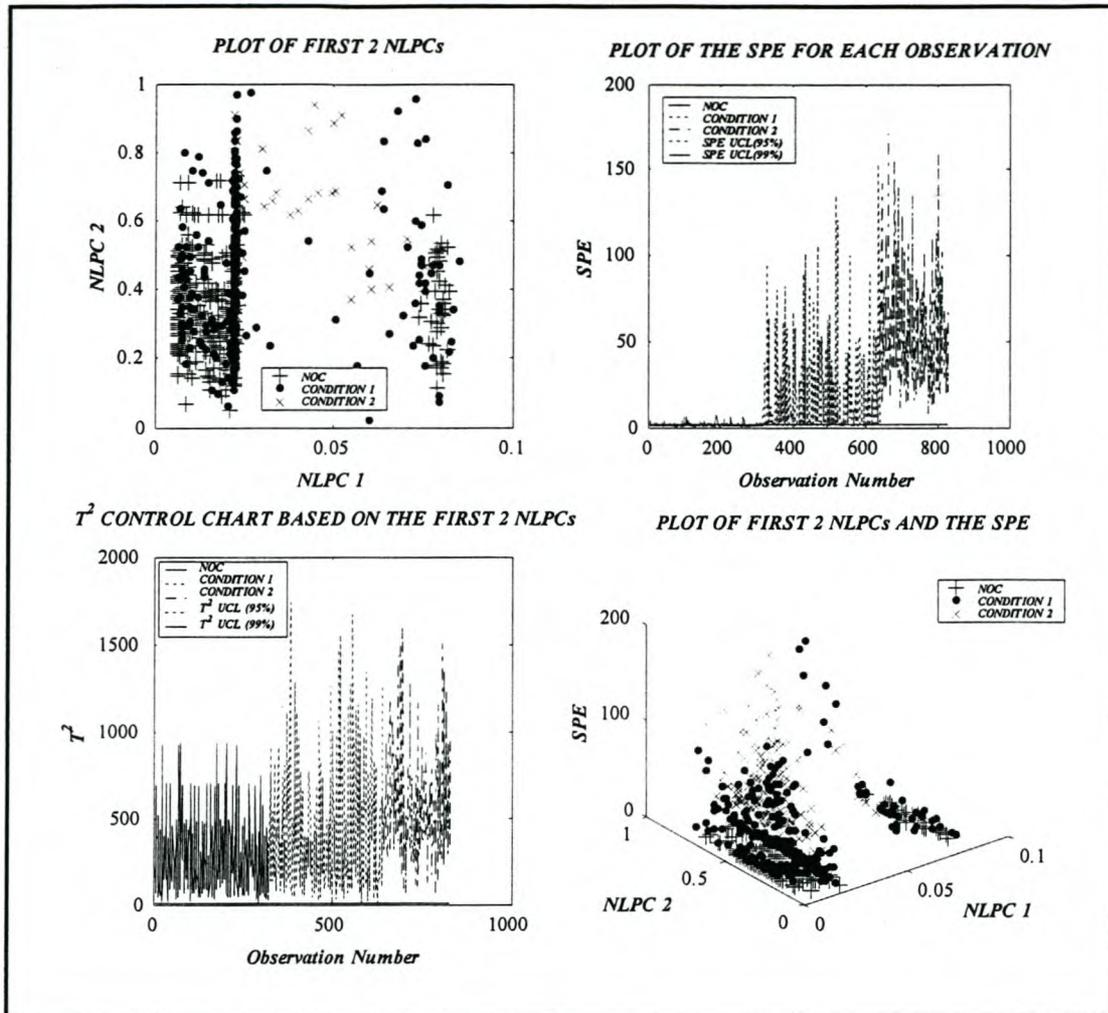
#### **5.2.4.1.3 NLPC MODEL-AANN WITH THREE HIDDEN LAYERS (NON-LINEAR ACTIVATION FUNCTIONS)**

The non-linear principal component model developed from data of class C3 in Section 4.5.3 represents the model of the NOC of the process. The model yielded the first three non-linear principal components. The first two non-linear principal components obtained from the model was used as inputs to train a feedforward backpropagation NN model to reconstruct the original NOC data. The correlation coefficient was used as a measure of how well the model reconstructed the original data. Table 5.12 gives the correlation coefficients for the reconstruction of the data using the first non-linear principal component and the first two non-linear principal components for the NOC, Condition 1 and Condition 2.

**Table 5.12: Correlation coefficients between original data and reconstructed data**

<b>Number Of NLPCs Used In Reconstruction Of Data</b>	<b>Correlation Coefficient For NOC</b>	<b>Correlation Coefficient For Condition 1</b>	<b>Correlation Coefficient For Condition 2</b>
First NLPC	0.92	0.64	0.15
First two NLPCs	0.923	0.64	0.15

From Table 5.12 it can be seen that the first two non-linear principal components of the NOC captures over 90% of the variance in the data. The first two non-linear principal components of Condition 1 and Condition 2 on the other hand only captures 64% and 15% respectively indicating that these conditions deviate from the NOC. From the plot of the first two non-linear principal components in Figure 5.9 it is not clear whether Condition 1 and Condition 2 are the onset of new events in the process as not many observations of Condition 1 and Condition 2 lie outside the NOC of the process.



**Figure 5.9:** Plot of the four control charts for the normal operating condition, Condition 1 and Condition 2

The SPE control chart in Figure 5.9 clearly indicates how Condition 1 and Condition 2 deviate from the NOC of the process. Most observations of Condition 1 lie outside the 99% upper control limit. Condition 2 has completely deviated from the NOC of the process with all observations lying outside the 99% upper control limit. The deviation of Condition 1 and Condition 2 is further verified by the high mean SPE associated with them as can be seen in Table 5.13 This deviation was not detected in the scores plot of Figure 5.9.

**Table 5.13: Mean SPE between original data and reconstructed data of the model**

Data Sets	Number Of Observations	Mean SPE
NOC (C3)	320	1.35
Condition 1 (C5)	320	15.00
Condition 2 (C7)	192	54.29

From the  $T^2$  control chart it is difficult to detect that the process is deviating from the NOC of the process as all three operating conditions lie outside the 99% upper control limit. The plot of the first two non-linear principal components and the SPE. Clearly indicate how Condition 1 slowly deviates from the NOC and how Condition 2 has completely deviated from the NOC. This deviation in process behaviour was also detected by the SPE control chart but not by the scores and  $T^2$  control chart. It can be seen that if a process deviation is not detected in one control chart it will most likely be detected in another control chart.

#### 5.2.4.2 3 Phase Froths

Data from the 3-phase froths was used in the development of principal component models via AANNs. Each AANN used data from class C4 as historical data, which defines the NOC, and data from class C6 and C8 as conditions which deviate from the NOC. Data from class C6 will be referred to as Condition 1 and data from class C8 will be referred to as Condition 2.

##### 5.2.4.2.1 LPC MODEL - AANN MODEL WITH ONE HIDDEN LAYER (LINEAR ACTIVATION FUNCTION)

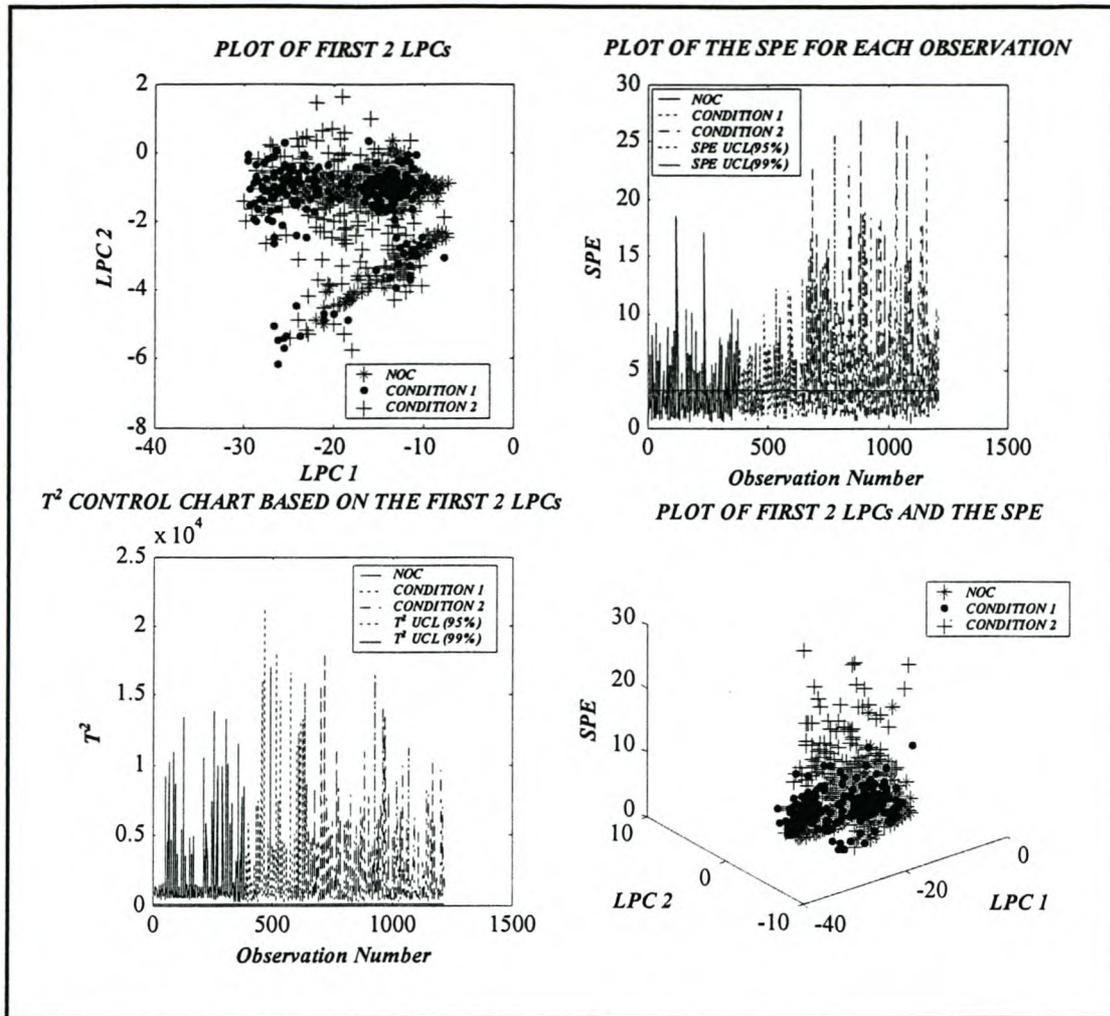
An AANN was trained with data from class C4 that represented the NOC of the process. An AANN model was obtained which yielded the first three linear principal components. The first two linear principal components obtained from the AANN model were used as

inputs to train a feedforward backpropagation neural network model to reconstruct the original NOC data. The correlation coefficient as defined by Equation 4.1 was used as a measure of how well the model reconstructed the original data. Table 5.14 gives the correlation coefficient for the reconstruction of the data using the first linear principal component and the first two linear principal components for the NOC, Condition 1 and Condition 2.

*Table 5.14: Correlation coefficients between original data and reconstructed data*

Number Of LPCs Used In Reconstruction Of Data	Correlation Coefficient For NOC	Correlation Coefficient For Condition 1	Correlation Coefficient For Condition 2
First LPC	0.9220	0.9282	0.8648
First two LPCs	0.9463	0.9501	0.8871

From the table above it can be seen that it can be seen that the first two linear principal components of the NOC captures over 94% of the variance in the data. The first two linear principal components of Condition 1 and Condition 2 on the other hand captures 95% and 88% of the variance respectively indicating that these conditions are in control. The plot of the first two linear principal components in Figure 5.10 indicates that Condition 1 and Condition 2 are in control as there are few observations of Condition 1 and Condition 2, which lie outside the NOC of the process.



**Figure 5.10: Plot of the four control charts for the normal operating condition, Condition 1 and Condition 2**

Table 5.15 gives the mean SPE for each operating condition, which is the average SPE of each observation. From Table 5.15 and the SPE control chart it can be seen that the SPE for Condition 2 is a little higher than that of the NOC indicating that a new event is occurring in the process which is not covered by the model. This was not detectable from the plot of the first two linear principal components in Figure 5.10.

**Table 5.15: Mean SPE between original data and reconstructed data of the model**

Data Sets	Number Of Observations	Mean SPE
NOC (C4)	384	2.92
Condition 1 (C6)	256	3.50
Condition 2 (C8)	576	4.74

From the  $T^2$  control chart it is difficult to determine whether Condition 1 and Condition 2 are deviating from the NOC of the process as can be seen in the  $T^2$  control chart. The plot of the scores of the first two linear principal components along with the SPE shows how Condition 2 slowly begins to deviate from the NOC with a few observations lying far above the NOC. Condition 1 seems to be in a state of control. This can be confirmed by the mean SPE been similar to that of the NOC and the correlation coefficient of the reconstruction of the data of Condition 1 been the same as the NOC.

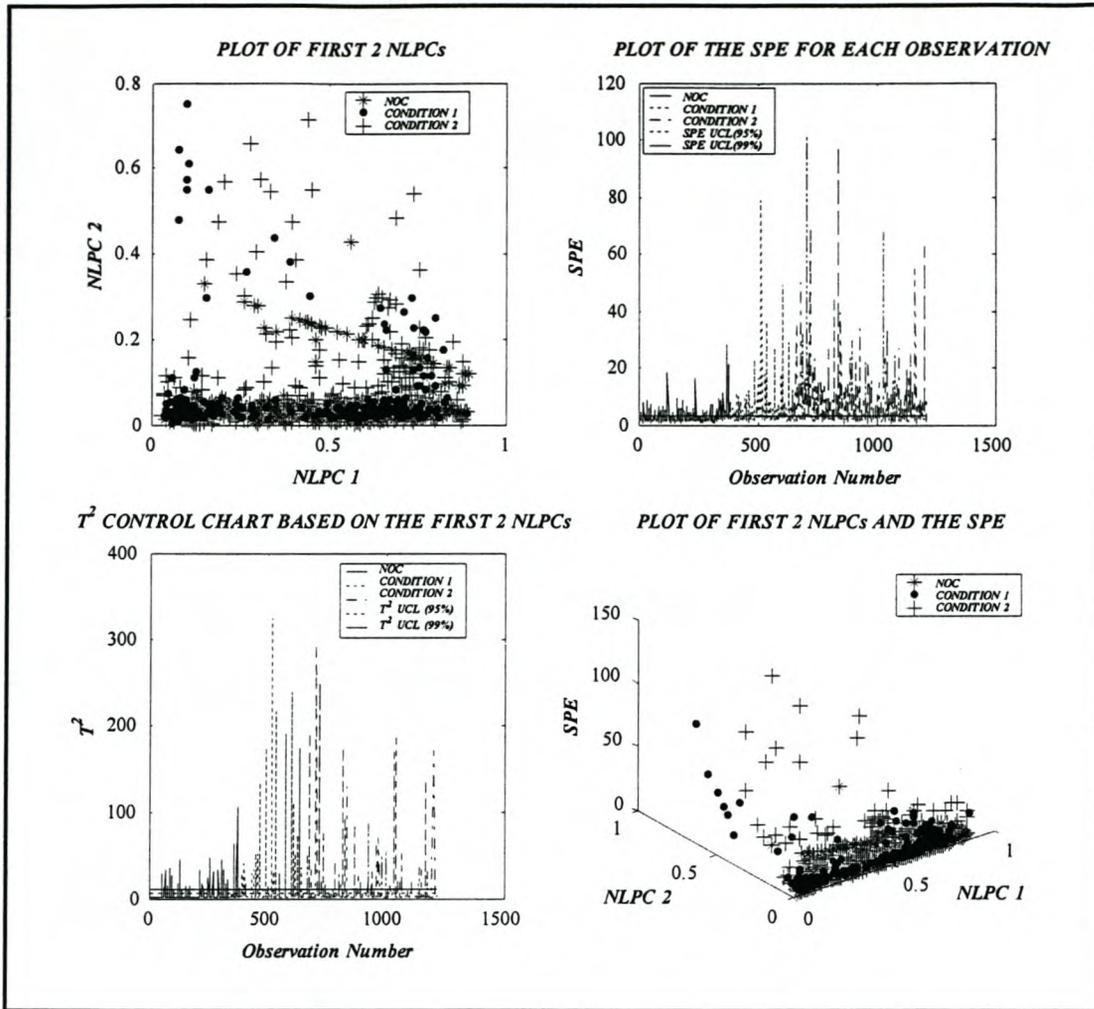
#### 5.2.4.2.2 NLPC MODEL - AANN MODEL WITH ONE HIDDEN LAYER (NON-LINEAR ACTIVATION FUNCTION)

The AANN was trained with data from class C4 that represented the NOC of the process. An AANN model was obtained which yielded the first three non-linear principal components. The first two non-linear principal components obtained from the AANN model was used as inputs to train a feedforward backpropagation neural network model to reconstruct the original NOC data. The correlation coefficient as defined by Equation 4.1 was used as a measure of how well the model reconstructed the original data. Table 5.16 gives the correlation coefficient for the reconstruction of the data using the first non-linear principal component and the first two non-linear principal components for the NOC, Condition 1 and Condition 2.

**Table 5.16: Correlation coefficients between original data and reconstructed data**

<b>Number Of NLPCs Used In Reconstruction Of Data</b>	<b>Correlation Coefficient For NOC</b>	<b>Correlation Coefficient For Condition 1</b>	<b>Correlation Coefficient For Condition 2</b>
First NLPC	0.9243	0.9183	0.84
First two NLPCs	0.9478	0.9303	0.8547

From Table 5.16 above it can be seen that the first two non-linear principal components of the NOC captures over 94% of the variance in the data. The first two non-linear principal components of Condition 1 and Condition 2 on the other hand captures 93% and 85% of the variance respectively, indicating that Condition 1 is in control and Condition 2 is slowly deviating from the NOC. From Figure 5.11 there is an indication that Condition 1 and Condition 2 may be deviating from the NOC of the process. As there are a few observations of Condition 1 and Condition 2, which lie outside the NOC of the process. From this plot, it is inconclusive if Condition 1 and Condition 2 are deviating from the NOC.



**Figure 5.11: Plot of the four control charts for the normal operating condition, Condition 1 and Condition 2**

Table 5.17 gives the mean SPE for each of the three conditions. The mean SPE of Conditions 1 and 2 is higher than that of the NOC as can be seen in Table 5.17, indicating the onset of new events that is not covered by the model.

**Table 5.17: Mean SPE between original data and reconstructed data of the model**

Data Sets	Number Of Observations	Mean SPE
NOC (C4)	384	2.85
Condition 1 (C6)	256	5.13
Condition 2 (C8)	576	6.59

The SPE control chart clearly indicates that Condition 1 and 2 are deviating from the NOC of the process as many of the observations of Condition 1 and 2 lie outside the 95 and 99% upper control limit. This is also detected by the  $T^2$  control chart as can be seen in Figure 5.11. From the plot of the scores of the first two non-linear principal components and the SPE it is difficult to detect if Condition 1 and 2 are deviating from the NOC. There is an indication that Condition 1 and 2 may be deviating from the NOC with a few observations lying outside the NOC. The deviation of Condition 1 and 2 from the NOC was detected by the SPE and  $T^2$  control charts but was not detected by the non-linear principal component control charts.

#### 5.2.4.2.3 NLPC MODEL - AANN MODEL WITH THREE HIDDEN LAYERS (NON-LINEAR ACTIVATION FUNCTIONS)

An AANN was trained with data from class C4 that represented the NOC of the process. An AANN model was obtained which yielded the first three non-linear principal components. The first two non-linear principal components obtained from the AANN model was used as inputs to train a feedforward backpropagation NN model to reconstruct the original NOC data. The correlation coefficient as defined by Equation 4.1 was used as a measure of how well the model reconstructed the original data. Table 5.18 gives the correlation coefficient for the reconstruction of the data using the first non-linear principal component and the first two non-linear principal components for the NOC, Condition 1 and Condition 2.

**Table 5.18: Correlation coefficients between original data and reconstructed data**

<b>Number Of NLPCs Used In Reconstruction Of Data</b>	<b>Correlation Coefficient For NOC</b>	<b>Correlation Coefficient For Condition 1</b>	<b>Correlation Coefficient For Condition 2</b>
First NLPC	0.926	0.923	0.85
First two NLPCs	0.937	0.899	0.85

From Table 5.18 it can be seen that the first two non-linear principal components of the NOC captures over 93% of the variance in the data. For Condition 1 the first non-linear principal component captures 92% of the variance of the data, which is, more than the variance captured by the first two non-linear principal components which is 90%. A possible reason why the data of Condition 1 could not be reconstructed better with the first two non-linear principal components could be due to feedforward backpropagation neural network model been poorly trained. The plot of the first two non-linear principal components in Figure 5.12 indicates that Condition 1 and Condition 2 are in control with most of the observations lying in the area defined by the NOC.

**Table 5.19: Mean SPE between original data and reconstructed data of the model**

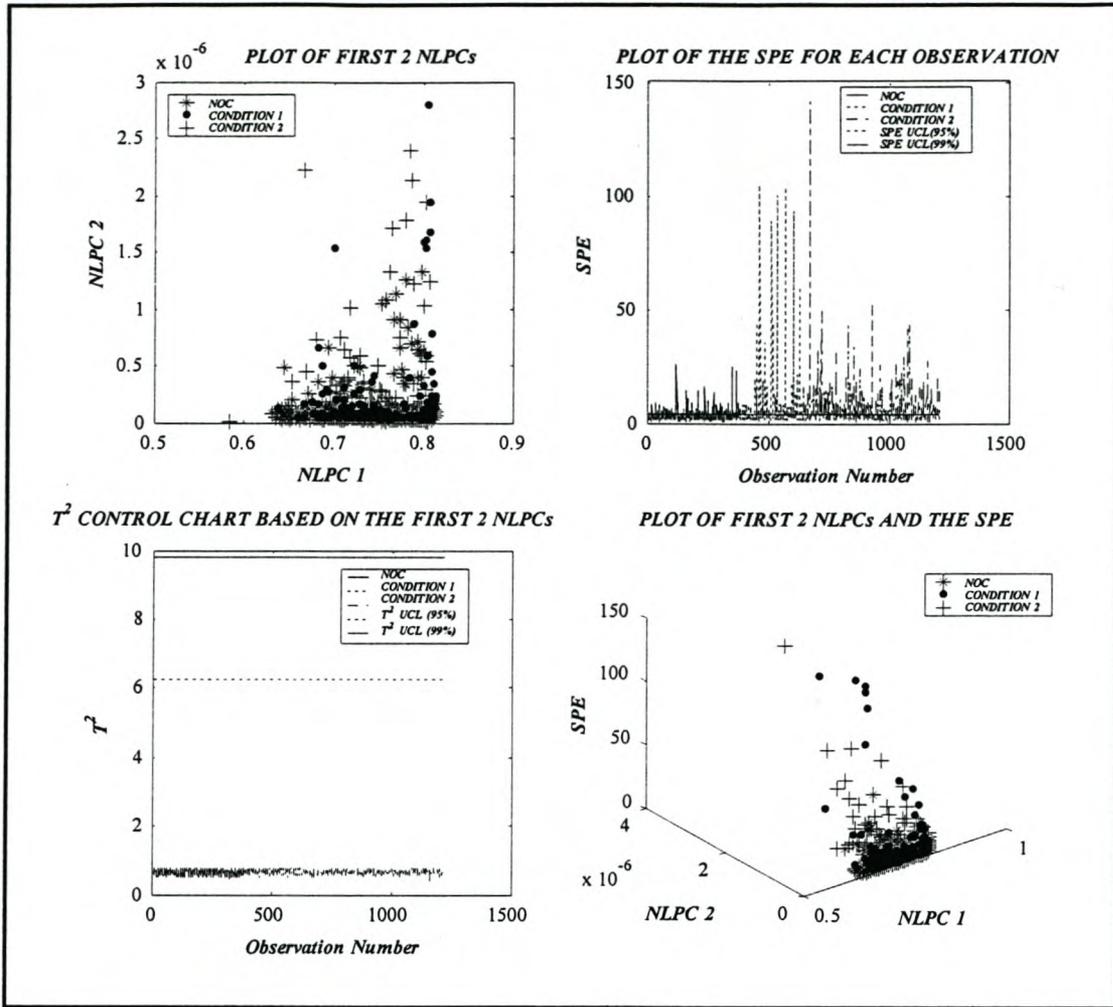
Data Sets	Number Of Observations	Mean SPE
NOC (C4)	320	3.43
Condition 1 (C6)	320	7.3
Condition 2 (C8)	192	6.3

The SPE control chart clearly indicates that several observations of Condition 1 and Condition 2 lie outside the 95 and 99% upper control limits. This is in contrast to the  $T^2$  control chart, which indicates that Condition, 1 and Condition 2 are in control as can be seen in the  $T^2$  control chart. The plot of the scores of the first two non-linear principal components and the SPE indicates that Condition 1 and Condition 2 are deviating from the NOC of the process. The SPE control chart was the only control chart, which was able to clearly detect that Condition 1 and Condition 2 were deviating from the NOC.

### 5.3 CASE STUDY TWO

A second set of data obtained from a flotation process is used to illustrate the technique of MSPC. The data used in Case 1 is of a different form to the data to be used in Case 2. In Case 1 the data used to perform MSPC was obtained from images of the flotation froth described in Section 4.1. In Case 2 the data set to be used to illustrate the technique of MSPC was obtained from five measured variables (such as airflow rate, temperature, concentration of frother, grade etc.) during the flotation process.

The data used in Case 2 was collected at three different stages of the flotation process. The first set of data collected consisted of ninety-nine observations and was collected when the flotation process was in-control or operating under process conditions. This data set will be referred to as the NOC (99\*5). The second and third sets of data collected



**Figure 5.12:** Plot of the four control charts for the normal operating condition, Condition 1 and Condition 2

Table 5.19 gives the mean SPE for each data set. From the mean SPE and the SPE control chart in Figure 5.12 it can be seen that Condition 1 and 2 deviate from the NOC. This was not detected from the non-linear principal component plot.

also consisted of ninety-nine observations each but was collected when the process had deviated from the in-control or normal operation of the process. The second and third data sets will be referred to as Condition 1 (99\*5) and Condition 2 (99\*5).

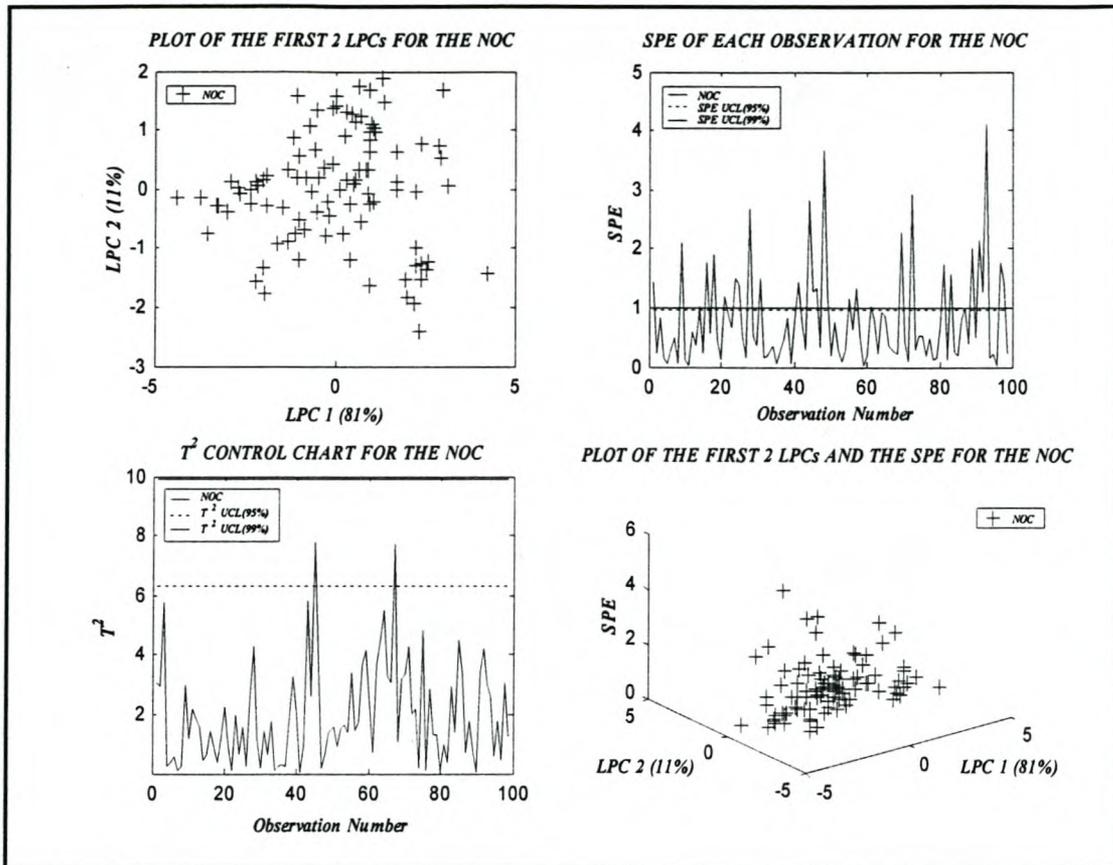
### 5.3.1 CONVENTIONAL LINEAR PRINCIPAL COMPONENT MODELS

A conventional LPCA was performed on the standardised data set of the NOC using the principal component function in Matlabs statistics toolbox. The LPCA on the NOC data yielded the scores, latent variables and the linear principal components. The first two linear principal components accounted for 84 % of the variation in the data as can be seen in Table 5.20.

*Table 5.20: The percentage of variance in the data accounted for by each linear principal component*

<i>Linear Principal Components</i>	<i>Percentage of Variance (%)</i>
<i>LPC 1</i>	<i>65</i>
<i>LPC 2</i>	<i>19</i>
<i>LPC 3</i>	<i>13</i>
<i>LPC 4</i>	<i>3</i>
<i>LPC 5</i>	<i>0</i>
<i>Total</i>	<i>100</i>

The plot of the scores of the first two linear principal components of the normal operating condition can be seen in Figure 5.13, which defines the normal operating region of the process in the reduced linear principal component space. If any new observations, which are projected onto the reduced linear principal component space lie outside the area, defined by the NOC, then the process is disturbed.



**Figure 5.13: Plot of the four control charts based on a conventional linear principal component model for the normal operating condition**

The SPE control chart of the normal operating condition with the 95 and 99% upper control limits is given in Figure 5.13. The  $T^2$  control chart of each observation for the NOC of the process together with the plot of the first two linear principal components and the SPE of the NOC are also given in Figure 5.13.

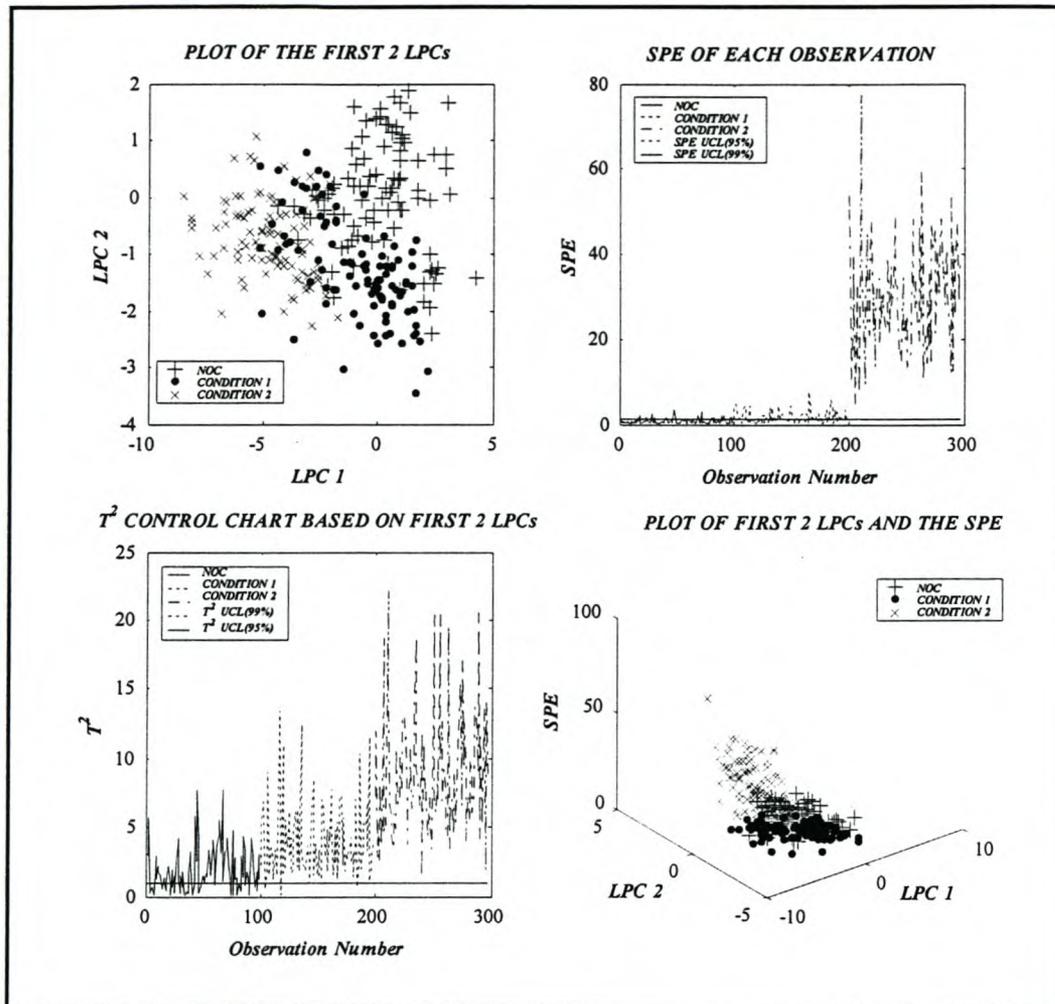
### 5.3.1.1 Onset Of New Conditions

Condition 1 and 2 was simulated by the conventional linear principal component model to monitor if Conditions 1 and 2 remains in a state of control or out of control. Conditions 1 and 2 were both standardised using the mean and standard deviation of the historical data (NOC), which was used to develop the conventional linear principal component model. The conventional linear principal component model using the first linear principal component and the first two linear principal components were used to simulate both data sets.

*Table 5.21: Correlation coefficients between original data and reconstructed data*

Number Of LPCs Used in Reconstruction of Data	Correlation Coefficient For NOC	Correlation Coefficient For Condition 1	Correlation Coefficient For Condition 2
First LPC	0.81	0.76	0.68
First two LPCs	0.92	0.90	0.69

Table 5.21 gives the reconstruction of the NOC, Condition 1 and Condition 2 with the first and first two linear principal components using the conventional linear principal component model. From this table it can be seen that Condition 1 slowly deviates from the NOC while Condition 2 has totally deviated. Figure 5.14 gives a plot of the scores of the first two linear principal components for the NOC, Condition 1 and Condition 2. It can be seen from this plot that the scores of Condition 2 lie completely outside the area defined by the NOC of the process. This plot would signal that Condition 2 has completely drifted out of control. Condition 1 is slowly drifting out of control of the NOC of the process.



**Figure 5.14:** Plot of the four control charts for the normal operating condition, Condition 1 and Condition 2

Table 5.22 gives the total SPE associated with each variable over the ninety-nine observations used to define the NOC, Condition 1 and Condition 2 of the process. Therefore for variable one the total SPE of the NOC is 35. The SPE of 35 is obtained by summing the SPE of each of the ninety-nine observations for variable one and similarly the SPE was obtained for other variables.

**Table 5.22: SPE associated with each variable over the 99 observations of the NOC, Condition 1 and 2**

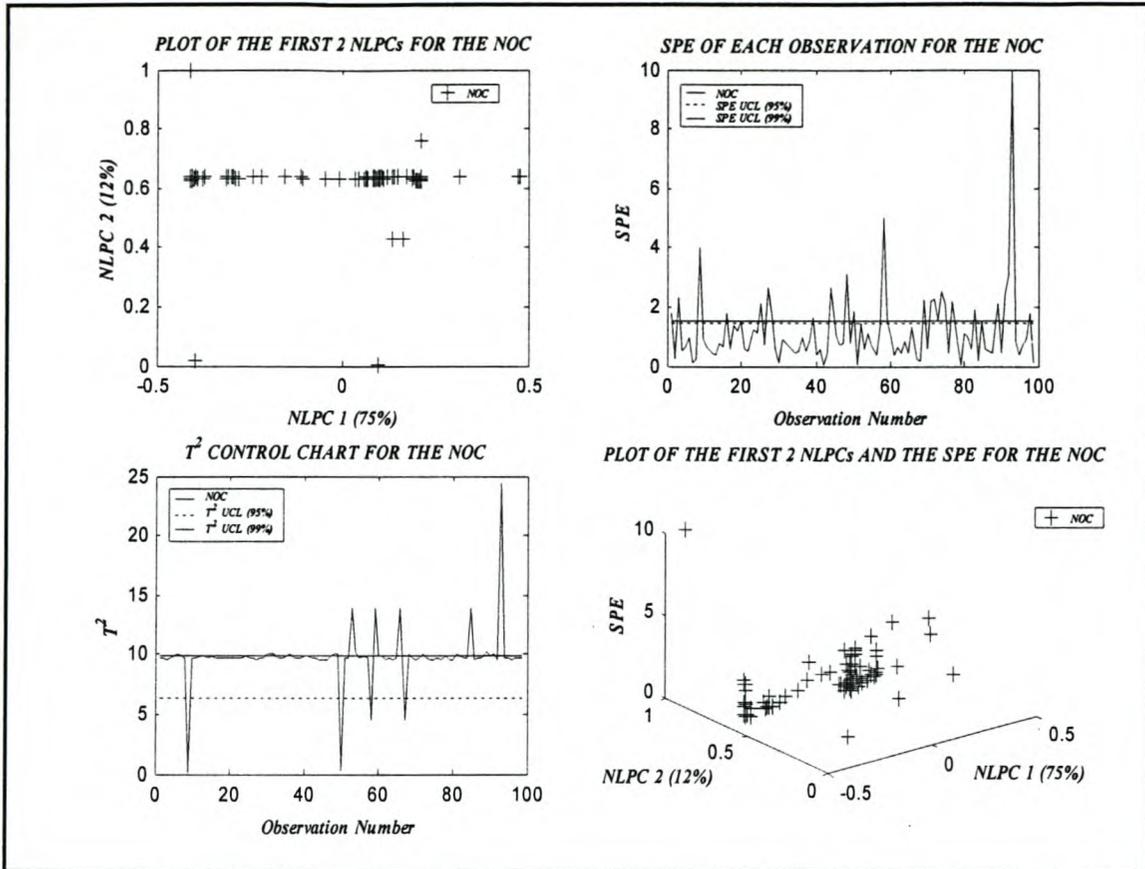
Variable	SPE of the NOC	SPE of Condition 1	SPE of Condition 2
Variable 1	34	48	1452
Variable 2	9	23	21
Variable 3	13	32	558
Variable 4	14	27	587
Variable 5	8	34	325

From Table 5.22 it can be seen that all five variables for Condition 1 have a much higher SPE over the 99 observations than that of the NOC of the process indicating that the process is deviating. Condition 2 has completely deviated according to the SPE values associated with each variable. This is confirmed from the SPE control chart in Figure 5.14 which shows that many of the observations from Condition 1 lie outside the 95 and 99% upper control while all the observations of Condition 2 lie outside the 99% upper control limit of the SPE control chart. From this plot, it is clearly evident that Conditions 1 and 2 have deviated from the NOC. The  $T^2$  control chart also indicates that Condition 1 is drifting out of control while it shows that Condition 2 has totally deviated from the NOC of the process with many observations lying outside the 99% upper control limit. The plot of the first two linear principal components and the SPE confirms that Condition 1 is deviating from the NOC while Condition 2 has completely deviated from the NOC. This is in accordance with the other three control charts.

### 5.3.2 NON-LINEAR PRINCIPAL COMPONENT MODEL- AUTOASSOCIATIVE NEURAL NETWORK WITH 3 HIDDEN LAYERS

The data set that defined the NOC of the process used in the development of the conventional linear principal component model was again used to develop a non-linear principal component model. The data that defined the NOC of the process was used to train a five-layer AANN model (Refer to Section 3.3.2.2 for details of the AANN). The AANN was trained until an acceptable sum squared error was obtained. The trained AANN model was then used to extract the first two non-linear principal components, which defined the NOC of the process.

A feedforward backpropagation neural network was designed and trained to reconstruct the NOC data from the two non-linear principal components extracted via the AANN (Refer to Figure 5.3). The SPE as defined by Equation 5.5 is calculated between the original input data to the AANN and the reconstructed data by the feedforward backpropagation neural network. The NOC data used to train and validate the AANN and feedforward backpropagation neural network models were unstandardised. The first two non-linear principal components extracted via the AANN for the NOC of the process accounted for 87% of the variance in the data as can be seen in the plot of the first two non-linear principal components in Figure 5.15. Figure 5.15 shows the four control charts, which define the NOC.



*Figure 5.15: Plot of the four controls based on the first two non-linear principal components, which defines the normal operating condition of the process*

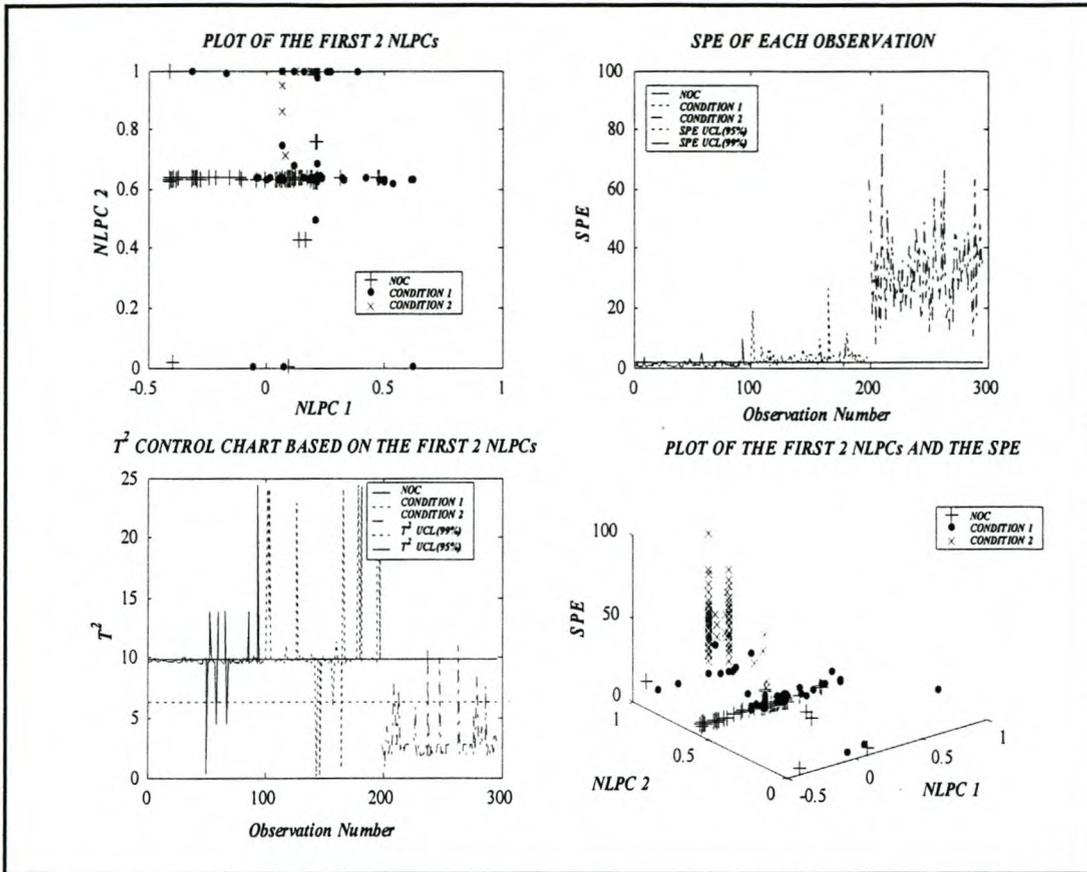
### 5.3.2.1 ONSET OF NEW EVENTS

The scores and SPE of a new observation to be plotted on the graph, which defines the NOC of the process, was obtained by simulating the sequential AANN model developed for the NOC. The AANN model yields the scores of the new observation. These scores are then fed into the feedforward backpropagation neural network model used for the reconstruction of the original data from the scores. The output of the feedforward backpropagation neural network model is then used to calculate the SPE. Table 5.23 indicates that Condition 1 and Condition 2 are deviating from the NOC.

**Table 5.23: Correlation coefficients between original data and reconstructed data**

<b>Number Of NLPCs Used in Reconstruction of Data</b>	<b>Correlation Coefficient For NOC</b>	<b>Correlation Coefficient For Condition 1</b>	<b>Correlation Coefficient For Condition 2</b>
First NLPC	0.753	0.695	0.58
First two NLPCs	0.872	0.761	0.648

The plot of the scores of the NOC, Condition 1 and Condition 2 are very closely clustered with the exception of a few points of the NOC and Condition 1 as can be seen in the plot of the first two non-linear principal components in Figure 5.16. From this plot it is difficult to determine if Condition 1 and 2 deviate from the NOC of the process.



*Figure 5.16: Plot of the four control charts based on the first two non-linear principal components for the normal operating condition, Condition 1 and Condition 2*

Table 5.24 gives the total SPE associated with each variable over the ninety-nine observations used to define the NOC of the process. For variable one, the total SPE is 26. The SPE of 26 is obtained by summing the SPE of for each of the ninety-nine observations for variable one. The SPE was calculated in a similar way for the remaining four variables.

**Table 5.24: SPE associated with each variable defining the NOCs of the process**

Variable	SPE of the NOC	SPE of Condition 1	SPE of Condition 2
Variable 1	26	62	1874
Variable 2	23	44	154
Variable 3	35	117	727
Variable 4	20	62	298
Variable 5	13	56	134

From Table 5.24 it can be seen that the SPE of Condition is higher than that of the NOC for all variables indicating that the Condition 1 is deviating from the NOC. The SPE of all variables of Condition 2 indicates that it has totally deviated from the NOC. This is verified from the SPE control chart in Figure 5.16, which shows that all observations of Condition 2 lie outside the 99% upper control limit. the plot of the first two non-linear principal components and the SPE shows how Condition 2 has completely deviated from the NOC. With this chart it is not possible to detect that Condition 1 is slowly deviating from the NOC.

## 5.4 SUMMARY

MSPC methods have been presented for the analysis and on-line monitoring of floatation processes based on image data of the froth. The only information needed to develop these methods is a historical database of images of the ideal froth. Control charts and their control limits are developed using information from the historical database of images of the ideal froth. Therefore, this approach relies upon the idea that future images of the ideal froth should have similar behaviour as those on which the control charts have been developed. The control charts used in this study for MSPC can be easily displayed, interpreted and are able to quickly detect a change in process conditions.

Conventional linear principal component models and principal component models via AANNs were developed for MSPC using 2-phase and 3-phase froth images. Control charts of the SPE, Hotellings  $T^2$  statistic, first two principal components and a control chart based on the first two principal components and the SPE were used to detect process deviations. The conventional linear principal component models as well as the principal component models via AANNs were able to detect both process deviations on all four-control charts for the 2-phase froth images. For the 3-phase froth images, the conventional linear principal component model was unable to detect the two process deviations on any of the four control charts. The AANN with one hidden layer (Linear activation function) was only able to detect the two process deviations on the  $T^2$  control chart. The AANN with one hidden layer (Non-linear activation function) was able to detect the two process deviations on the  $T^2$  and the SPE control chart whereas the AANN with three hidden layers (Non-linear activation functions) was only able to detect the two process deviations with the SPE control chart.

MSPC performed on the 2-phase froth images using AANNs indicated more clearly the two process deviations than the conventional linear principal component model. With the 3-phase froth images the conventional linear principal component model was unable to detect any of the two process deviations on all four control charts whereas the AANN models were able to detect the two process deviations on the SPE and  $T^2$  control chart.

Therefore it is proposed that AANNs be used to perform MSPC based on froth image data.

The  $T^2$  control charts based on conventional linear principal component analysis performed better than the  $T^2$  control charts based on NLPCA using AANNs. This is due to the calculation of Hotelling's  $T^2$  statistic. The  $T^2$  statistic is usually calculated from data obtained from conventional linear principal component analysis, which is based on standardised data. The data used to perform PCA using AANNs was unstandardised. Therefore the  $T^2$  statistic calculated from this data is not accurate. Another reason for the poor performance of these control charts has to do with the calculation of the 95 and 99% upper control limits. The calculation of the upper control limits of the  $T^2$  control chart is inappropriate in the non-linear case since a non-linear mapping does not necessarily guarantee that the generated data will follow a normal distribution.

It is important to remember what the MSPC charts are detecting when an observation lies outside the normal operating condition of the process. Each observation is a sub-image (31\*31 pixels) of a larger image. Therefore, if a sub-image lies outside the normal operating of the process it means that a small portion of the froth has changed from the ideal froth. If a number of sub-images lie outside the normal operating condition, it means that the froth texture and structure are changing implying that an incipient process deviation is about to be taking place.

A reason for the SPE and  $T^2$  control charts looking so spiky has to do with the pre-processing of the images. The data sets used for the normal operating condition, Condition 1 and Condition 2 were formed by taking a number of images (248\*248 pixels) of a class, and splitting them into sub-images and then randomly grouping the sub-images. This was done in order to train the AANN models in Chapter 4.

The implications of the findings of chapter five are:

- MSPC using AANNs are able to detect changes in a laboratory flotation process. This creates the platform for further investigations in this area on a larger scale, for example on a pilot plant level.
- If the system succeeds on a pilot plant scale for monitoring of flotation systems the design can be extended to complement and even replace some modules of conventional control systems in flotation systems because of its simplicity.
- The multivariate statistical approach using AANNs will be much easier to develop for flotation systems than traditional approaches to monitoring because of the ready availability of routine operating data.

## CHAPTER 6

### 6 CONCLUSIONS

This chapter presents a review of the work done in the course of this thesis. It provides a summary of the important discoveries that were made during this study.

The objective of this study was the development of conventional linear principal component models and principal component models using AANNs, using image data acquired from the surface froth of flotation cells. These models were to be used to extract features from the froth images. The extracted features were to be used for classification and MSPC.

This thesis started with a short background for the motivation of a machine vision system for monitoring and control of flotation processes. An overview and history of other work on specifically the image processing based control of the flotation process was given. It was concluded that current machine vision algorithms, for image analysis of the froth surface perform adequately in real time but they are not optimal in certain areas. The reasons for this are due to their high computational cost, and their inability to analyse fine froths effectively. A machine vision system, which could analyse fine froths effectively

would have significant benefits for industrial floatation plants if the above-mentioned limitations of present machine vision systems could be overcome.

To overcome the problems of computational expensive algorithms for image analysis and fine froths, this work has proposed the use of principal component analysis for the extraction of features from froth images and the use of these extracted features for MSPC of floatation processes in real-time and for the classification of froth images. Three techniques for the performance of principal component analysis on froth images have been evaluated. The first technique evaluated was conventional linear principal component analysis, the second technique was principal component analysis using AANNs and the third technique was local principal component analysis. Three types of AANNs were used to perform principal component analysis. Principal component analysis was performed on eight classes of froth images using the three principal component analysis techniques.

Froth images of resolution 248 x 248 pixels were used as the basis for principal component extraction. Since each pixel represent a variable, subimages of 31 x 31 pixels were used, otherwise the computational burden was intractable. Although these subimages were the maximal size that could be processed, future work should concentrate on optimization of image size, as smaller images may also have been sufficient.

Three principal components were extracted from each of the eight classes of froth images. It was found that conventional linear principal component analysis extracted more of the variance in the data with the first linear principle component than that of the PCA using AANNs and local principal component analysis for all classes of froth images except for froth images of class C3. The AANNs were able to capture more of the variance with the first three principal components, from each of the eight classes of froth images, than conventional linear principal component analysis. Local principal component analysis extracted the least amount of variance with the first three principal components for all eight classes of images. The reason for this was due to present

clustering techniques not been able to effectively cluster data of high dimensionality. Clustering of the data is an important step in local PCA thus if the data is not properly clustered the conventional LPCA of each cluster yields poor results.

The AANN with three hidden layers (Non-linear activation functions) extracted the best features from all eight classes of froth images. This was concluded from the distinct clusters each class of images was grouped into, on a plot of the first three principal components. This distinct clustering of each class of images was not evident from the principal component plots of the other AANNs.

Conventional linear principal component analysis was able to capture a little more of the variance in the data with the first principal component compared to that of PCA using AANNs. PCA using AANNs was able to capture more of the variance in the data with the first three principal components compared to that of conventional linear principal component analysis. The features extracted by PCA using AANNs are more unique than that of conventional LPCA. Conventional LPCA has an advantage over PCA using AANNs in that no optimisation is required, since the calculations are essentially based on linear algebra, but the AANN approach is well suited to the processing of very large data sets which is encountered when capturing features from froth images in real time. It is therefore recommended that an AANN with three hidden layers (Non-linear activation function) be used in extracting features from froth images.

The extracted features from the three AANNs were used for the development of a neural network classifier for the classification of froth images. Two classifiers were designed using two neural network architectures. The first classifier designed was a probabilistic neural network classifier which was based on radial basis networks while the second classifier was designed using a single hidden layer feedforward backpropagation neural network classifier. It was shown that the performance of the classifiers were highly dependant on the types of features used as inputs to the classifiers. Classifiers trained on features extracted from the AANN with three hidden layers (Non-linear activation function) gave a 100% correct classification. The probabilistic neural network classifier

proved to be more robust than the feedforward backpropagation neural network classifier in classifying features extracted by the three AANNs from froth images.

MSPC was performed using conventional linear principal component models and principal component models using AANNs. These models extracted features from froth images and the extracted features were used to detect non-conforming process behaviour. MSPC charts based on Hotellings  $T^2$  statistic, SPE and the first two principal components were used to detect process deviations. Features obtained from froth images by AANN models were shown to be able to detect process deviations more clearly than features extracted from froth images by conventional linear principal component models. Therefore it has been shown that that conventional linear principal component analysis as well as PCA using AANNs perform adequately in been able to monitor flotation processes based on image data.

One of the difficulties in using AANNs for PCA is the training involved in obtaining good models. A large amount of time has to be spent in training the AANNs in order to develop good models. Another problem associated with training of the AANNs is that a large number of training exemplars are required in order to generate models which have a good generalisation ability. In this study a large amount of time was spent in training the AANNs for the extraction of the first principal component whereas little time was spent in training the AANNs for the extraction for the second and third principal components for all eight classes of froth images. It is therefore believed that if the amount of time spent on training the AANNs for the extraction of the second and third principal component was increased, more of the variance in the froth images would be able to be extracted. Due to the computational limitations of the computer being used for this study all available froth images could not be used to train the AANNs. It is a well known fact that the greater the number of exemplars used in training a neural networks the better is its ability to generalise.

One limitation in using an AANN with three hidden layers (Non-linear activation function) is that it cannot model curves that intersect with themselves. Control limits

could not be inserted on the two dimensional principal component plots due to the data been unstandardised. The calculation of the control limits for the  $T^2$  control charts for the non-linear principal component models are not accurate due to the fact that non-linear mappings do not necessarily guarantee that the generated data will follow a normal distribution. This problem can be addressed by calculating the control limits using the non-parametric technique of kernel density estimation. This approach has the advantage that no a priori assumption of normality is required. The calculation of the  $T^2$  statistic used in the  $T^2$  control charts of the AANN models have not given good results due to the data been unstandardised.

Moreover, in this thesis the training and validation data were all derived from the same set of original images (albeit different subimages). In future work, more random sampling, with regard to time should be considered.

In summary, the findings in this study they are as follows:

- The advantage of the effectiveness of AANNs in performing linear and non-linear principal component analysis has been illustrated.
- This work has created a platform from which to further pursue the viability of a machine vision system based on principal component analysis. The next step would be to test the algorithms on a pilot plant scale and if they prove successful there would be the potential for implementation at an industrial level. It must be emphasised that before this stage can be reached there is a lot of work still to be done.
- The approach focuses on simplicity, as opposed to complex strategies of conventional control systems.
- The merit of the proposed system as a monitoring device is that it is able to detect incipient process deviations.
- A machine vision system based on principal component analysis will enhance the ability of the operator in detecting disturbances, which will then allow him to make the necessary set point changes. A froth classifier based on the features

extracted by principal component analysis can be installed on a flotation plant and will highly increase the operators ability to control the flotation process on-line.

- If the system succeeds in the monitoring phase the design can be extended to complement and even replace some modules of conventional control systems because of its simple and robust features.
- The multivariate statistical monitoring tools presented in this study have proven to be useful in monitoring flotation on a laboratory scale. Therefore this study creates the basis to analyse if these techniques are not applicable on a larger scale.

It was not shown conclusively that PCA via AANNs performs better in extracting features from images of the surface froth of flotation systems than conventional LPCA, based on an least squares criterion. However, the AANN features appeared to cluster the data better than linear PCA. A probabilistic neural network classifier better classifies the features extracted by PCA than a feedforward backpropagation neural network classifier. MSPC using conventional linear principal component models and principal component models via AANNs, are able to monitor froth flotation systems.

## REFERENCES

**Aldrich, C.**, 1998-1999, Introduction to Exploratory Data Analysis and Empirical Modelling, University of Stellenbosch.

**Aldrich, C., Moolman, D.W., Eksteen, J.J. & Van Deventer, J.S.J.**, 1995, Characterisation of Flotation Processes with Self-Organising Neural Nets, Chemical Engineering Communications, 39, 25-39.

**Banford, A.W., Atkas, Z. & Woodburn, E.T.**, 1998, Interpretation Of The Effect Of Froth Structure On The Performance Of Froth Flotation Using Image Analysis, Powder Technology, 98, 61-73.

**Basso, A. & Kunt, M.**, 1992, Autoassociative Neural Networks for Image Compression, European Transactions on Telecommunications and Related Technologies, 3 (6), 593-598.

**Botha, C.P.**, 1999, An On-Line Machine Vision Flotation Froth Analysis Platform, Master's Thesis, University of Stellenbosch.

**Bourland, H. & Kamp, Y.**, 1988, Auto-associative by Multilayer Perceptrons and Singular Value Decomposition, Biological Cybernetics, 59, 291-294.

**Caudill, M.**, 1990, Neural Networks go Biotech, PC AI, 4(6), 54-55.

**Chang, K. & Ghosh, J.**, 2001, A Unified Model For Probabilistic Principal Surfaces, IEEE Transactions On Pattern Analysis And Machine Intelligence, 23 (1), 22-41.

- Chen, S., Cowan, C.F.N. & Grant, P.M.**, 1991, Orthogonal Least Squares Learning Algorithm For Radial Basis Functions Networks, *IEEE Transactions on Neural Networks*, 2 (2), 302-309.
- Chen, G. & Macvoy, T.J.**, 1998, Predictive On-line Monitoring of Continuous Processes, *Journal Process Control*, 8(5-6), 409-420.
- Cipriano, A., Guarini, M., Vidal, R., Soto, A., Sepulveda, C., Mery, D. & Briseno, H.**, 1998, A Real-time Visual Sensor For Supervision Of Flotation Cells, *Minerals Engineering*, 11(6), 489-499.
- Connors, R.W & Harlow, C.A.**, 1980, A Theoretical Comparison of Texture Algorithms, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-2 (3), 204-222.
- Cottrell, G.W., Munroe, P. & Zisper, D.**, 1987, Image compression by neural networks: An example of extensional programming, Technical report, ICS report 8702 University of California at San Diego.
- Cybenko, G.**, 1989, Approximation by superpositions of a sigmoidal function, *Math. Contr. Signals. Syst.*, 2, 303-314.
- De Mers, D. & Cottrell, G.W.**, 1993, Non-linear dimensionality reduction, *Neural Inform. Proc. Syst.*, 5, 502-516.
- Dong, D. & Mcavoy, T.J.**, 1994, Nonlinear Principal Component Analysis based on principal curves and neural networks, *Computers and Chemical Engineering*, 20(1), 65-78.
- Dong, D. & Mcavoy, T.J.**, 1996, Batch Tracking via Nonlinear Principal Component Analysis, *ALCHE Journal*, 42(8), 2199-2208.

**Dougherty, E.R.**, 1992, An introduction to Morphological Image Processing, SPIE Optical Engineering Press.

**Eklundh, J.O.**, 1979, On the Use of Fourier Phase Features for Texture Discrimination, Computers Graphics and Image Processing, 9, 199-201.

**Eksteen, J.J.**, 1995, The Development and Technology Transfer of an Industrial MachineVision System for the control of a Platinum Flotation Plant, Master's thesis, University of Stellenbosch.

**Ellis, R.**, 2001, , Master's thesis, University of Stellenbosch.

**Fukunaga, K. & Koontz, W.**, 1970, Application of Karhunen-Loeve Expansion to Feature Selection and Ordering, IEEE Transactions & Computing, C-19, 311.

**Gordon, A.D.**, 1999, Classification, 2<sup>nd</sup> ed, USA, Chapman & Hall/CRC.

**Hargrave, J.M., Miles, N.J. & Hall, S.T.**, 1996, the use of grey level measurement in predicting coal flotation performance , Minerals Engineering, 9(6), 667-674.

**Haralick, R.M., Shanmugan, K. & Dinstein, I.**, 1973, Textural features for image classification, IEEE Transactions on Systems, Man, and Cybernetics, SMC-3 (6), 610-621.

**Haralick, R.M.**,1979, Statistical and Structural Approaches to Texture, Proceedings of the IEEE, 67(5), 786-803.

**Hastie, T. & Steutzle, W.**, 1989, Principal Curves, J. Amer. Stat. Assoc., 84, 502-516.

**Hytyniemi, H. & Ylinen, R.**, 2000, Modelling Of Visual Flotation Froth Data, Control Engineering Practise, 8, 313-318.

- Jia, F., Martin, E.B. & Morris, A.J.**, 1998, Non-Linear Principal Component Analysis for Process Fault Detection, *Computers and Chemical Engineering*, 22, s851-s854.
- Johnson, R.A. & Tsui, K.W.**, 1998, *Statistical reasoning and methods*, New York, John Wiley & Sons.
- Jolliffe, I.T.**, 1986, *Principal Component Analysis*, New York, Springer-Verlag.
- Kramer, M.A.**, 1991, Non-linear Principal Component Analysis Using Autoassociative Neural Networks, *ALCHE Journal*, 37, 233.
- Kresta, J., MacGregor, J.F., Marlin, T.E.**, 1991, Multivariate statistical monitoring of process Operating Performance, *Canadian Journal of Chemical Engineering*, 69, 35-47.
- Lane, S., Martin, E.B., Kooijmans, R., Morris, A.J.**, 2001, Performance monitoring of a multi-product semi-batch process, *Journal of Process Control*, 11, 1-11.
- Macgregor, J.F. & Kourti, T.**, 1995, Statistical process control of multivariate processes, *Control Eng. Practice*, 3(3), 403-414.
- Manly, B.F.J.**, 1994, *Multivariate Statistical Methods*, 2<sup>nd</sup> ed, USA, Chapman & Hall/CRC.
- Malthouse, E.C.**, 1998, Limitations of Nonlinear PCA as Performed with Generic Neural Networks, *IEEE Transactions on Neural Networks*, 9(1), 165 – 173.
- Martin, E.B. & Morris, A.J.**, 1996, Non-Parametric Confidence Bounds For Process Performance Monitoring Charts, *Journal Of Process Control*, 6(6), 349 – 358.
- Mason, R.L. & Young, J.C.**, 1998, Hotellings  $T^2$ : A multivariate statistic for industrial process control, *Annual Quality Congress Transactions*, 78-85.

**Massacci, P.**, 2000, The Use Of An Image Processing Based Sensor For On-Line Analysis Of Flotation Performance, Rome-Italy, Elsevier, B8a 158-B8a 162.

**Monahan, A.H.**, 2000, Non-Linear Principal Component Analysis by Neural Networks: Theory and Application to the Lorenz System, *Journal of Climate*, 13, 821-835.

**Moolman, D.W.**, 1995, The Interpretation of Flotation Froths by using Computer vision and Connectionist Networks, Doctorate Thesis, University of Stellenbosch.

**Moolman, D. W., Aldrich, C. & Van Deventer, J.S.J.**, 1995a, The monitoring of froth surfaces on industrial flotation plants using connectionist image processing techniques, *Minerals Engineering*, 8(1/2), 23-30.

**Moolman, D.W., Aldrich, C., Van Deventer, J.S.J. & Bradshaw, D.J.**, 1995b, The interpretation of flotation froth surfaces by using digital image analysis and neural networks, *Chemical Engineering Sciences*, 50(22), 3501-3513.

**Moolman, D.W., Aldrich, C., Van Deventer, J.S.J. & Stange, W.W.**, 1995c, The classification of froth structures in a copper flotation plant by means of a neural net, *International Journal of Mineral Processing*, 43, 193-208.

**Moolman, D.W., Aldrich, C., Schmitz, G.P.J. & Van Deventer, J.S.J.**, 1996a, The interrelationship between surface froth characteristics and industrial flotation performance, *Minerals Engineering Sciences*, 9(8), 837-854.

**Moolman, D.W., Eksteen, J.J., Aldrich, C. & Van Deventer, J.S.J.**, 1996b, The significance of flotation froth appearance for machine vision control, *International Journal of Mineral Processing*, 48, 135-158.

**Morris, A.J.**, 1994, Neural networks for process control, Talk to NSYN Launch Event, 29 September.

- Niemi, A.J., Ylinen, R. & Hytoyniemi, H., 1997, On Characterisation Of Pulp And Froth In Cells Of Flotation Plant, *International Journal of Mineral Processing*, 51, 51-65.
- Nijhuis, A., De Jong, S. & Vandeginste, B.G.M., 1997, Multivariate statistical process control in chromatography, *Chemometrics and intelligent laboratory systems*, 38, 51-62.
- Nomikos, P. & Macgregor, J.F., 1994, Monitoring Batch Processes Using Multiway Principal Component Analysis, *ALCHE Journal*, 40(8), 1361 – 1375.
- Nomikos, P. & Macgregor, J.F., 1995, Multivariate SPC Charts for Monitoring Batch Processes, *Technometrics*, 37(1), 41 – 59.
- Raich, A. & Cinar, A., 1996, Statistical Process Monitoring And Disturbance Diagnosis In Multivariate Continuous Processes, *ALCHE Journal*, 42(4), 995 – 1009.
- Rios, A. & Kabuka, M.R., Image Compression with A Dynamic Autoassociative Neural Network, 503 – 510.
- Sader-Kazemi, N. & Cilliers, J. J., 1997, An image processing algorithm for measurement of flotation froth bubble size and shape distributions, *Minerals Engineering*, 10(10), 1075 – 1083.
- Shao, R., Jia, F., Martin, E.B. & Morris, A.J., 1999, Wavelets And Non-Linear Principal Components Analysis For Process Monitoring, *Control Engineering Practice*, 7, 865 – 879.
- Siew, L.H., Hodgson, R.M. & Wood, E.J., 1988, Texture measures for carpet wear assessment, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(1), 31-42.

**Sun, C. & Wee, W.G.**, 1983, Neighbouring Grey Level Dependence Matrix for Texture Classification, *Computer Vision, Graphics, and Image Processing*, 23, 341 – 352.

**Swingler, K.**, 1996, *Applying neural networks: A practical guide*, Great Britain, Hartnolls Ltd.

**Symonds, P.J. & De Jager, G.**, 1992, A technique for automatically segmenting images of the surface froth structures that are prevalent in industrial flotation cells, *Proceedings of the South African Symposium on Communications and Signal Processing*, University of Cape Town, Rondebosch, South Africa, Friday 11<sup>th</sup> September, 111-115.

**Tan, S. & Mavrovouniotis, M. L.**, 1995, Reducing Data Dimensionality through Optimising Neural Network Inputs, *AIChE Journal*, 41(6), 1471-1480.

**Van Gool, L., Dewaele, P. & Oosterlinck, A.**, 1983, Texture Analysis Anno, *Computer Vision, Graphics, and Image Processing*, 29, 336-357.

**Vincent, L.**, 1993, Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms, *IEEE Transactions on Pattern Analysis and Machine intelligence*, 2, 176-201.

**Vincent, L. & Soille, P.**, 1991, Watersheds in digital spaces: An efficient algorithm based on immersion simulations, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13, 583-598.

**Weszka, J.S., Dyer, C.R. & Rosenfeld, A.**, 1976, A Comparative Study Of Texture Measures For Terrain Classification, *IEEE Transactions on systems, Man and Cybernetics*, SMC- 6(4), 269- 285.

**Woodburn, E.T., Stockton, J.B. & Robins D.J.**, 1989, Vision-Based Characterisation Of Three-Phase Froths, International Colloquium-Developments in Froth Flotation, South African Institute of Mining and Metallurgy, Gordons Bay, South Africa, 1, 1-30.

**Wright, B.A., Sweet, C.G., Francis, J.J., De Jager, G., Bradshaw, D.J. & Cilliers, J.J.**, 2000, The use of image processing based sensor for on-line analysis of flotation performance, Proceedings of the XXI International Mineral Processing Congress, Rome-Italy, July 23-27, B8a-158-B8a-162

**Yoon, C. & Macgregor, J.F.**, 2000, Statistical And Causal Model-Based Approaches To Fault Detection And Isolation, ALCHE Journal, 46(9), 1813 – 1824.

**Zhang, J., Martin, E.B. & Morris, A.J.**, 1996, Non-Linear Performance Monitoring, IEE, Conference Publication No. 427, 924-929.

**Zhang, J., Martin, E.B. & Morris, A.J.**, 1997, Process Monitoring Using Non-Linear Statistical Techniques, Chemical Engineering Journal, 67, 181-189.

# **APPENDIX A: SPECIFICATIONS OF THE AUTOASSOCIATIVE NEURAL NETWORKS**

During training of three AANNs various parameters had to be fixed while others needed to be varied in order to achieve the best performance of the AANNs.

## **Fixed Parameters**

### **1) PERFORMANCE FUNCTION**

The performance function used to measure the error of the AANNs was the mean square error.

### **2) TRAINING ALGORITHM**

Various training algorithms were available to train the networks such as the Levenberg-Marquardt, One-Step-Secant, Resilient backpropagation and the various conjugate gradient algorithms. Of all the algorithms available the scaled conjugate algorithm was chosen to train the AANN. This was due to the fact that large data sets had to be trained by the AANNs and the scaled conjugate gradient algorithms computational requirements are significantly less, when compared to those of the other algorithms, because no line search is performed by the scaled conjugate gradient algorithm in each iteration during training.

### **3) ERROR GOAL**

The error goal was set to 0.001

## **VARIED PARAMETERS**

### **1) LEARNING RATE**

The learning rate was varied between 0.001 - 0.000000000001

### **2) LEARNING INCREMENT**

The learning increment was set to 1.05

### **3) EPOCHS**

The number of epochs used to train the networks was varied between 200-1000

## **INITIALISATION OF THE AUTOASSOCIATIVE NEURAL NETWORKS**

### **AANN-One Hidden Layer (Linear Activation Function)**

The two layer AANN was initialised using the `Initlay`<sup>26</sup> function. `Initlay` is a network initialisation function, which initialises each layer according to its own initialisation function. Each layer in the network was initialised using the `Initwb`<sup>27</sup> function. The `Initwb` function causes the initialisation to revert to the individual initialisation parameters for each weight matrix and bias. For feedforward networks the weight initialisation was set to random, which sets weights to random values between -1 and 1. It is normally used when the layer activation functions are linear.

<sup>26</sup> Is a function in the neural network toolbox of Matlab which is used to initialise a neural network

<sup>27</sup> Is a function in the neural network toolbox of Matlab which is used to initialise each layer of the neural network when the layers contain linear transfer functions.

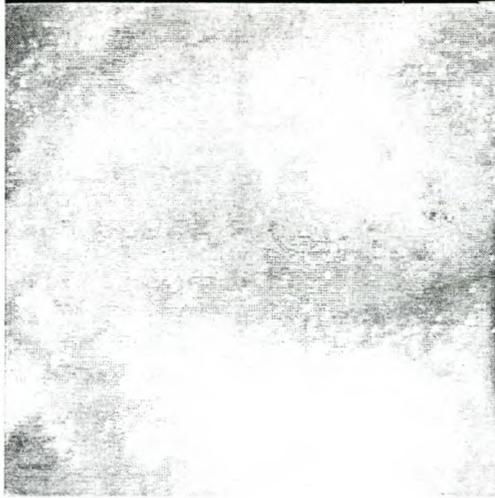
### **AANN-One Hidden Layer (Non-Linear Activation Function)**

This AANN was designed exactly the same as that in Section 3.3.1 with the exception been that the single hidden layer was initialised using the *Initnw* function. The function *Initnw* is normally used for layers of feedforward neural networks where the activation function is sigmoidal. It is based on the technique of Nguyen and Widrow and generates initial weight and bias values for a layer so that the active regions of the layer's neurons will be distributed roughly evenly over the input space. It has two advantages over purely random weights and biases:

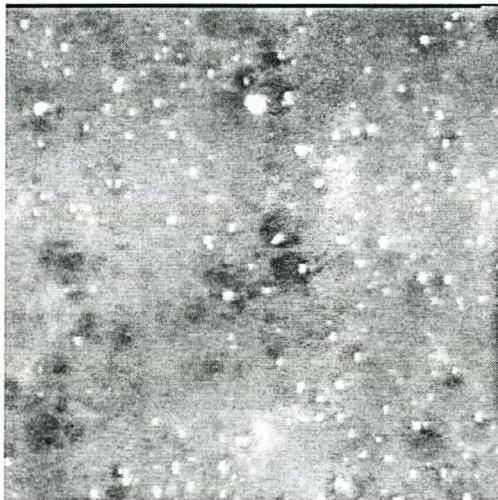
- Few neurons are wasted (since the active regions of all the neurons are in the input space).
- Training works faster (since each area of the input space has active neuron regions).

### **AANN-Three Hidden Layer (Non-Linear Activation Functions)**

The AANN is initialised in a similar manner as the AANN with one hidden layer. Layers two to four is initialised using the *Initnw* function since the activation functions in those layers are non-linear. While layer five is initialised using the *Initwb* function since the activation functions used in this layer are linear.



*Figure B.2: 2-phase froth - 100% CTAB*

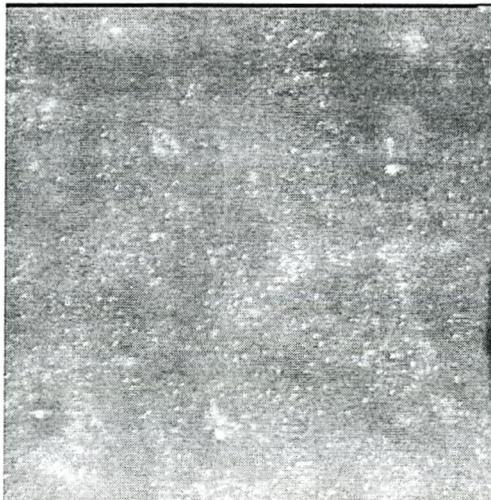


*Figure B.3: 2-phase froth - 90% CTAB*

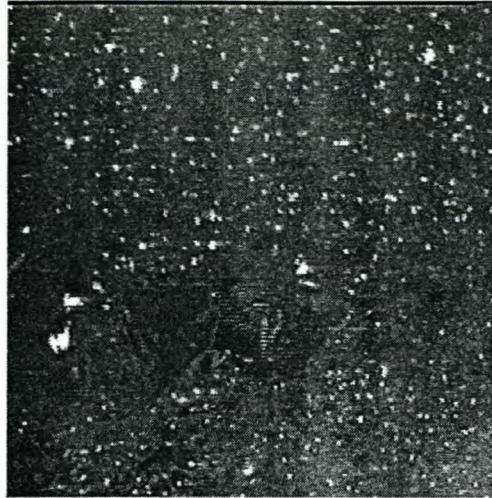
## APPENDIX B: FROTH IMAGES

Images of the two phase froths, 100% CTAB , 90% CTAB, 80% CTAB and three phase froths, 100% CTAB quartz, 90% CTAB quartz and 80% CTAB quartz

### 2-PHASE FROTHS



*Figure B.1: 2-phase froth - 100% CTAB*

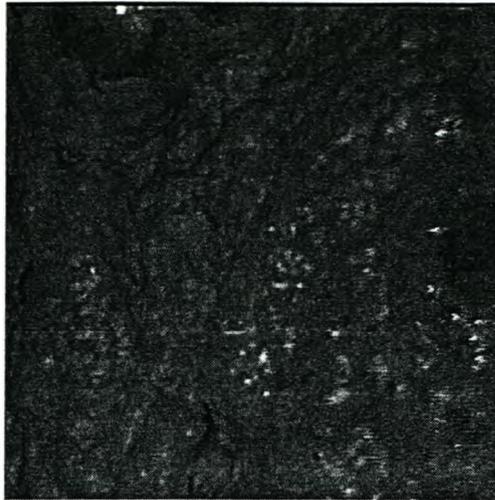


*Figure B.4: 2-phase froth - 80% CTAB*

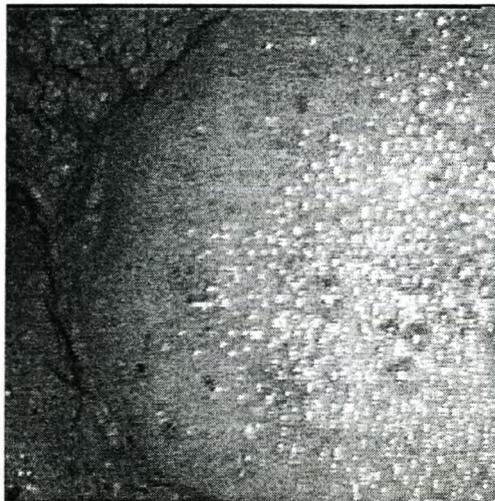
### 3-PHASE FROTHS



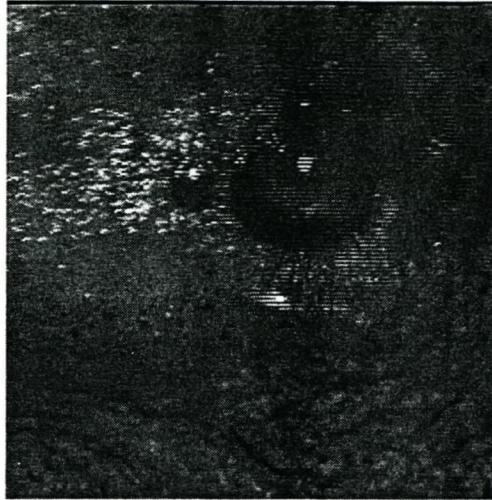
*Figure B.5: 3-phase froth - 100% CTAB, quartz*



*Figure B.6: 3-phase froth - 90% CTAB, quartz*



*Figure B.7: 3-phase froth - 90% CTAB, quartz*



*Figure B.8: 3-phase froth - 80% CTAB, quartz*