

**ADVANCED PROCESS MONITORING
USING WAVELETS AND
NON-LINEAR
PRINCIPAL COMPONENT ANALYSIS**

by

STEVEN FOURIE

A dissertation submitted in partial fulfilment
of the requirements for the degree

MASTER OF ENGINEERING (CONTROL ENGINEERING)

in the

FACULTY OF ENGINEERING

UNIVERSITY OF PRETORIA

April 2000

ADVANCED PROCESS MONITORING USING WAVELETS AND NON-LINEAR PRINCIPAL COMPONENT ANALYSIS

Author: Steven Fourie
Supervisor: Prof. P. L. de Vaal
Department: CHEMICAL ENGINEERING
UNIVERSITY OF PRETORIA
Degree: Master of Engineering (Control Engineering)

SYNOPSIS

The aim of this study was to propose a nonlinear multiscale principal component analysis (NLMSPCA) methodology for process monitoring and fault detection based upon multilevel wavelet decomposition and nonlinear principal component analysis via an input-training neural network.

Prior to assessing the capabilities of the monitoring scheme on a nonlinear industrial process, the data is first pre-processed to remove heavy noise and significant spikes through wavelet thresholding. The thresholded wavelet coefficients are used to reconstruct the thresholded details and approximations. The significant details and approximations are used as the inputs for the linear and nonlinear PCA algorithms in order to construct detail and approximation conformance models. At the same time non-thresholded details and approximations are reconstructed and combined which are used in a similar way as that of the thresholded details and approximations to construct a combined conformance model to take account of noise and outliers. Performance monitoring charts with non-parametric control limits are then applied to identify the occurrence of non-conforming operation prior to interrogating differential contribution plots to help identify the potential source of the fault.

A novel summary display is used to present the information contained in bivariate graphs in order to facilitate global visualization. Positive results were achieved.

Acknowledgments

Keywords: Process monitoring; Fault detection; Non-linear Principal Component Analysis

SINOPSIS

Die hoofdoel van hierdie ondersoek was om 'n nuwe metode voor te stel vir nie-lineêre multivlak hoofkomponent-analise vir prosesmonitering en foutopsoring. Die beginsel is gebaseer op multivlak "wavelet"-ontbinding en nie-lineêre hoofkomponent-analise deur middel van 'n inset-verandering neurale netwerk.

Normale bedryfsdata vanaf 'n nie-lineêre industriële proses word eers vooraf verwerk om hewige geraas en beduidende uitskietpeke in die data te verwyder. Dit word gedoen deur eers die data deur middel van "wavelet"-analise te ontbind in detail- en benaderings- "wavelet"-koëffisiënte en dan die "wavelet"-koëffisiënte groter as 'n sekere limiet uit te filter. Die gefilterde "wavelet"-koëffisiënte word dan gebruik vir die hersamestelling van gefilterde details en benaderings. Die beduidende details en benaderings word gebruik as insette vir die lineêre en nie-lineêre hoofkomponent-analise-algoritmes sodat detail- en benadering-konformasie Modelle saamgestel kan word. Terselfdertyd word ongefilterde details en benaderings herkonstrueer vanaf ongefilterde detail- en benaderingskoëffisiënte wat dan gekombineer word om 'n gekombineerde konformasie Model saam te stel met die hoofdoel om geraas en uitlopers in nuwe data in ag te neem.

Werkverrigtingsmoniteringsgrafieke met nie-parametriese beheerlimiete word dan gebruik om die voorkoms van nie-konformerende of abnormale bedryf op te spoor. Nadere ondersoek mbv differensiële bydrae grafieke word gebruik om te help met die opsoring van die moontlike oorsaak van die fout.

'n Nuwe metode om die inligting in bivariate grafieke in 'n kompakte en eenvoudiger wyse voor te stel is gebruik en gee 'n beter geheelbeeld van die prosesverloop. Die geskiktheid van die moniteringstelsel is getoets op nuwe data en positiewe resultate is verkry.

Sleutelwoorde: Prosesmonitering; Foutopsoring, Nie-lineêre Hoofkomponent-Analise

Acknowledgments

I am deeply indebted to the many people who have made the completion of this work possible. Particular Professor P. L. de Vaal for his guidance, assistance, supervision and for his patience to read the whole manuscript.

I would also like to thank Sasol for their financial assistance through the bursary and for allowing me to use their data for analysis. Particular thanks to Johan Gericke, my mentor, for suggesting the topic and for his assistance.

Finally, special thanks to my brother, Tommie, the rest of my family and for my best friend, Wessel Nel, for spending several weekends without me. For Corlia, your love, patience and encouragement made this work possible.

Steven Fourie

Pretoria



If a tool could do its job
after obeying a command or its own feeling
...
neither the architects (experts) would require assistants
nor the masters slaves.

Aristotle, Politics A4, 1253b34 - 1254a1

T ABLE OF C ONTENTS



UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Chapter 1 – Introduction.....	1-1
1.1. Main objectives.....	1-1
1.2. Background.....	1-2
1.3. Data volume.....	1-2
1.4. Process shifts, drift and poor process data.....	1-2
1.5. Current and desired practice.....	1-4
1.6. Data analysis and process monitoring.....	1-5
1.7. Summary and overview.....	1-6
Chapter 2 –Abnormal situation management.....	2-1
2.1. Introduction.....	2-1
2.2. What is an abnormal situation?.....	2-2
2.3. Significance.....	2-2
2.4. Goal.....	2-2
2.5. Background: Chemical plants and control.....	2-5
2.6. Past and current work.....	2-6
2.7. Methodology.....	2-7
2.8. Sources of abnormal situations.....	2-8
2.9. The solution.....	2-10
Chapter 3 –Process Description.....	3-1
3.1. Introduction.....	3-1
3.2. Objective.....	3-1
3.3. significance of process description.....	3-2
3.4. Scenarios.....	3-3
3.5. Background Process Information.....	3-5

3.5.1. Process Objective.....	3-5
3.5.2. Process Overview.....	3-5
3.6. Control Objective.....	3-6
3.7. Header Functional Description.....	3-7
3.7.1. 43 Bar Header, Letdown to 8 Bar and 4 Bar Headers.....	3-7
3.7.2. 8 Bar Header, Letdown from 40 Bar and to 4 Bar Headers.....	3-7
3.7.3. 4 Bar Header, Letdown from 40 Bar and Venting to Atmosphere.....	3-9
3.8. Process B Reactor.....	3-10
3.8.1. General.....	3-10
3.8.2. Operation.....	3-10
3.8.3. Reactor Coolant System.....	3-10
3.9. Process A Reactors.....	3-10
3.9.1. Process A Reactors and Universal Equipment.....	3-11
3.9.2. Process A Reactor Trains.....	3-11
3.10. Steam Relationships.....	3-14
3.11. Process Variables.....	3-14
Chapter 4 –Software Development & Installation.....	4-1
4.1. Introduction.....	4-1
4.2. To get started.....	4-2
4.3. The Program.....	4-3
4.3.1. Database.....	4-3
4.3.2. Introduction display.....	4-5
4.3.3. Main interface.....	4-5

Chapter 5 – Data Setup.....	5-1
5.1. Introduction.....	5-1
5.2. Data Features.....	5-1
5.3. Data Setup Interface.....	5-1
5.4. Data Viewer.....	5-3
5.5. Normalisation and Standardisation.....	5-4
5.6. Statistics Viewer.....	5-5
5.7. Application.....	5-6
Chapter 6 – Wavelet Analysis.....	6-1
6.1. Introduction.....	6-1
6.2. Previous work on Feature Extraction of Dynamic Transients.....	6-1
6.3. What is a Wavelet?.....	6-3
6.4. Wavelet Analysis Methodology.....	6-5
6.5. The Discrete Wavelet Transform.....	6-6
6.5.1. Introduction.....	6-6
6.5.2. One-Stage Filtering: Approximations and Details.....	6-6
6.5.3. Multiple-Level Decomposition.....	6-8
6.5.4. Wavelet Reconstruction.....	6-9
6.5.5. Reconstruction Approximations and Details.....	6-9
6.5.6. Filters Used to Calculate the DWT and IWT.....	6-11
6.6. Wavelet Denoising through Thresholding.....	6-12
6.7. Algorithms.....	6-14
6.8. On-Line Multiscale Rectification.....	6-16
6.9. Practical Issues of OLMS.....	6-18

6.9.1. Value of threshold.....	6-18
6.9.2. Depth of Decomposition.....	6-18
6.9.3. Selected Wavelet Filter.....	6-19
6.10. Application.....	6-19
6.10.1. Software Setup.....	6-19
6.10.2. Experimental.....	6-24
6.10.2.1. <i>Wavelet Analysis</i>	6-24
6.10.2.2. <i>Multiresolution Decomposition</i>	6-25
6.10.2.3. <i>Wavelet Thresholding</i>	6-27
6.10.2.4. <i>Multilevel Signal Reconstruction</i>	6-27
Chapter 7 – Linear Principle Component Analysis.....	7-1
7.1. Introduction to Linear Principal Component Analysis.....	7-1
7.2. Introduction to Multiscale PCA (MSPCA).....	7-2
7.3. Methodology of MSPCA.....	7-3
7.4. Principle of LPCA.....	7-6
7.5. Characteristic Roots and Vectors.....	7-6
7.6. The Method of Principal Components.....	7-7
7.7. Some Properties of Principal Components.....	7-8
7.7.1. Transformations.....	7-8
7.7.2. Interpretation of Principal Components.....	7-8
7.7.3. Generalized Measures and Components of Variability.....	7-8
7.7.4. Correlation of Principal Components and Original Variables.....	7-9
7.8. Scaling of Data.....	7-10
7.8.1. Introduction.....	7-10

7.8.2. Data as Deviations from the Mean: Covariance Matrices.....	7-10
7.8.3. Data in Standard Units: Correlation Matrices.....	7-11
7.9. Using Principal Components in Quality Control.....	7-11
7.9.1. Type I Errors.....	7-11
7.9.2. Goals of Multivariate Quality Control.....	7-11
7.10. Selecting the Number of Principal Components.....	7-12
7.10.1. Introduction.....	7-12
7.10.2. A Simple Cross-validation Procedure.....	7-13
7.10.3. Enhancements.....	7-15
7.11. Application.....	7-16
7.11.1. Software Setup.....	7-16
7.11.2. Experimental.....	7-20
Chapter 8 – Input Training Neural Networks.....	8-1
8.1. Introduction.....	8-1
8.2. The Backpropagation Algorithm.....	8-3
8.2.1. General Backpropagation.....	8-3
8.2.2. Backpropagation Applied to IT-Nets.....	8-6
8.3. Levenberg Marquardt.....	8-7
8.4. Concept of Input Training.....	8-10
8.5. Training IT-Nets.....	8-12
8.6. Input Training.....	8-12
8.7. Testing and Using IT-Nets.....	8-13
Chapter 9 – Nonlinear Principal Component Analysis.....	9-1
9.1. Introduction.....	9-1

9.2. Nonlinear Principal Component Analysis.....	9-1
9.3. Application.....	9-3
9.3.1. Software Setup.....	9-3
9.3.2. Experimental.....	9-8
Chapter 10 – Process Monitoring.....	10-1
10.1. Introduction.....	10.1
10.2. Interpretation.....	10.2
10.3. Action and Warning Limits.....	10-3
10.4. Non-Parametric Bounds.....	10-4
10.5. Detection Limit Adjustment for On-Line Monitoring.....	10-5
10.6. Residual Analysis.....	10-6
10.7. Biplots.....	10.7
10.8. Hotellings T2 Statistic: An Overall Measure of Variability.....	10-7
10.9. The Q-Statistic.....	10-10
10.10. Contribution Plots.....	10-11
10.11. Bivariate Summary Plots.....	10-12
10.12. Application.....	10-14
10.12.1. Software Setup.....	10-14
10.12.2. Experimental Data.....	10-19
10.12.3. Evaluation of Compared Models.....	10-20
10.12.4. Experimental.....	10-23
Chapter 11 – Conclusions.....	11-1
11.1. Summary.....	11-1
11.2. Further Development.....	11-2
11.3. Practical Implications.....	11-2

LIST OF SYMBOLS

A	approximation in wavelet multiresolution analysis
a	wavelet dilation parameter
a_0, b_0	discrete wavelet transform parameters
b	Bias / wavelet translation parameter
D	detail in wavelet multiresolution analysis
$DWTf$	discrete wavelet coefficient
e	Vector of network errors
E	Sum of squares function
$f(t)$	a function in the time domain
g	Current gradient (propagated error) at current node
$g(k)$	the k th wavelet synthesis filter
H	wavelet analysis filter
$h(k)$	the k th wavelet analysis filter
I	identity matrix
J	Jacobian
k	Number of principal components retained
l	Characteristic roots
L	Diagonal Matrix
m, n	discrete wavelet transform parameters
n	Number of samples per variable
p	Number of correlated variables
r	Correlation coefficient
S	Covariance Matrix

t	Original network output / time
\mathbf{u}	Characteristic vectors / Eigenvectors
V	Input weights
w	Network weights
W	Network weights
\mathbf{W}	weight matrix
x	Input data point to Neural Network
\mathbf{x}	Input vector to Neural Network / Original variable
$\bar{\mathbf{x}}$	Mean of \mathbf{x}
z	Output data (IT-net approximation) / Uncorrelated variables

Greek and other symbols

δ	Propagated error at hidden layer
α	Learning rate
ϕ	Nonlinear mapping function relating NN outputs to inputs / wavelet scale function or orthogonal function
σ	Sigmoidal transfer function
σ	standard deviation
ψ	wavelet function
$\ \cdot\ $	Euclidean norm



Subscripts

p	p th training sample
n	Number of observed variables
k	Observed variable number
j	j th hidden node
T	Transpose
$\hat{}$	estimation

LIST OF FIGURES

Figure No.	Figure Title	Pg.
Chapter 1		
Figure 1.1.	On-line process monitoring activity.....	1-6
Figure 1.2.	The NLMSPCA methodology.....	1-7
Chapter 2		
Figure 2.1.	Control without Abnormal Situation Management.....	2-5
Figure 2.2.	The ASM solution structure.....	2-12
Figure 2.3.	Control with Abnormal Situation Management.....	2-14
Chapter 3		
Figure 3.1	Process schematic of the most important process plants	3-6
Figure 3.2.	Steam distribution system	3-8
Figure 3.3.	Process B reaction, cooling and recycle loop.....	3-12
Figure 3.4.	Nonlinear relationships between PG, steam production and steam export	3-13
Figure 3.5.	Differential relationships between PG feed steam export	3-15
Chapter 4		
Figure 4.1	Setup progress display.....	4-2
Figure 4.2.	Path setup success display.....	4-3
Figure 4.3.	Saving the path.....	4-3
Figure 4.4.	Database Setup display.....	4-4
Figure 4.5.	Database creation success display.....	4-4

Figure 4.6. Introductory display..... 4-5

Figure 4.7. Main Interface..... 4-6

Chapter 5

Figure 5.1 Data Setup Interface..... 5-2

Figure 5.2. Open File Interface..... 5-3

Figure 5.3. Data Viewer Interface..... 5-4

Figure 5.4. Standardised data..... 5-5

Figure 5.5. Statistics Viewer: Correlation Coefficient for Training data set..... 5-6

Figure 5.6. Statistics Viewer: Correlation Coefficient for Testing/Validation
data set 5-7

Figure 5.7. Plot of Variables Representing Normal Operation on the same
Axes..... 5-7

Figure 5.8. Individual Standardised Plot of Variables Representing Normal
Operation..... 5-8

Chapter 6

Figure 6.1 A comparison of the sine wave and daubechies 5 wavelet..... 6-4

Figure 6.2 Time-Scale Characteristics of Wavelets..... 6-4

Figure 6.3 Basic Discrete Wavelet Filtering..... 6-7

Figure 6.4 Wavelet decomposition without downsampling, and with
downsampling..... 6-7

Figure 6.5 Multilevel decomposition tree (An octave band non-subsampled
filter bank.)..... 6-8

Figure 6.6 First-level reconstruction of approximation..... 6-10

Figure 6.7 First-level reconstruction of detail..... 6-10

Figure 6.8	Wavelet filter computing scheme.....	6-12
Figure 6.9	Fist step in the wavelet analysis algorithm.....	6-15
Figure 6.10	Reconstruction of the wavelet coefficients using reconstruction filters.....	6-16
Figure 6.11	Time delay introduced due to dyadic length requirement in wavelet decomposition.....	6-17
Figure 6.12	OLMS rectification.....	6-17
Figure 6.13	Wavelet analysis main interface.....	6-19
Figure 6.14	Wavelet analysis viewer – Reconstructed details and approximation.....	6-21
Figure 6.15	Wavelet analysis viewer – Wavelet coefficients.....	6-22
Figure 6.16	Completed multiresolution analysis of variable one.....	6-23
Figure 6.17	MRA data selection and viewer.....	6-24
Figure 6.18	Wavelet decomposition and separation of stochastic and deterministic components.....	6-25
Figure 6.19	Multiresolution analysis plot.....	6-26
Figure 6.20	Dataset 1 containing the approximations of all eight variables.....	6-28
Figure 6.21	Comparison between the original signal, de-noised signal and the approximation coefficients used for model derivation.....	6-29

Chapter 7

Figure 7.1	Linear PCA main interface	7-16
Figure 7.2.	Cumulative variability plot	7-17
Figure 7.3.	Proportional variability plot	7-18
Figure 7.4.	Principal Component plot	7-19
Figure 7.5.	Principal component selection interface	7-20

Figure 7.6.	First four principal components of dataset 1.....	7-21
-------------	---	------

Chapter 8

Figure 8.1.	A 2-4-1-4-2 Autoassociative Network.....	8-2
Figure 8.2.	Two-layer feedforward network	8-3
Figure 8.3.	Concept of Input Training.....	8-12

Chapter 9

Figure 9.1.	IT-Net Parameter Setup Interface.....	9-3
Figure 9.2.	IT-Net Training Parameter Interface.....	9-5
Figure 9.3.	Epoch View Interface.....	9-6
Figure 9.4.	IT-Net Simulation Interface.....	9-7
Figure 9.5.	Neural Network Mapping Interface.....	9-8

Chapter 10

Figure 10.1.	Important Components of an Industrial Performance Monitoring Assessment, and Diagnosis Strategy.....	10-1
Figure 10.2.	Histogram Plots of the Eight Variables Used for Training.....	10-3
Figure 10.3.	Traditional Bivariate Plot.....	10-13
Figure 10.4.	Bivariate Summary Plot Calculation.....	10-13
Figure 10.5.	Bivariate Summary Plot for 6 Biplots at One Time Instance.....	10-14
Figure 10.6.	Bivariate Plot Setup Interface.....	10-15
Figure 10.7.	SPE Setup Interface.....	10-16
Figure 10.8.	NLMSPCA Monitor Interface.....	10-17
Figure 10.9.	Data Representing Abnormal Operation.....	10-19
Figure 10.10.	SPE Plots for the Test Data Based on the 4-3 LMSPCA Model with 95% and 99% Non-Parametric Limits.....	10-20

Figure 10.11. Scores Plots for the Non-Conforming Test Data Based on the 4-3 LMSPCA Model.....	10-21
Figure 10.12. SPE Plots for the Test Data Based on the 4-3 NLPCA Model with 95% and 99% Non-Parametric Limits.....	10-22
Figure 10.13. Scores Plots for the Non-Conforming Test Data Based on the 4-3 NLPCA Model.....	10-22
Figure 10.14. SPE Plots for the Test Data Based on the 4-3 NLMSPCA Model with 95% and 99% Non-Parametric Limits.....	10-23
Figure 10.15. Scores Plots for the Non-Conforming Test Data Based on the 4-3 NLMSPCA Model.....	10-24
Figure 10.16. Summary Plot of the Bivariate Scores Plots Based on the 4-3 Model.....	10-25
Figure 10.17. Differential and Residual Contribution Plots to Investigate the Cause of Process Deviation in the Non-Conforming Data.....	10-25

LIST OF TABLES

Table No.	Table Title	Pg.
Chapter 3		
Table 3.1.	Process Variables Used in the Investigation.....	3-14
Chapter 5		
Table 5.1.	Process Variable Description.....	5-9
Chapter 6		
Table 6.1.	Filter Representation.....	6-12
Chapter 7		
Table 7.1.	PRESS values for Selecting the Number of PC's to Retain.....	7-14
Table 7.2.	Cumulative variability for pc's of dataset 1 and dataset 2.....	7-22
Chapter 9		
Table 9.1.	Training Parameters for IT-Net and Mapping Model.....	9-9

LIST OF DEFINITIONS

Cross-validation – Cross-validation is widely used as an automatic procedure to choose a smoothing parameter in many statistical settings. The classical cross-validation method is performed by systematically expelling a data point from the construction of an estimate, predicting what the removed value would have been and comparing the prediction with the value of the expelled point.

Covariance matrix – For a given data matrix \mathbf{X} with m rows and n columns the covariance matrix of \mathbf{X} is defined as

$$\text{cov}(\mathbf{X}) = \frac{\mathbf{X}^T \mathbf{X}}{m - 1}$$

This assumes that the columns of \mathbf{X} have been ‘mean centered’, i.e. adjusted to have a zero mean by subtracting of the original mean of each column.

Correlation matrix – Referring to the definition of covariance matrix, if the columns of \mathbf{X} have been ‘autoscaled’, i.e. adjusted to zero mean and unit variance by dividing each column by its standard deviation, the equation for calculating the covariance gives the correlation matrix of \mathbf{X} .

Details – Details, generally known as the wavelet coefficients, are coefficients that capture the details of the signal lost when moving from an approximation at one scale to the next coarser scale.

Epochs – One training cycle after which the neural network parameters (weights and biases) are updated.

Floor(x) – Rounds the elements of x to the nearest integers towards minus infinity.

FMH – Finite impulse response median hybrid filter (Heinonen and Neuvo, 1987). A FMH is a median filter which has a pre-processed input from M linear FIR filters. Thus, the FMH filter output is the median of only M values, which are the outputs of M FIR filters. FMH filters are nonlinear filters, are most effective when applied to piecewise constant signals contaminated with white noise, require careful selection of the filter length, and are limited to off-line use.

Hessian Matrix – For a given data matrix \mathbf{X} , the hessian matrix is given by Equation 8.25 (see Chapter 8).

Jacobian Matrix – The jacobian matrix is a matrix that contains the first derivatives of the network errors with respect to the weights and biases of a neural network and is given by Equation 8.26 (see Chapter 8).

Loadings – The loadings of a data matrix \mathbf{X} containing n variables (columns) with m samples (rows) each are transformed variable vectors containing information on how the *variables* in \mathbf{X} relate to each other and are the eigenvectors of the covariance matrix of \mathbf{X} .

MSE – Mean Squared Error

Orthonormal – An orthonormal matrix \mathbf{A} is a square matrix with the following properties:

1. $|\mathbf{A}| = \pm 1$, where $|\mathbf{A}|$ is the determinant of \mathbf{A} .
2. $\sum_{i=1}^p a_{ij}^2 = \sum_{j=1}^p a_{ij}^2 = 1$ for all $i = j$. The sum of squares of any row or column is equal to unity.
3. $\sum_{i=1}^p a_{ij}a_{ik} = 0$ for all $j \neq k$. The sum of crossproducts of any two columns is equal to zero and implies that the coordinate axes, which these two columns represent, intersect at an angle of 90° .

This implies that $\mathbf{A}\mathbf{A}' = \mathbf{I}$. If \mathbf{A} is orthonormal, $\mathbf{A}^{-1} = \mathbf{A}'$ where \mathbf{A}^{-1} is the inverse of \mathbf{A} .

Orthogonal – Referring to the definition of an orthonormal matrix, a matrix is orthogonal if it satisfies Condition 3 of orthonormality but not Conditions 1 and 2.

PCA – Principal Component Analysis finds combinations of variables that describe major trends in a data set. It also summarises the data in terms of a smaller number of latent variables which are linear combinations of the original variables.

Rotation – Rotation is a method by which a set of data vectors is converted to what is called simple structure. The object of simple structure is to produce a new set of vectors, each one involving primarily a subset of the original variables with as little overlap as possible so that the original variables are divided into groups somewhat

independent of each other. This is, in essence, a method of clustering variables that might aid in the examination of the structure of a multivariate data set.

Scales – The scales or extend of the time-frequency localisation corresponds to the wavelet decomposition level and is the contribution in different regions of the time-frequency space into which a signal is decomposed by varying the scaling parameter of the scaling function. The scaling functions are smoother versions of the original signal and the degree of smoothness increases as the scale increases. As the scaling parameter changes, the wavelet covers different frequency ranges (large values of the scaling parameter correspond to small frequencies, or large scale; small values of the scaling parameter correspond to high frequencies, or very fine scale).

Scores – The scores of a data matrix X with n variables (columns) with m samples (rows) each are vectors containing information on how the *samples* in X relate to each other. They are thus individual transformed observations of X and weighted sums of the original variables.

SPE – Squared prediction error

Threshold – A threshold is certain chosen or calculated limit that has the effect of zeroing a value, variable or coefficient if it is larger than the specified threshold and leaving it unchanged if it smaller or equal to the threshold.

Wavelet – The wavelet transform is a tool that cuts up data, functions or operators into different frequency components, and then studies each component with a resolution matched to its scale. It is an extension of the Fourier transform that projects the original signal down onto wavelet basis functions, providing a mapping from the time domain to the time-scale plane. In general wavelets have the following three properties:

1. Wavelets are building blocks for general functions
2. Wavelets have space-frequency localisation
3. Wavelets have fast transform algorithms

1.1. Main Objectives

The main objectives of this study are as follows:

1. To develop a Nonlinear Multiscale Principal Component Analysis (NLMSPCA) methodology for process monitoring that is able to effectively detect abnormal situations during their early development stage and to give a preliminary diagnosis of the cause of the problem.
2. To develop NLMSPCA monitoring software as a Matlab toolbox that incorporates the whole NLMSPCA methodology for step-by-step development and easy application.
3. Application of the NLMSPCA methodology to real nonlinear multivariate chemical process data so that the performance of the NLMSPCA methodology can be tested and validated.

In order to develop and explain the NLMSPCA methodology, an overview discussion is given on the following topics:

1. Linear principal component analysis
2. Nonlinear principal component analysis
3. Wavelet analysis
4. Input-training neural networks
5. Autoassociative neural networks
6. Statistical performance monitoring charts
7. Non-parametric density estimation

During these discussions those features that make these topics so significant for process monitoring will be highlighted since it is these features that will be combined to form the final NLMSPCA methodology. A discussion on abnormal situation management (ASM) is also included in order to emphasize and justify the need and significance of this work and also to put it into perspective with the global ASM methodology.

1.2. Background

In modern process plants controlled by distributed control systems, the role of operators has changed from being primarily concerned with control to a broader supervisory responsibility: analyzing operational data, identifying unusual conditions as they develop and responding rapidly and effectively by taking corrective actions.

Any action taken on a process operation generally relies on a description of the state of the operation or events that are occurring. Timely and correct interpretation of data through improved process monitoring and fault detection will lead to improved quality, reduced cost, safer operations, and waste reduction (Kosanovich et al., 1996; Davis et al., 1995). However, there are significant obstacles to using data for process monitoring and fault detection, including the sheer volume of the data, large numbers of variables, process noise, and the non-stationary tendency of the process data due to process and monitoring sensor drift.

1.3. Data Volume

The role of the operator has become a more challenging task than before because of the overwhelming volume of data operators have to deal with (Chen et al., 1999) due to chemical processes becoming increasingly measurement rich. Large volumes of data are recorded and are often not used until the process has undergone a significant upset. Although there may be hundreds of measurements in a typical chemical process, there are relatively few events generating this information.

High dimensional data analysis is becoming increasingly common as new problems are placing greater demands on computing resources. With high dimensional data, it is difficult to understand the underlying structure: it is difficult to "see the wood for the trees." Additionally, the storage, transmission and processing of high dimensional data places great demands on systems. This data can be very useful for process monitoring if the appropriate tools are applied. Hence, it is desirable to reduce the dimensionality of the data, whilst maintaining as much of its original structure as possible.

1.4. Process Shifts, Drift and Poor Process Data

Under ideal conditions a process would be stationary, i.e. retain the same mean and covariance structure over time. However, this is rarely observed over a long period of time so that most processes will exhibit non-stationary behavior over a long enough period. The process data may exhibit large amounts of normal systematic variation on

several time scales. This normal process drift is continuous on some time scales and discontinuous on others while variations due to faults can be relatively minor in comparison.

When a process suffers an out-of-control situation, the process behavior and normal process variation can be manifested in a variety of unnatural patterns such as cyclic, trend, systematic and sudden shift patterns. The root causes of process deviations and poor process data quality, as shown below are (Ghanim & Jordan, 1996):

(1) Trend:

- Tool, device wear.
- Aging.
- Human factors.
- Operator fatigue.
- Poor maintenance.
- Gradual change in standards.
- poorly or uncalibrated instrumentation
- high noise levels

(2) Stratification:

- Incorrect calculation of control limits.
- Oscillation caused by poorly tuned controllers etc
- The misplacing of a decimal point.
- Any causes for mixtures.

(3) Mixture:

- Error in plotting.
- Incomplete operation.
- Change in method of measurement.
- Carelessness in setting.

- Change to different kind of parameters material.

(4) Cycle:

- Seasonal effects.
- Operator fatigue.
- Rotation of people on the job.
- Difference between gauges.
- Difference between day and night shifts.

(5) Systematic:

- Difference between shifts.
- Difference between assembly lines.
- Systematic manner of dividing the data.
- Presence of a systematic variable in the process.

(6) Sudden shift:

- New operator.
- New inspector.
- New machine/process.
- New machine/process settings.
- Out of design conditions (e.g. weeping or flooding)
- Temporary unstable phenomena caused by condition changes (e.g., change of crude oil or utility system)

The result is that it is normal for the process data to show considerable variation over time. This variation is often much larger than changes due to process faults. It has also been observed that the process mean shows more erratic behavior than the process covariance, i.e. how the process variables co-vary.

1.5. Current and Desired Practice

Fault detection in the petrochemical industry is routinely done with preset upper and lower limits for each variable in the petrochemical industry. However, the method sometimes does not detect faults in a short time, and furthermore, some kinds of faults are fully missed or are only found after a long delay. Operators usually take a succession of plant data as a trend (i.e. a slow-changing behavior) and unconsciously neglect fast-changing components as noise. Furthermore, they neglect the fact that significant information about faults is also contained in high-frequency components of measured data (Daiguji et al., 1997). The result is that most of these variations and especially the root causes of these variations cannot be observed or detected by current monitoring systems. Therefore, techniques are needed that are able to detect any form of process variation and systematic changes, and are also able to guide in the investigation of the root causes of these process deviations.

Without proper pre-treatment, the necessary interpretation is difficult, if not impossible. Gross data must be eliminated or modified and noise levels reduced. In many cases, critical information occurs over short duration, and hence, is difficult to detect. Rioul and Vetterli have described how wavelets can be used to pre-process data in order to better locate and identify significant events (Davis et al., 1995). Combining this type of data pre-processing with multivariate statistics holds great promise for generating useful insights into the problem of process monitoring, data analysis, and data interpretation.

A wide variety of data treatment methods and chemometrics techniques are available for application to process data, however, it is often not apparent what methods will be useful in meeting monitoring and fault detection goals (Wise, et. al., 1996). These applications can be roughly divided between those directed at maintenance of process instruments, e.g. calibration, and those concerned with maintenance of the process itself, e.g. statistical process control and fault detection. The focus of this study is on the latter. For this study principal component analysis (PCA) modeling methods, which are commonly used for multivariate statistical process control (MSPC), are used and modified to be robust over long time periods in the presence of process drift while remaining sensitive to faults.

1.6. Data Analysis and Process Monitoring

The terms data analysis and process monitoring, as used in the context of process applications, collectively refer to the interpretation and evaluation of sampled process measurements. Data analysis as used in this work is intended to describe how data are manipulated and used together with fundamental understandings to infer the state of a

physical process. Monitoring, on the other hand, refers to the classification of the data based upon a calibration model of expected behavior so that abnormal situations can be detected and fault modes isolated. Figure 1.1 is a simplified view of the on-line process monitoring activity.

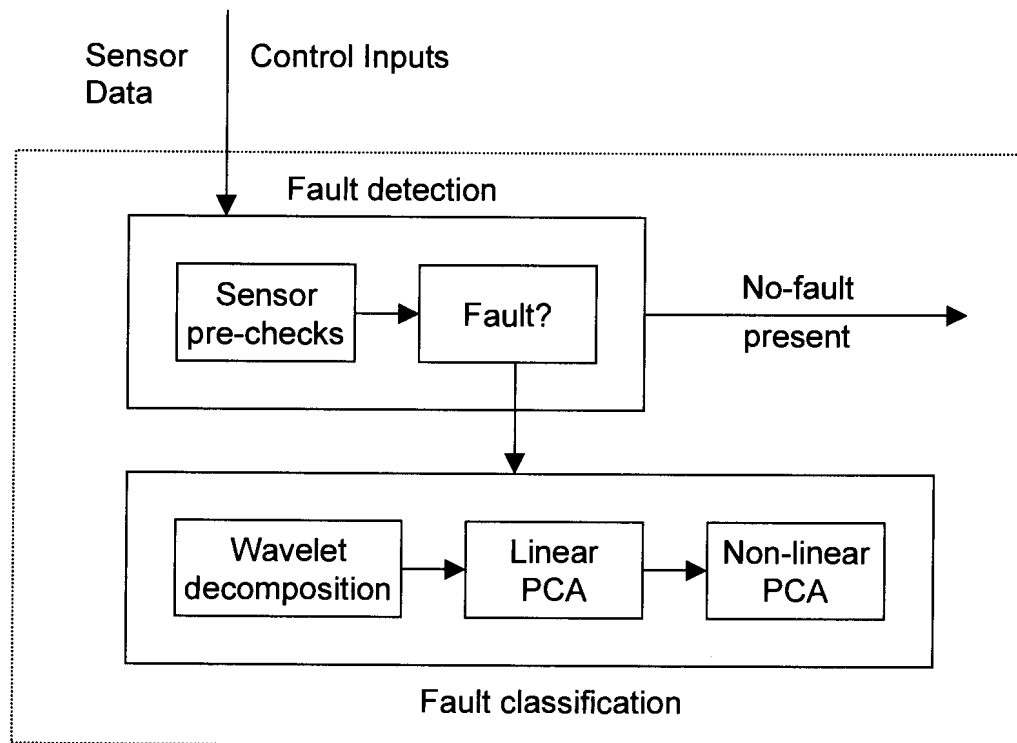


Figure 1.1 On-line process monitoring activity

1.7. Summary and Overview

Due to the aspects discussed in the previous sections of this chapter significant research has been done in recent years in more advanced techniques for multivariate process performance monitoring because of its increasing strategic importance. This research delivered promising results and followed the approach of reducing the dimensionality of the data by summarizing the data in terms of a smaller number of latent variables which are linear and nonlinear combinations of the original variables (Bakshi, 1998; Dunia and Quin, 1998; Dunia et al., 1996; Jia et al., 1998; Kosanovich and Piovoso, 1997; Nounou and Bakshi, 1998; Shao et al., 1999; Tong and Crowe, 1995; Wang et al., 1999). The most popular techniques are linear and nonlinear principal component analysis (LPCA and NLPCA). However, these analyses only concentrated on one or neither of the aspects of multiscale decomposition and NLPCA.

This study presents the non-linear multiscale principal component analysis (NLMSPCA) methodology which is an effort to combine the best of these techniques, with a few adjustments, to detect deterministic changes and extract those features that

represent abnormal operation. It combines the ability of non-linear PCA to decorrelate the variables by extracting both linear and non-linear relationships with that of wavelet analysis to extract deterministic features and approximately decorrelate autocorrelated measurements.

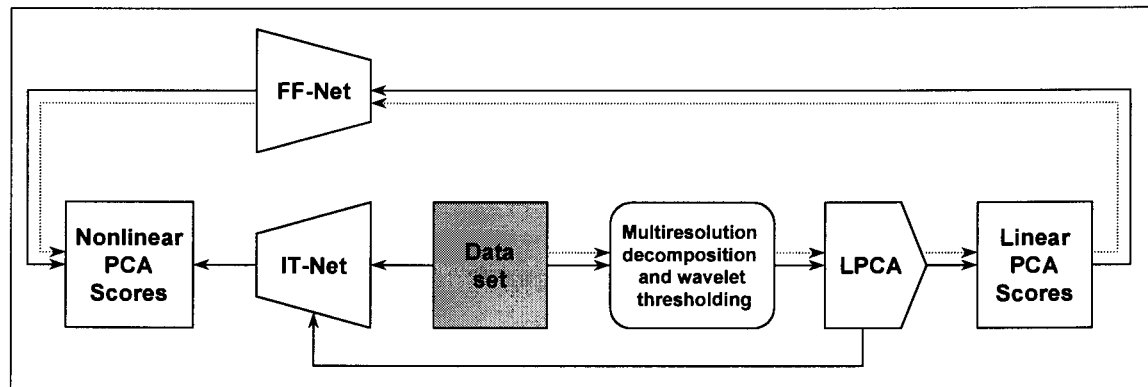


Figure 1.2. The NLMSPCA methodology (— Steps in the design phase; --- Steps in the implementation phase)

The non-linear multiscale methodology illustrated in Figure 1.2 consists of first decomposing each variable on a selected family of wavelets. Level dependent thresholding is then applied to the wavelet coefficients to select a smaller subset of wavelet coefficients. Thresholding of the coefficients at each scale identifies the region of the time-frequency space and scale where there is a significant contribution from the deterministic features of the signal. Details and approximations in the time domain are reconstructed from the thresholded and nonthresholded wavelet coefficients. The thresholded and nonthresholded details and approximations are treated separately. The nonthresholded details and approximations are combined. PCA is performed independently on the thresholded details and approximations at each scale and consists of both linear and non-linear PCA so that the process of extracting linear and non-linear correlations from the data can be performed separately. The same procedure is repeated on the combined nonthresholded details and approximations. For both linear and non-linear PCA an appropriate number of loadings are selected. Applying linear PCA results in a new set of uncorrelated ordinates. By retaining sufficient data variability, the underlying non-linear structure is not compromised and only those linear principal components associated with noise are discarded. Since the structure of the noise is not known a-priori, cross-validation as discussed in Section 7.10 is generally applied to assist in validating this.

Non-linear PCA is performed based upon the input-training neural network (IT-net) approach. Internal network parameters are trained using the Levenberg Marquardt algorithm while network inputs are updated using an extended backpropagation algorithm. This combined training approach results in faster convergence than just using backpropagation alone. After training the IT-net another network is trained that maps the observed data to the reduced data. An autoassociative network is then

constructed by combining the mapping network and the IT-net. The non-linear principal component scores are identified from the input layer of the IT-net. The advantage of this method is that both linear and non-linear correlations can be extracted from the process data to obtain a more parsimonious description of the original data. This method results in a conformance and generalized conformance principal component model.

Performance monitoring charts consisting of SPE and bivariate non-linear principal component scores plots with data-driven, non-linear control limits are derived to facilitate the comprehensive and robust occurrence of non-conforming operation. Detection limits for the scores and model residuals are computed at each scale from data representing normal operation and are calculated using the non-parametric technique of kernel density estimation.

The signal is reconstructed to the time domain and the scores and residuals for the reconstructed signal computed. The actual state of the process is confirmed by checking whether the signal reconstructed from the coefficients violates the detection limits of the PCA models. Since the reconstructed signal in the time domain is generated from the large wavelet coefficients, this approach integrates the task of monitoring with that of extracting the signal features representing abnormal operation, with minimum distortion and time delay. Consequently, there is no need for a separate step for prefiltering the measured variables. Furthermore, since the covariance matrix for all the scales together contain all the scale dependent information, the final detection limits to confirm the state of the process also adapt to the nature of the signal features. NLMSPCA transforms conventional single-scale linear PCA to a nonlinear multiscale modeling method, which is better suited for modeling data containing nonlinear contributions that change over time and frequency.

For on-line monitoring, the NLMSPCA algorithm is applied to measurements in a moving window of dyadic length.

A problem existing control chart displays are faced with is the space they occupy, limiting the display to only a few graphs at a time. A new approach is presented which allows the information to be displayed by univariate and bivariate control charts of the principal component scores and time-series plot of the squared prediction error (SPE), to be viewed in a compact manner so that the same information contained in multiple graphs can be viewed on a single display.

This advanced on-line process performance monitoring scheme is illustrated through application to a nonlinear multivariate chemical process. A complete toolbox has been created in Matlab to facilitate the design and testing of the advanced process monitoring scheme.

2.1. Introduction

Process monitoring and fault detection forms part of a much larger topic called Abnormal Situation Management (ASM). What follows may be regarded as an unnecessary long introduction to abnormal situation management. However, it is very important in the sense that it provides a background and bird's eye view over a subject for which I find it impossible to determine even estimated boundaries and puts this research topic of process monitoring in perspective to the global topic of ASM. Furthermore, it also provides some ideas for further research topics. I'm sure another ten years of intense research by a vast number of researchers can be spent on the subject of Abnormal Situation Management. The ASM Solution Anatomy model accompanying this work was developed from information collected from various sources (Anderson and Vamsikrishna, 1996; Bullemer and Nimmo, 1998; Cochran and Bullemer, 1996; Embrey, 1986; Harrold, 1998; Lorenzo, 1991; Musliner and Krebsbach, 1998; Nimmo, 1995, 1996, 1998a, 1998b; Rothenberg and Nimmo, 1996; Sticles and Melhem, 1998), including Internet searches, and represents a "generic" ASM solution.

Abnormal Situations have always challenged operations personnel, and they likely always will. Abnormal Situation Management is a particular challenge at this point in history because increased demands for higher efficiency and productivity have motivated the aggressive application of increasingly complex processes. The tremendous increases in the sophistication of process control systems through the development of advanced sensor and control technologies, and highly integrated approaches to production planning have led to productivity levels only dreamed of by previous generations of process engineers. The persistent paradox in the domain of supervisory control is that as automation technology increases in complexity and sophistication, operations professionals are faced with increasingly complex decisions in managing abnormal situations. However, the capacity of human operators to deal with this complexity, and the sophistication of their tools and user support technologies, has remained essentially unchanged and has not kept pace with the task demands imposed by abnormal situations. These sensor and control technologies have not eliminated abnormal situations and will not in the future. Consequently, operations personnel continue to intervene to correct deviant process conditions. Thus, the focus of this program is to develop collaborative decision support technologies that will significantly improve abnormal situation management practices.

Venkat Venkatasubramanian, professor at Purdue University's School of Chemical Engineering (Lafayette, Ind.), compares chemical plants with people who have a very

complex illness. "One or two doctors are unable to diagnose the illness. It takes a team of specialists each looking at the symptoms, each developing an opinion, performing additional tests, and then conferring with team members to reach a final conclusion." Similar to an ill patient, diagnosing a complex chemical process requires combinations of mathematical models, expert systems, neural networks, statistical techniques, and operations personnel, each working to independently diagnose an abnormal situation, with final diagnoses developed through cooperative problem solving.

2.2. What is an abnormal situation?

No standard definition of ASM exists. Although individual perceptions of abnormal situation management vary, there is consensus that "normal" and "abnormal" represent two distinct modes of operation. Abnormal Situations comprise a range of minor to major process disruptions or series of disruptions that cause plant operations to deviate from their normal operating state and in which operations personnel have to intervene to correct problems with which the control systems cannot cope. The nature of the abnormal situation may be of minimal or catastrophic consequence. A disturbance may simply cause a reduction in production; in more serious cases it may endanger human life.

Furthermore, abnormal operations are more likely during transition events such as startup and shutdown. Errors in situation assessment can be a source of abnormal situations, assumptions can direct plant personnel down the wrong diagnostic path and due to the response times required to correctly deal with a situation the problem may escalate.

2.3. Significance

To appreciate the significance of ASM one has to focus on the costs that accumulate with plant "hiccups," interruptions, unscheduled shutdowns, equipment failures, small losses of containment and quality problems. It is believed that solving these less dramatic disturbances potentially could yield a very high payback for companies. Estimates compiled by the ASM consortium (Harrold, 1998) indicate that elimination of all abnormal situations in petrochemical plants alone could add 5% to profits.

2.4. Goal

The goal as explained here represents a long-term goal. The Abnormal Situation Management approach is not just another attempt to introduce an "expert" artificial

intelligence device. Its success will hinge on its design as an embedded element in industrial automation system technology—integration is not enough. This long term goal is to drastically decrease the total costs of preventable process disruptions—saving industry millions of rands—by developing technologies that will offer better methods for informing operators, aiding operators during process disruptions, and preventing process disruptions in the first place.

This system should improve operator performance and offer a new challenge to operations by having the ability to interact with operations and production goals through the control system. The system should understand operations and maintenance rather than individual process variables. It should draw on other management techniques, such as incident investigation reports and the plant's corporate memory, as sources of knowledge. Useful design structures from process hazard analysis need to be captured within the system and used as rules for maintenance and operations activities.

The system should also address the communication issues identified in the site studies and provide solutions for plantwide communication, from the field to the control room. The existing industrial automation system technology from a wide selection of suppliers does not take into account casual users of the system. The same man-machine interface is provided for all users. The Abnormal Situation Management System should have the intelligence to recognize a user and provide information suitable for that person's discipline and knowledge of the industrial automation system.

Research should also address issues such as the impact of using a predictive plant state estimator on the alarm philosophy and man-machine interface. It should also incorporate an understanding of process operations and production goals and their relationship to safety, quality, environmental, and economic conflicts.

A comprehensive approach to the design of the human-machine system interaction is needed so that operations personnel receive information appropriate to their needs, while at the same time appropriate members of the operations staff are able to collaborate to solve the problem as a team. Individual needs vary as a function of a large number of variables: the current situation, the task being performed, individual preferences and styles—and others yet to be determined. In order to serve these needs, the information requirements need to be carefully assessed, not just for the current job functions present in existing plants, but for the job functions that will evolve as better decision aids become available and operators receive more support.

Systems must evolve so that the operator is not routinely swamped with information, aggravated by the user interface, required to use error-prone techniques to enter data, or exposed to situations in which being misled is even a remote possibility. The system must completely prevent adverse consequences from happening when the interaction of individuals predictably leads to misunderstandings, misperceptions, and mistakes. It

must also reduce, by orders of magnitude, the level of what post-incident review teams always label "human error."

There should be no such thing as a break down in lock-out, tag-out procedures—the user-machine system interaction model should utterly prevent such things from being possible. There should never again be coloured text on clashing coloured backgrounds on operational displays—the user interface development tools should make it very clear to the developer why such a design is inappropriate. Users should never again have difficulty in navigating from one display to another, should never again be able to enter a value for a set point that is outside the controller's capabilities, or ever again perceive the data from one unit as coming from another. And, looking to the future, decision support systems must never act like a back-seat driver when what the user needs is a helpful child—or vice versa.

When an abnormal situation is detected, operations and engineering teams must dynamically diagnose the root cause and correct the failure whilst trying to continue to meet the safety, environmental and production goals. At the same time they must track the underlying chain of events that led to the root cause(s) of the abnormal situation. As the abnormal situation evolves, some goals may need to be shed (that is, product quality, throughput, efficiency) if they compete with more critical goals (environmental or human safety).

The plant personnel should have a clear and up-to-date understanding of the types of abnormal situations recently experienced by their plants, the identified root cause and understanding of the incident investigation, and understanding of the correct steps to resolve this problem. Some plants have a variety of opinions on what was the root cause and generally lack understanding of the sources of abnormal situations and their impact on plant productivity.

Another goal is to enhance the ASM initiative to provide ways to detect and correct human errors before an undesired consequence occurs. Solution components for this problem are also beginning to emerge, but there is little consensus yet as to how to apply them. Operator intent recognition can help systems act in task-specific ways. Task modeling can help online information systems provide relevant (as opposed to canned) support. Tailored user interface displays can ensure that colour-deficient users can differentiate key data, users preferring graphs can see lots of graphs, and users needing quantitative information can see lots of appropriate numbers. And, user-centered design methodologies can ensure that this whole problem area is addressed in an empirically rigorous way when the analytically rigorous methods are lacking.

It is not a question of whether all these needs can be achieved, but rather a question of how long before they are achieved. Most of the technology is already available and just needs to be utilized and adapted.

2.5. Background: Chemical Plants and Control

A typical chemical plant will have of the order of 1000 readable “points” and a few hundred writable “points”. In addition to PID control loops, industries use distributed control systems (DCS) to simultaneously control thousands of process variables such as temperature and pressure and which can be programmed with numerous “alarms” that alert the human operator when certain constraints are violated (e.g., min/max values, rate limits). Control systems can be designed, programmed, and tuned to provide automated control for normal or near-normal operation. The major human role in this control is to supervise these highly automated systems. This supervisory activity requires: monitoring plant status; adjusting control parameters; executing pre-planned operations activities; and detecting, diagnosing, compensating and correcting for abnormal situations. The operator has a view of the values of all control points, plus any alarms that have been generated. The actions the operator is allowed to take include changing set points, manually asserting output values for control points, and turning on or off advanced control modules. Figure 2.1 gives an illustration of a typical control approach without abnormal situation management.

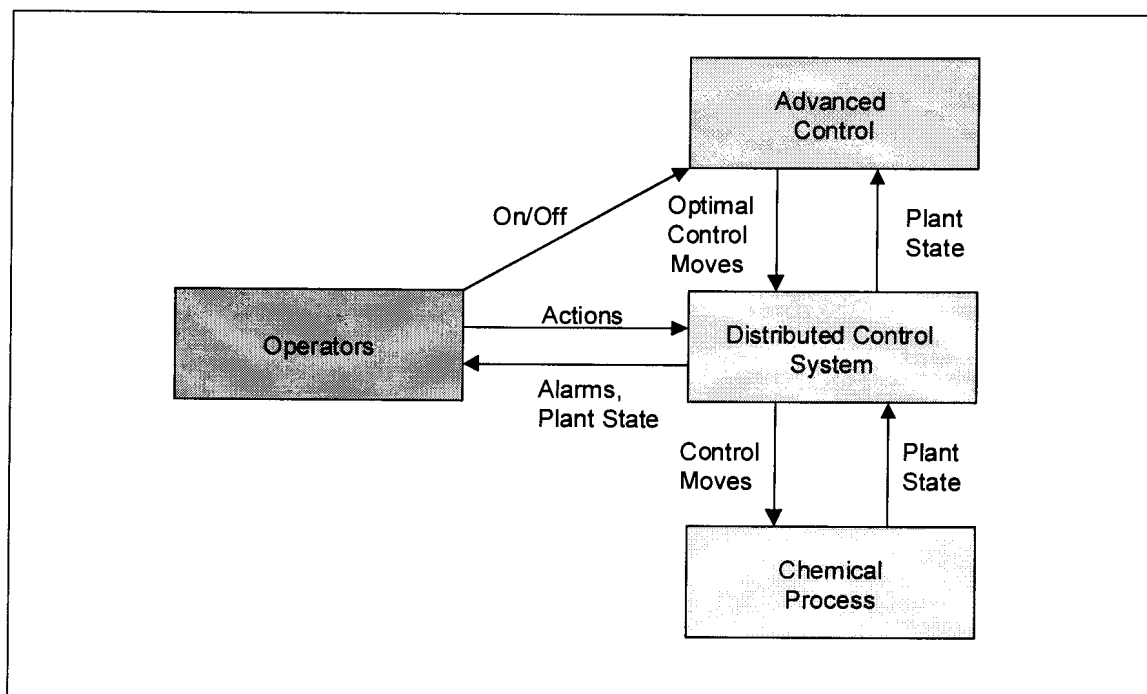


Figure 2.1. Control without Abnormal Situation Management

When the process becomes unsafe, safety instrumented systems designed to initiate a process shutdown, take over. But between normal operation and shutdown, processes can deviate into abnormal situations lasting a few minutes, or several days. Often deviations are undetected because automatic control readjusts the process. When an abnormal situation comes to the operator's attention, the common response is to place

loops in manual, reduce feed and energy streams, and manually attempt to return the process to a normal (steady) state—all the time searching for the initial cause of the problem. Frequently, the switch from automatic to manual control only worsens the situation, and a shutdown follows.

2.6. Past and Current Work

Previous approaches using technologies to assist operations in identifying and managing abnormal situations evolved large, specialized applications. These applications compared theoretical process models to real-time plant operations and generated alerts, recommendations, and predictions. Some success has been achieved with these solutions, but a lot of "care-and-feeding" is required to keep them current with ever-changing plant operations. Also, some systems use linear models that can ignore the nonlinearity and limitations of real equipment, and results in developing false predictions of equipment or process responses.

Attempts to integrate knowledge-based systems with plant operations have been few in number and mildly successful, mainly due to the complexities associated with:

- Integrating multiple proprietary platforms;
- Keeping the knowledge base current with ever-changing plant operations;
- Identifying and implementing models and methods best suited to handle the variety of complex problems of chemical process plants; and
- Getting all the operations "experts" to agree on what actions to take once the problem has been identified.

To address the problems associated with process disturbances, several industry leaders have joined forces with Honeywell to form the Abnormal Situation Management Consortium with the aid of a National Institute of Science & Technology Advanced Technology Program (NIST-ATP). Participating in the consortium are: Amoco, Chevron, Exxon, Mobil, Novacor Chemicals, Shell, Texaco and two software suppliers—Gensym and Applied Training Resources. This group is the offspring of the Alarm Management Task Force formed in the late 1980s to address problems associated with alarm functions in industrial automation systems and to suggest alarm-management enhancements. That group's work resulted in an important set of new features—defined and requested by users of the system—being included in the latest software release for Honeywell's TDC 3000X system, Release 500. The consortium estimates that by addressing the situations that are directly preventable, the losses attributable to abnormal situations can be reduced by 64 percent.

2.7. Methodology

If we are to address the problem and prevent incidents and provide tools for operators to perform more efficiently in abnormal situations we must understand the root causes of these incidents and the steps that need to be taken to eliminate or prevent escalation from an abnormal condition to a major catastrophe. The control system design needs to move from a reactive mode to a predictive mode and a long time before an alarm is initiated the system must predict the event using the latest state estimation tools.

The methodology of this research field needs to involve studying plants, reviewing previous years' history of plant incidents for different plants and sharing "best management practices". A systematic and statistical review of these incidents, together with interviews of operations personnel, can identify root causes of incidents, including problems introduced by today's industrial automation system technology and enabling technologies and the impact of system integration. Visits to sites also need to include human factor and personal performance reviews and research into how people and systems communicate. Today's offering of object-oriented software designs, relational databases, modular software development and maintenance tools, open communication standards, and acceptance of PCs makes development and deployment of knowledge-based ASM applications possible, but users still need to understand what they need and want.

This solution requires that technical challenges be overcome in three strategic areas:

- **Human-machine interaction:** A comprehensive approach to the design of the human-machine system interaction is needed so that a single user interface environment provides operations personnel with information appropriate to their needs, while at the same time supporting the collaboration of appropriate members of the operations staff in solving the problem as a team.
- **System architecture:** To provide accurate, timely support in abnormal situations, a system architecture needs to be developed composed of multiple processing modules, data bases and knowledge bases. These various software modules must communicate their conclusions with each other in real time and must remain coordinated among themselves and with human operators. Many past efforts have failed because this problem alone is so challenging.
- **System customization:** A major practical challenge in collaborative decision support technologies is configuring their capabilities to the idiosyncratic and dynamic nature of the plant processes and operations. Aspects of the software modules will need to be customized with specific knowledge about the

operations, equipment, personnel, and procedures of a specific site. Acceptable solutions will need to be self-adaptive or easily customized by plant personnel.

The system needs to be developed in a layered architecture based upon an opened standard, and so to enable it to run on any DCS which supports that standard.

Applications need to work together to determine the current state of the plant, decide upon the most appropriate goals to pursue, develop plans for pursuing those goals, and for executing those plans and monitoring the execution process. In addition, applications need to be responsible for communicating with plant personnel and for monitoring the Abnormal Situation Management System itself.

2.8. Sources of abnormal situations

Whilst major catastrophes are of concern they are fortunately infrequent and the major costs can be attributed to loss in production, quality problems, economic and conversion efficiency, equipment replacement and a collection of environmental issues.

The problems identified as contributing to abnormal situations falls into two major areas: human performance and performance of the industrial automation system and associated control equipment.

A lot of inspiration can be found in the excellent work done by Don Lorenzo for the Chemical Manufacturers Association, Inc. in his work "A Manager's Guide to Reducing Human Errors Improving Human Performance in the Chemical Industry". In this book Lorenzo states:

"Historically managers in the CPI have found human errors to be significant factors in almost every quality problem, production outage, or accidents at their facilities. One study of 190 accidents in chemical facilities found the top three causes were insufficient knowledge (34%), procedure errors (24%), and operator errors (16%). A study of accidents in petrochemical and refining units identified the following causes: equipment and design failures (41%), operator and maintenance errors (41%), inadequate or improper procedures (11%), inadequate or improper inspection (5%), and miscellaneous causes (2%). In systems where a high degree of hardware redundancy minimizes the consequences of single component failures, human errors may compromise over 90% of the system failure probability".

Safety groups estimate that human performance has been responsible for 80 percent of catastrophic incidents. The consortium's study identified several key personnel areas that hinder effective management of abnormal situations. These include: procedures not being followed, procedures that are too complex or unusable, lack of knowledge or

understanding, insufficient time to make effective decisions, and "information overload." In general, these are the results of poor context sensitivity and a lack of effective communication between the system and the people interacting with it.

Errors in sensor reading and valve positions cause a significant burden on the operations team. Operators have made poor judgment calls because the automated system reflects one value and the local traditional instrumentation registered a different value. The operator will often put trust in the device that is right most of the time especially if the other has maintenance or historical problems. Often the correlation between one process value and other variables are significantly complex, a good engineer may be able to discern that a pressure variable is incorrectly reading low given that a temperature is currently very high. Poor judgment on the part of the operator may result in erroneous diagnostics with potential catastrophic consequences.

Often, varied opinions lead to the development of multiple uncoordinated initiatives to address symptoms of a problem, whilst the root cause has not been correctly identified. The operations team believes that problems are caused by mechanical failures and the engineering teams are convinced that equipment failures are due to operational problems.

A contributing factor that does not raise the profile of this situation, and in some ways masks the problem, is the lack of measurement. This is especially true of the short upset, that may affect quality or cause slight loss of production, but which has a significant effect on net profit. Most large incidents are investigated, but the financial losses are often not recorded, making it difficult to help see the true cost in loss of product, quality restrictions, accident and injury expenses, and insurance reimbursement for damaged equipment or property. Currently only the obvious process variables are monitored like pressure, flow and temperature. However, there are other non-process variables that could provide needed diagnostic information such as noise, smell, real-time video images, infra red cameras for hot spots and many others that good field operators use every day using their human sensors.

Often escalation is caused by a series of "hidden" multiple failures in different systems. The skill level of the individual diagnosing and correcting these failures can have a significant impact on the success or disaster scenario. During a disturbance, when the highest degree of concentration is crucial, operators are currently faced with high noise levels and interference from outside sources such as phone calls, people traffic through the control room, unhelpful observers and lack of access to the control system due to the heavy traffic generated by alarms.

The largest contributor still remains the problem of time. For example, normal operation of a polyethylene process is relatively slow, but during abnormal operation a run-away reaction can cause very fast actions and there is no room for delay or error in correcting problems.

The Union Carbide's Bhopal Plant accident (Nimmo, 1996) started out as a minor problem and eventually escalated. The operator was trained and understood the actions needed to make the plant safe. As he implemented the procedures he soon discovered that backup systems were not available, cooling systems had been stripped down for use in other working parts of the plant, the flare stack was under maintenance and he was not aware of the full extent of what was in commission and was not available. When things went wrong it was not from the operators' wrong choices, but from their inability to take the correct action. That incident was based on a series of unfortunate circumstances and lack of management of change and coordination of information.

2.9. The solution

Developing a complete ASM solution requires implementing two parts, or layers as illustrated in Figure 2.2. The first layer validates incoming data and generates advisories of what is happening during an abnormal situation. The second layer predicts where the process is likely to go if current conditions persist. Some ASM solutions describe "closing-the-loop" between the ASM solution and the process. This is a form of supervisory control, with provision for the operations team to remain part of the diagnosing and prescribing process. While not all ASM solutions include all pieces of both layers, most provide the following pieces for constructing the advisory layer.

A control system interface that uses robust, real-time communication standards, such as OPC (OLE for process control), gateways to proprietary systems, or custom written application program interfaces, is necessary to obtain information from the control system about process measurements, valve positions, device status, etc.

Sensor validation to quickly detect sensor malfunctions or failures is critical to the integrity and acceptance of the ASM solution. For example, "failed" sensor input signals remain below a minimum value longer than a defined period, while "frozen" sensor input signals do not exceed the expected noise band for a period of time.

Jack Stout, president of Nexus Engineering (Kingswood, Tex.), explains, "The advanced diagnostics available in 'smart' transmitters and digital valve controllers is valuable in validating individual sensors. Many control systems can alarm, based on these diagnostic errors. ASM solutions differ by requiring sensor validation to include establishing sensor relationships to produce 'signatures' of equipment module and/or process unit performance. Informing the operations team that a pump has tripped because of cavitation, and that an empty vessel caused the cavitation, is a simple example of ASM sensor validation, alarming, and messaging."

Point retrieval of real and calculated process variable information is important in developing ASM solutions. Real process variables include temperatures, flows, pressures, analyzer results, control valve positions, etc. Calculated process variables include outputs to valves, totalized volumes, on-line material and energy balance calculations, etc. Combining real and calculated information is critical in developing performance "signatures".

Message handling and viewing must provide accurate, concise, and timely information about the current and future state of the process. ASM solution message complexity can vary from single line text messages to context sensitive help systems, allowing the operations team to view the appropriate level of detail. Some ASM solution message handlers automatically "pop" the initial alert on the operator's screen. After that, navigation buttons for cause-and-effect, details, procedures, and trouble-shooting are available.

Alarm handling that alerts the operations team of escalating circumstances during an abnormal situation requires advanced alarm management. Merely generating alarms, as many control systems do, is inadequate. As processes move through varying operational states, the operations team must remain focused on the task at hand. Spending time to work through complex alarm scenarios and then implementing advanced alarm management techniques will help the operations team to be more effective during a crisis.

Incident history archives are files of past process performance data. Initially the data may come from an existing data historian and can be used to playback past situations (good and bad) for testing the expertise of the ASM solution. Rolling data archives combine information collected by the point retrieval module and the sensor validation module into files that allow other modules to work with "smoothed" data.

Custom and generic displays are the operation team's window into the ASM workings. Custom displays are one-of-a-kind displays created specifically for a particular part of the process. Generic displays are templates for repetitive process areas (i.e., tank farms) with relevant data mapped into the display based on operator or event occurrences.

Combined, these pieces form the **advisory layer** to provide the operations team with early-warnings of a process' current health. However, the ASM solution requires additional sophistication to predict where the process is going.

The ASM solution **prediction layer** should develop equipment and plant signatures during normal operations and compare these to current operating signatures. Elements of this layer especially benefit by mixing mathematical models, neural networks, and statistical techniques to implement a solid ASM predictive layer.

For illustration purposes, the predictive layer consists of two parts: modeling, and planning and executing.

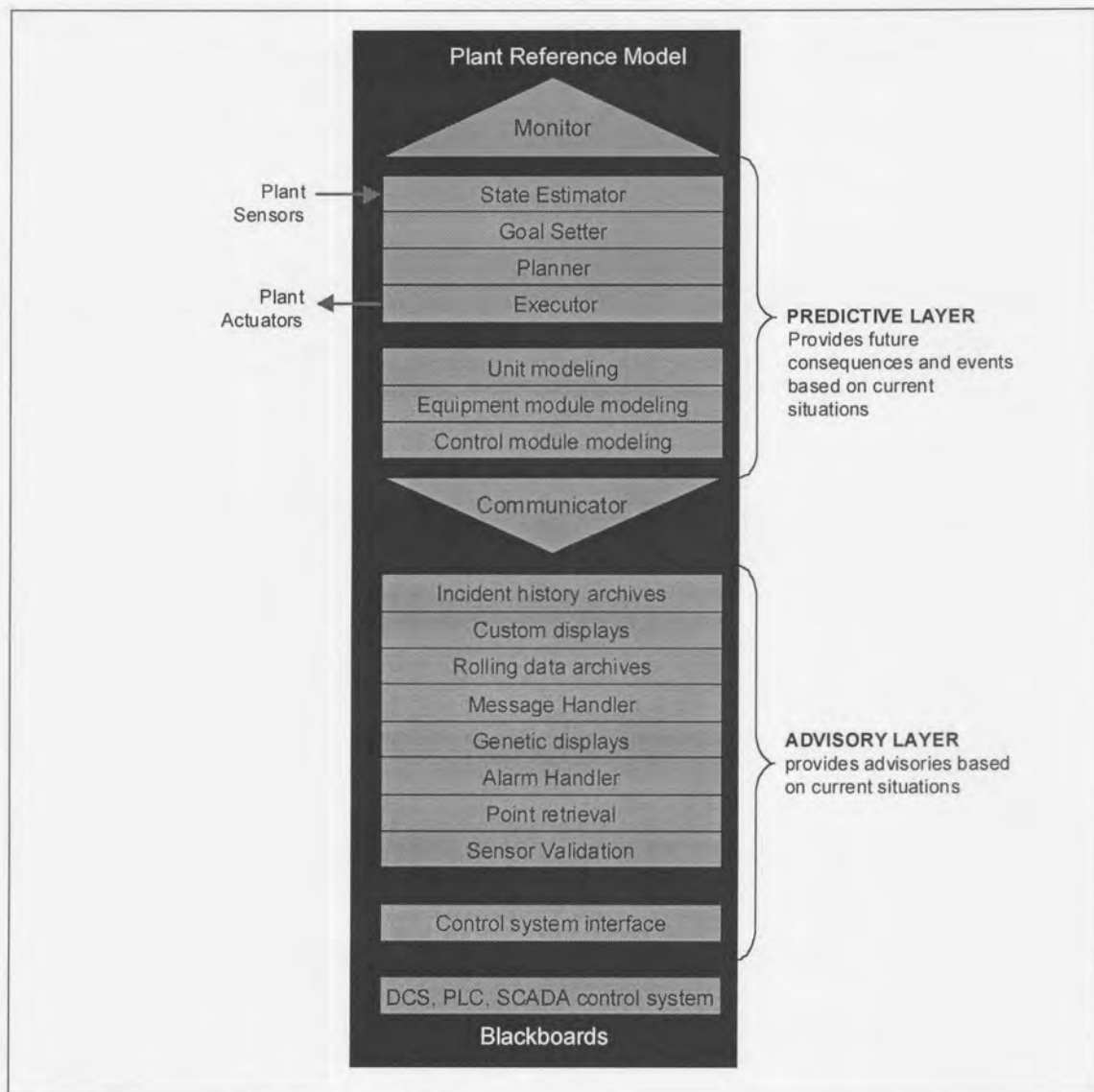


Figure 2.2. The ASM solution structure

Modular objects

ASM problems are so complex that no single mathematical modeling technique is appropriate for each piece of plant equipment. Applying the appropriate model is easier when plant equipment is viewed as individual objects. For example, the model most appropriate for centrifugal pumps may differ from the model chosen for gear pumps. Developing models in an object-oriented programming environment to match plant objects, makes assembly and maintenance of the larger, more complex process models easier.

Control module (measurements, valve outputs, etc.) modeling allows development of sensor related calculations. For example, a "rate-of-change" calculation may be a more appropriate model for a temperature measurement than working directly with the process variable.

Equipment module (pumps, on/off valves, exchangers, headers, etc.) modeling combines control module models with equipment status to form mixed expression logic formulas. For example, combining the process variable value of a flowmeter in a calculation with the on/off status of a pump to determine if a flow rate should be present, avoids a low flow "nuisance" alarm when the pump is stopped.

Unit modeling combines control and equipment module calculations to form mathematical models of equipment, such as distillation columns, fluidic catalytic-crackers, fractionators, waste-heat boilers, and compressors.

The top layer of the anatomy diagram introduces very innovative concepts, especially for many chemical operations. But, as chemical complexity (and product value) increases, as quality demands continue to toughen, as pressure to reduce emissions builds, and as demands to "stay-on-line" echo through chemical operations, innovative thinking transforms good performing companies into great performing companies.

"Closing-the-loop" of an ASM solution requires very specialized functions, such as state-estimator, goal-setter, planner, executor, communicator and monitoring modules.

State estimator modules can determine the current process state, such as improving, staying the same, or getting worse, based on information provided from the lower layers of the anatomy at varying levels of abstraction, by fusing diverse sensor data and other available information (e.g. prior control moves, known malfunctions, human observations).

Goal-setter modules gather and maintain information relevant to quality and production goals established prior to the abnormal situation occurrence. It decides which of the currently-threatened operational goals should be addressed.

Planner modules develop and recommend recover-plans to address threatened goals selected by Goal Setter after refining multiple test results from current and historic knowledge of the process represented in the modeling and advisory layers.

Executor modules close-the-loop, monitor success, execute plans, and update other Abnormal Situation Management System components in progress towards goals.

Communicator modules communicate effectively with multiple plant personnel including DCS operators and field personnel located outside the control room.

Monitor modules observe the performance of the Abnormal Situation Management System components and may adjust or adapt the system's behavior in response to observed performance.

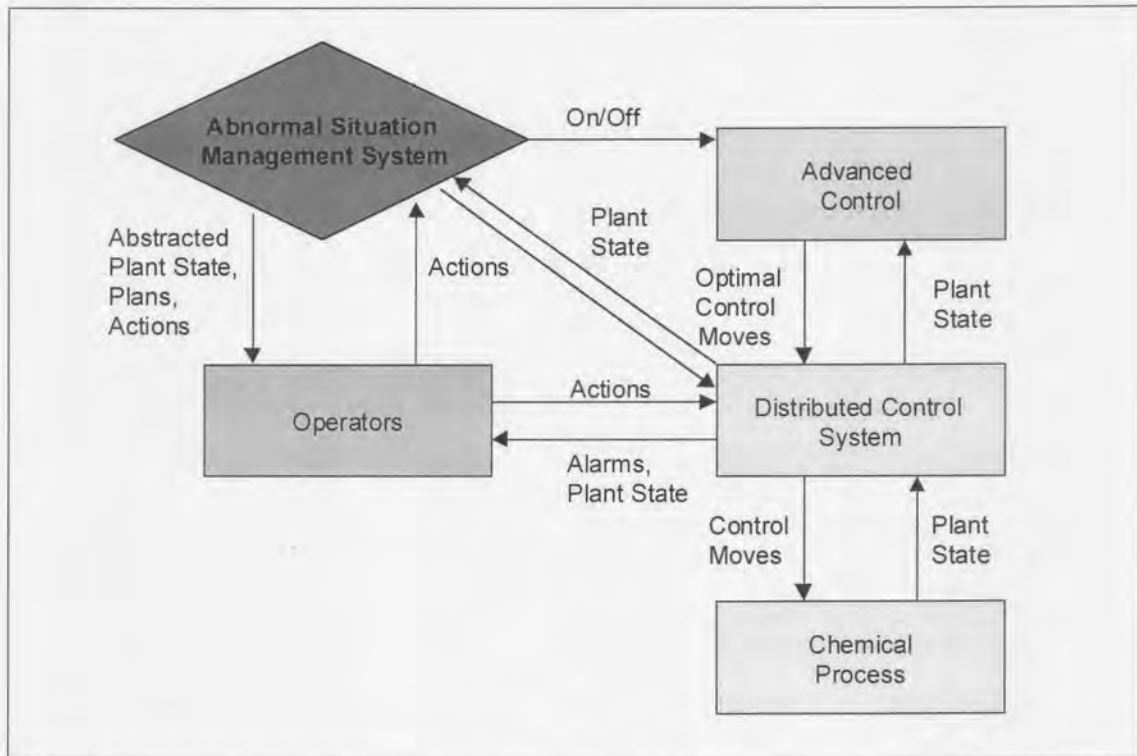


Figure 2.3. Control with Abnormal Situation Management

These functions interact by exchanging information on shared blackboard data structures. The Plant Reference Model blackboard captures descriptions of the plant at varying levels of abstraction and from various perspectives, including the plant's physical layout, the logical processing layout, the operational goals of each component and the current state and suspected malfunctions. Figure 2.3 shows how the Abnormal Situation Management System interacts with the existing system.

Abnormal situation management solutions are specialized applications of expert systems designed to work like the plant's best operator, on their best day, every day. These systems never get bored, distracted, or take a break; they remember what happened last week, last month, and last year, and provide accurate, consistent information, even in the heat of "battle".

One of the first initiatives within the solution proposal should be to provide a knowledge base of previous incidents. This system should allow the capturing of operating experience into a form of corporate memory. The learning and information from experience constitutes a fundamental source of requirements complementing those articulated by persons and organizations with an interest in the system and traditional

standards and regulations. It is well known that accidents often occur that could have been prevented by knowledge lurking in the corporate memory but forgotten and not applied. It is this knowledge that is to be fed back into the design process. The aim would be to extract this knowledge and apply it into an incident recall module. Once we have a better monitoring and investigation system we can apply the learning to design, operations and maintenance. This information could only be of use if the system understands the context of the operations:

- starting up
- shutting down
- operating normally
- what are the production targets
- what maintenance activities are being implemented or planned.

Hence, one of the strengths of solution should be its planning and goal setting ability. When contemplating ways to improve human performance Lorenzo (Lorenzo, 1991) states " there are two basic types of errors that managers must address:

- (1) Errors whose primary causal factors are individual human characteristics unrelated to the work situation and
- (2) Errors whose primary causal factors are related to the design of the work situation.

By providing resources necessary to identify and eliminate error-likely situations, managers can improve the performance shaping factors (PSFs) and dramatically reduce the frequency of human errors. This strategy Lorenzo calls the work-situation approach, and it involves the following elements:

- Implementing good human factors engineering of control systems, process equipment, and the work environment
- Providing clear, accurate procedures, instructions, and other job aids
- Providing job-relevant training and practice
- Providing ways to detect and correct human errors before an undesired consequence occurs
- Providing avenues for workers to achieve their social and physical needs

Therefore, the solution approach should form an integrated part of the current control system and have good human factors built into the control system. This will involve a new way of implementing control schematic diagrams. A style guide and implementation strategy needs to be developed for the next generation control system. More attention is recommended in control room design and the integration of different supplier's equipment. Also, we need to discover what new control screens are required and which of the more traditional need to be suppressed.

Another key issue is the configuration of the alarm system and the use of colours, symbols, priorities and how the alarm will be filtered e.g. context sensitive. In the past little attention was given to the design of the alarm system. Without proper consideration a process monitoring device is added to the control/monitoring system and on the surface it is a logical and justifiable action. During normal operating conditions this device does not cause any conflict, and may be a useful addition to the operator. However, during an abnormal situation this device is low priority and often becomes a nuisance to the operator trying to manage priority alarms.

What is required is some form of intelligence to put the alarm in context with the plant situation, eliminate unnecessary information and forward meaningful information to the operator and avoid information overload. For example, a temperature of 500 °C is appropriate under normal operations and any deviation by plus or minus 10 degrees should be annunciated but during plant shutdown the temperature may vary by new parameters, hence, the rules associated with the alarm need to change. Some processes actually cause the temperature to go outside the range of the transmitter and a new alarm is generated (BAD PV). This message is not very helpful because the operator now has to assume that the process is operating outside the range of the transmitter. This message could also mean that the transmitter is not working correctly and the temperature has not changed. To resolve this problem we need more sensor diagnostic information and better maintenance tools. The introduction of SMART Sensors has made a significant contribution to industry and has provided on-line calibration services and better diagnostics, however, more is required.

Sensors have made a significant contribution to error detection, improved reliability, and maintainability, however, what is really required is not just raw data but useful plant information. Measurement devices that are always suspect should be removed as their contribution may be only negative. When a loop shares different technology every effort must be made to ensure consistency and common calibration.

If the control system is well designed it will anticipate and prevent many situations from occurring. However, when the control system cannot maintain control, many plants are equipped with safety devices to ensure that the plant can be shutdown to a safe state. It is currently the job of the operations and engineering teams to identify the root cause of the situation and execute compensatory or corrective action in a timely and efficient

manner. A disturbance may simply cause a reduction in production; in more serious cases it may endanger human life, hence, the requirement for mechanical relief systems and automatic shutdown equipment which will always mitigate any failure of the control and ASM systems.

Training is recognized by most manufacturers as a major consideration in these situations. However there is not always a good understanding of the impact of poor training, hence the problems are not always eliminated. Manufacturers find it difficult to justify high-fidelity simulators or find the time needed for adequate training. Hence, operators lack confidence, gain experience in only normal operations which can contribute to difficulty in taking the correct actions within the time constraints imposed by an abnormal event.

DCSs are not incompatible with problem-based alarming: Indeed, mass balance analyses, expert systems, and statistical diagnostic techniques are becoming more widespread, albeit very slowly. What is needed to accelerate this trend is better ways to combine and aggregate data, better tools for easier, perhaps even automatic, development of such problem monitors, and higher-level, more comprehensive representations for plant equipment and processes.

3.1. Introduction

Since every abnormal situation is unique, it is difficult to study abnormal situation management, and in this case process monitoring, as a single subject. The best way to approach abnormal situation management is to study the theory and to then address a specific case. In this study process monitoring, as the first step to abnormal situation management, will be applied to a single nonlinear process and will lay the foundation for further investigation and development.

The advanced process monitoring methodology was applied to a real industrial process in order to evaluate its application capabilities. Due to the proprietary nature of the industrial example, only a cursory explanation of the industrial process is provided. The sensitive names of the process have been substituted with imaginary names and only normalised and standardised data are displayed. The data used, however, are real; the results of applying the methods are presented and discussed for process monitoring.

For the purpose of investigation a current problem in the steam export system at Company A was investigated since it contains all the interesting and important aspects of a typical abnormal situation.

3.2. Objective

At the time of the investigation Company A was busy building a new plant that would put a greater demand on the steam export system. The steam distribution system currently provides in the complete steam demand at Company A. However, with the new plant this demand will increase substantially. When any situation in the plant causes a decrease in the steam production, the steam export system won't be able to supply in the whole steam demand. Selective supply will then need to be applied since some processes will be more sensitive to a decrease in steam supply. A decrease in steam supply to the new plant for example will cause it to shut down.

The problem operators are faced with is the high nonlinearity that exists between steam production and steam distribution. No current accurate model exists that can relate the steam production to the steam distribution to a specific plant. This has the

effect that the influence of an upset in the steam production on a specific plant cannot be accurately anticipated in order to take preventative action with the result that an upset in the steam supply is only discovered when it is too late.

It will be to great advantage if any upset to the steam export system can be anticipated in advance in order to either take the necessary preventative actions to prevent it, or if it is not possible, to minimise the effect it would have on the whole system. In order to do this the cause of the upset needs to be identified as early as possible. The effect of the upset also needs to be quantified in order to quantify the preventative or impact minimisation actions.

So the main objective of the Abnormal Situation Management scenario under investigation would be to minimise any effect on the steam export from Process A. However, since only the process monitoring part is investigated this will not be possible yet. It should however be possible to identify the specific abnormal situation before it is noticed by the current alarm system or operator and identify the root cause of this abnormal situation in the steam export system. Quantification will also be partly possible. Only single faults will be investigated. In this study the objective is to confirm the abnormal situations identified since it was known prior to investigation from the plant history data.

3.3. Plant description

In order to understand the nonlinearity between steam production and steam distribution and why it is so difficult to generate a process model or to detect abnormal variation in the steam production or supply, one needs to look at least at an overview process description. The steam production and supply form a network throughout the whole factory. Appendix D gives an overview impression of the whole factory illustrating that the factory consists of a magnitude of separate process units linked with each other.

The most important fact to keep in mind is that during normal plant operation all the units are monitored independently. From the process description one gets a general idea of the multitude of interactions and the sheer magnitude of the process that needs to be monitored. These interactions cause many variables to be highly correlated. A general problem faced with observing such a magnitude of variables is deciding first which abnormal situation objective needs to be met (i.e. early detection of decrease in steam availability) and secondly which variables to monitor to meet this objective. These variables should be most representative of the whole process. Thirdly one needs to decide where the central monitoring system is going to be located since it will include variables from different process units. Although only a few variables are

selected in the end for monitoring purposes, there are many other factors that have an effect on these variables and on the normal operation of the plant. Therefore, the NLMSPCA system should be robust enough to detect abnormal operations despite other changes or disturbances occurring.

Since each unit is controlled separately (lack of plant wide control) it is currently almost impossible to determine when an abnormal situation is starting to occur. If two separate situations are developing in Unit 1 and Unit 2, without affecting the normal operation of these units, the effects will be carried over to unit 3 unnoticed. If, for example the two situations together have an abnormal affect on Unit 3 it will only be noticed after being carried over to Unit 3 which is some time after the initial 'symptoms' occurred in Unit 1 and Unit 2.

3.4. Scenarios

The advanced monitoring system was formally evaluated in five scenarios of which one was selected for discussion. These scenarios included sudden and unexpected malfunctions, problems originating in process equipment and the process itself.

The first set of influences investigated was that of a cutback in pure gas (PG) and reformed gas or fresh feed to Process A. Its effect on the steam export and in particularly its effect on the 43 bar steam export and export to gasification was investigated. For this purpose five sets of data were used during the month of November 1998. Firstly, data representing normal operation was gathered and used for training the system. For investigation purposes each data set represented an upset which caused a cutback in either the pure gas (PG), reformed (RG) gas or both.

Case Study 1

Date : 5 November 1998

6:30 Substation caught fire at Coal handling

Cut 700 000 m³/h on tailgas

Also affects gassification

Gassifiers 13-22 and 37 to 46 are shut down

7:15 PG Train 3 from Rectisol out of control due to insufficient gas

7:20 PG Train 5 from Rectisol out of control due to insufficient gas

Case Study 2

Date : 10 November 1998

15:22 Signal on Methane reforming compressor disappeared which caused train 5 to trip. Pure gas was cut by 100 000 m³/h.

16:17 Train 5 back in operation.

Case Study 3

Date : 17 November 1998

23:58 Oxygen 4 trip

00:05 Cut 100 000 m³/h pure gas

00:10 Trip 5 reformers

Case Study 4

Date : 20 November 1998

13:20 50 000 m³/h pure gas import from East

13:22 Fire wash feed pump tripped (train 4)

15:52 Rectisol train 4 trip

Cut to 180 000 m³/h

Case Study 5

Date : 26 November 1998

Loose compressors at cold separation (cooling compressors). This causes the feed to methane reforming to be halved (cutback on reformed gas). The other half (90 000 m³/h) that does not go back to Process A is flared.

For discussion in this report, case study 3 was selected. A second case study chosen for discussion did not involve an upset to the process itself, but involved the identification of an error in some calculation procedures after replacement of two control valves in the steam export system which influenced other parts of the system.

3.5. Background Process Information

3.5.1. PROCESS OBJECTIVE

The overall objective of the steam system is to distribute steam at High, Medium and Low pressure to consumers in the factory for use among other as an energy source. This is done by producing high-pressure steam (40 bar and 43 bar) with boilers and Process A, and letting this down to medium pressure (8 bar) and low pressure (4bar).

3.5.2. PROCESS OVERVIEW

The boilers produces superheated steam at 40 bar and Process A produces saturated steam at 43 bar. This is distributed to consumers and letdown to the 8 bar and 4 bar headers. Most important to notice is that Process A needs a fixed amount of the steam that it produces for internal use. Only the excess steam is exported. If an upset in the steam production is caused, Process A will first satisfy its own internal demand before exporting steam. This problem is addressed by example in Section 3.9 and will provide a better understanding of how the steam distribution network operates. Figure 3.1 gives a schematic that puts Process A and the steam production system in perspective to the rest of the plant. Figure 3.2 gives an illustration of the steam distribution network.

The 40 bar superheated steam is letdown to 8 bar via two letdown stations each with a desuperheater. The 40 bar also supplies the 4 bar with steam via three letdown stations each with a desuperheater. The major consumers of 40 bar superheated steam are Gasification, Oxygen plant, Power Generation and Process A.

The 43 bar saturated steam is letdown to the 8 bar header via 4 letdown stations and to the 4 bar header via 1 letdown station. Each of these letdown stations has a condensate knockout-drum. The major consumers of 43 bar saturated steam are Chemical Work-up, Phenosolvan and Gasification.

The 8 bar header receives steam from the 40 and 43 bar headers and supplies steam to the 4 bar header via three letdown stations each with two letdown valves in parallel and a desuperheater. Consumers of 8 bar include Benfield, Phenolsovan, and Rectisol.

The 4 bar header receives steam from the 40 and 43 bar headers. In the case of a high pressure on this header steam is vented to atmosphere via 4 vent valves. Consumers of 4 bar steam include Rectisol, Benfield and Chemical Work-up.

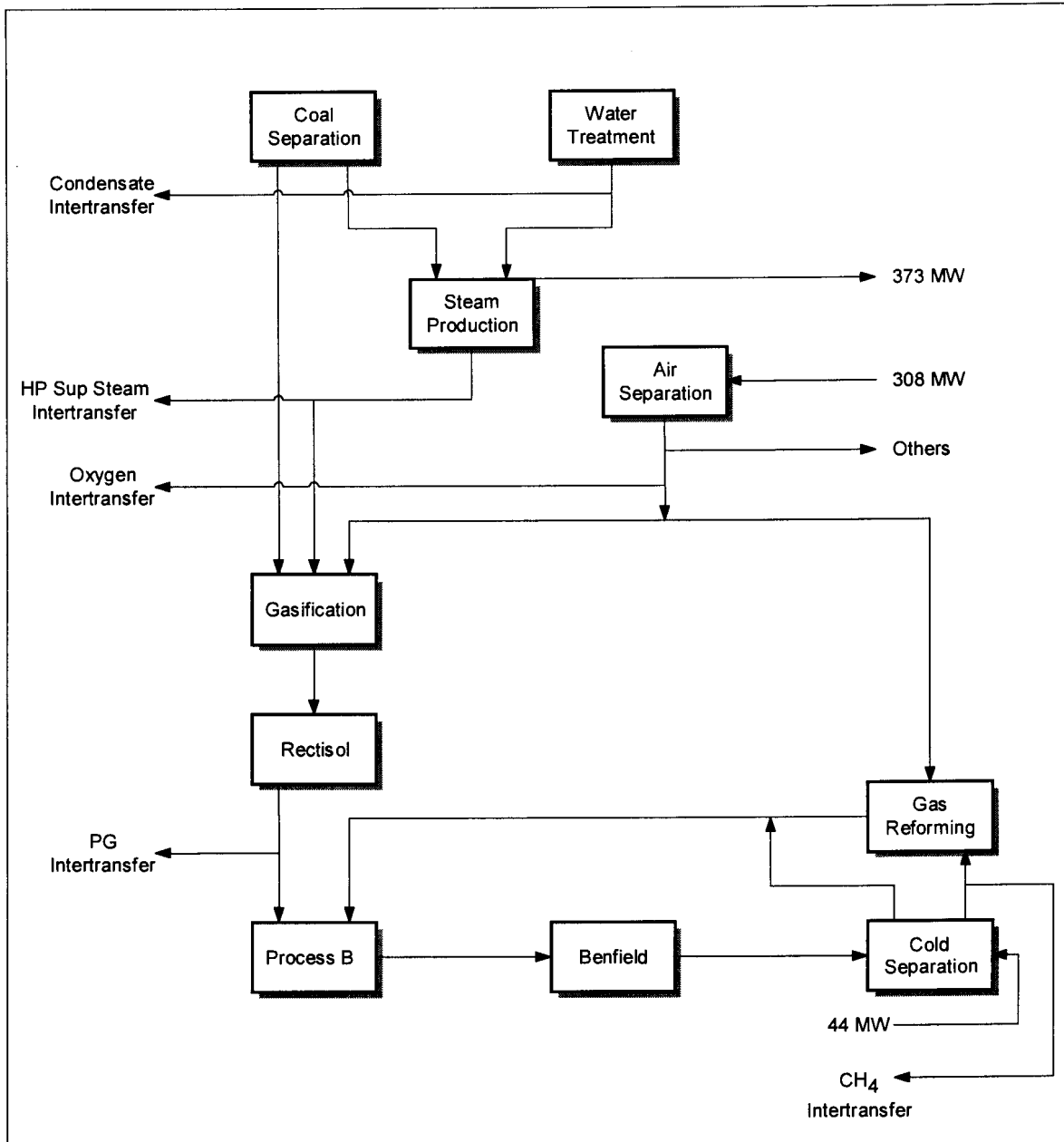


Figure 3.1. Process schematic of the most important process plants

3.6. Control Objective

The objective of controlling the steam letdown stations is to ensure a stable pressure on the headers and reliable temperature control when desuperheating.

In the case of one letdown valve going out of operation no deadband must exist and when the valve is brought back into operation bumpless transfer must be guaranteed.

Temperature needs to be controlled in such a way to ensure that as little as possible condensate will be present in the headers.

The system must also ensure that when pressure is lost on one header it must not affect the other headers drastically.

3.7 Header Functional Description

3.7.1. 43 BAR HEADER, LETDOWN TO 8 BAR AND 4 BAR HEADERS

The objective is to control the pressure on the 43 bar header by letting down to the 8 bar and 4 bar headers. This is achieved by utilizing a pressure controller to which the operator enters the desired pressure setpoint.

The steam is supplied by the Process B reactors at 43 bar and 256 °C. The steam is letdown to the 8 bar header via 4 letdown stations (2 existing and 2 new) each with a knockout drum and control valve. The letdown to the 4 bar header is accomplished with one control valve with a knockout drum upstream from the control valve (1 new station).

A direct acting pressure controller operates in split range, first opening 3 of the 4 valves letting down to 8 bar then switching back to the last valve letting down to 8 bar.

This scheme supplies all the excess steam available on the 43 bar header to the 8 bar and 4 bar headers.

3.7.2. 8 BAR HEADER, LETDOWN FROM 40 BAR AND TO 4 BAR HEADERS

The objective is to control the pressure on the 8 bar header by letting down from 40 bar and to 4. This is achieved by utilizing a pressure controller to which the operator enters the desired pressure setpoint.

The main steam supply is from the 43 bar header. The 40 bar header will supply any additional steam needed via two letdown stations in split-range. The reverse acting pressure controller operates in split-range between letting down to 4 bar (0-55%) and the letdown from 40 bar (55-100%).

A feedforward signal from the outputs of the valves letting down from 43 bar to 8 bar via a summation block is utilized to act when the pressure in the 43 bar header changes. When this situation occurs pressure will be stabilized by either shutting the 8-4 letdown valves or be made up from the 40 bar header.

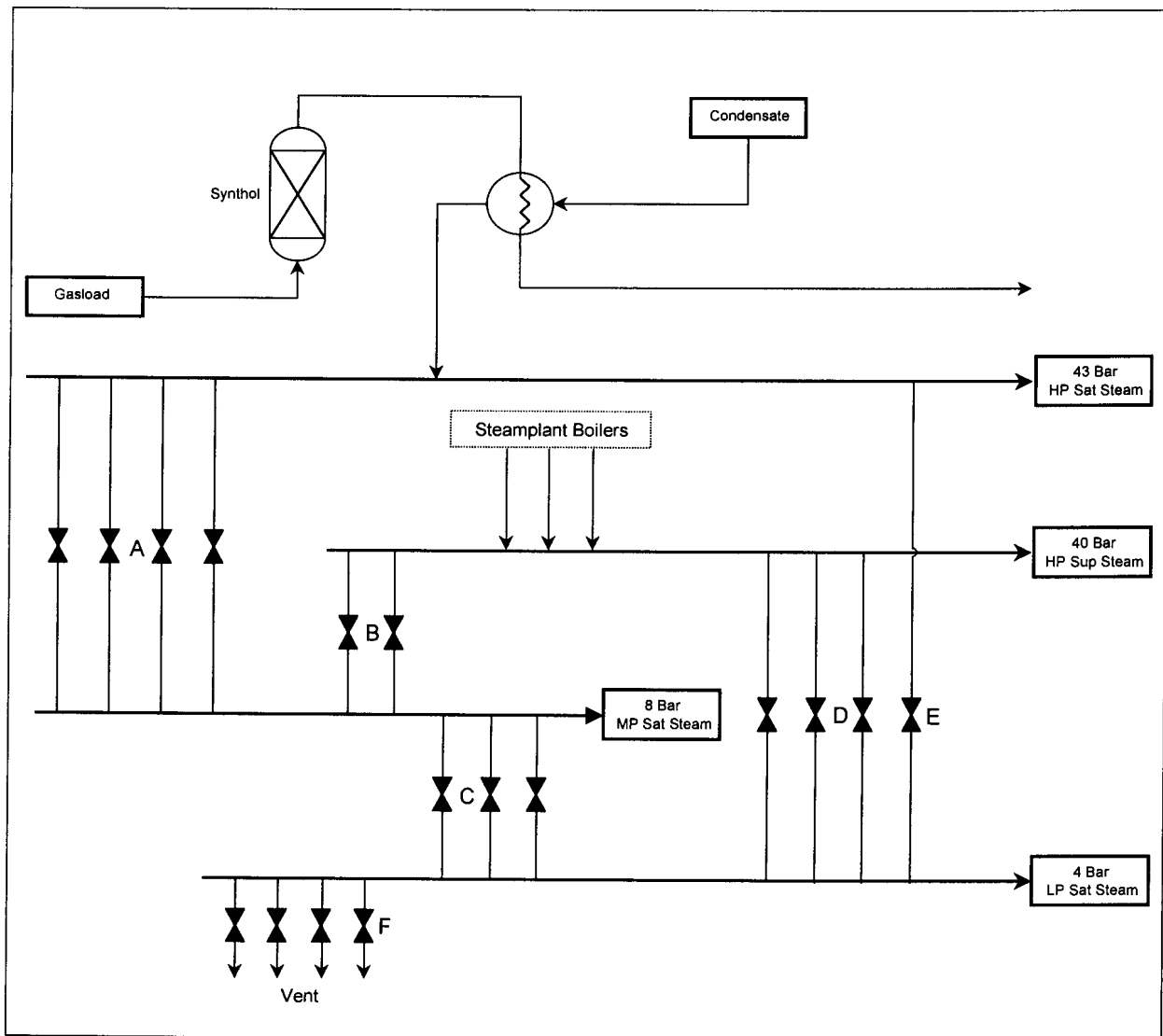


Figure 3.2 Steam distribution system

- A. 43 to 8 bar letdown
- B. 40 to 8 bar letdown
- C. 8 to 4 bar letdown
- D. 40 to 4 bar letdown
- E. 43 to 4 bar letdown
- F. 4 bar to atmosphere vent

If an under pressure situation occurs on the 40 bar an under pressure controller will override the pressure controller via an override low selector and close the valves letting down from 40 bar.

An over pressure situation on the 8 bar will cause the 6 valves letting down to 4 bar to open in split-range and relieve the situation. The valve on the northern side of the factory will be placed first in the split-range to alleviate the pressure drop problem in the northern side of the factory.

Temperature controllers on each of the letdown stations (except the new 8-4 letdown station) are used to control the amount of desuperheating.

The output of two of the valves has been characterized because the valves do not have a linear effect on the process i.e. when the one valve is 90% open and it is closed by 10% the other valves open 10% because of the bumpless transfer. When this occurs, a bump in the process is experienced because of the non-linear characteristic of the valve.

If the pressure in the 40 bar header drops, an under pressure controller will override the 8 bar pressure controller via an override low selector. When this happens a direct acting pressure controller will initialize to prevent windup. The output of one of the controllers is limited between 55 and 100%. This is done because when the controller reaches 55% both the valves letting down from 40 bar will be closed and any further reduction in output will have no effect. It is also desired that this controller does not influence the 8-4 letdown stations.

3.7.3. 4 BAR HEADER, LETDOWN FROM 40 BAR AND VENTING TO ATMOSPHERE

The objective is to control the pressure on the 4 bar header by letting down from 40 bar and venting to atmosphere by utilizing two pressure controllers to which the operator enters the desired pressure setpoint.

The 4 bar header receives feed from the 43 bar, 40 bar and 8 bar and vents to atmosphere. The steam is letdown from the 43 bar via one letdown station (new), from the 40 bar via three stations with desuperheating (existing), from 8 bar via three stations (two existing with desuperheating, 1 new without desuperheating) with two valves on each station and vents to atmosphere via four valves.

If the situation occurs where pressure on the 4 bar decreases, a reverse acting controller will increase the letdown from 40 bar in split range to increase pressure. If however an under pressure situation on the 40 bar system occurs at the same time a direct acting controller will override the pressure controller and close the letdown valves.

In the situation where the pressure on the 4 bar header increases the reverse acting controller will decrease the letdown from 40 bar until normal situation is reestablished.

3.8. Process B Reactor

3.8.1. GENERAL

The Process B reactor was designed as a replacement for the existing Train 8 CFB reactor which remains as a standby "swing" reactor. The Process B reactor makes use of an existing Train 8 quench column, product cooling train, and total feed compressor. The existing cooling train is debottlenecked by a quench column top pumparound cooler that preheats the total BFW to the Process A area. The Process B reactor has its own reactor coolant system.

3.8.2. OPERATION

The Process B reactor takes its total feed from the existing Train 8 CFB reactor inlet line. The gas enters the bottom of the Process B reactor through a gas sparger. It flows up through a distributor grid that supports the fluid catalyst bed. As the feed gas flows through the bed, hydrocarbons, water, and oxygenates are synthesized via the Fisher-Tropsch reaction. All reactor products are in the vapor phase at reactor conditions. Water and carbon dioxide are formed via the water-gas shift reaction. A mixture of water - and oil-soluble oxygenated hydrocarbons are byproducts. The reaction is exothermic. A portion of the heat of reaction heats the feed gas from the inlet temperature of the reactor to the operating temperature. The excess heat of reaction is removed by generating high pressure steam in the cooling coils.

Catalyst that is entrained from the bed with the gas stream is separated in internal cyclones and returned to the bed. The cyclones discharge effluent gas into a plenum from which the effluent gas exits the reactor. The effluent line ties into the existing Train 8 reactor effluent line upstream of the existing hot quench tower.

3.8.3. REACTOR COOLANT SYSTEM

The reactor coolant system removes the excess heat of reaction from the reactor by generating high pressure steam. BFW is fed to the steam drum through a level control valve to maintain the drum level. Saturated water is fed to the BFW circulation pumps. The BFW is pumped to the reactor cooling coils which have on/off valves on the inlets.

The reactor temperature is controlled by the operator varying the number of cooling coils in operation. Water is partially vaporized as it flows through the coils and a saturated water/steam mixture returns to the steam drum. The generated steam is disengaged from

the water in the top section of the drum. The steam exits through a demister and flows through a pressure control valve that maintains the steam generating pressure constant .

The steam separator separates any water that is carried over from the drum plus condensate formed by dropping the pressure. The steam then goes to the high pressure saturated steam header. Condensate is fed to the low pressure steam header. A continuous blowdown stream (for conductivity control) goes to the blowdown header.

3.9. Process A Reactors

3.9.1. PROCESS A REACTORS AND UNIVERSAL EQUIPMENT

The process flow can be separated into the following steps:

1. Fresh feed gas is taken in at the Process B plant. This gas consists of a mixture of the following:
 - Pure gas from the Gasification and Rectisol plants,
 - Hydrogen-rich gas from the Cold Separation plant, and
 - Reformed gas from the Gas Reforming plant
2. The fresh feed gas that is added to an internal recirculation stream, is then compressed by the total feed gas compressor, heated and fed to the reactor.

3.9.2. PROCESS A REACTOR TRAINS

1. The gas that enters the reactor picks up catalyst and carries it through a reaction chamber where the Fisher-Tropsch synthesis takes place. Two banks of cooling coils are provided to remove the heat from the reaction. High pressure steam is generated in the cooling coils and then exported via the steam drum. The catalyst is separated from the gas by means of five sets of cyclones. The reactor outlet gas is separated downstream into different products.
2. The reactor outlet gas is fed to the quench tower, where a light oil stream as well as a heavy oil circulation stream is injected into the gas stream.
3. Overhead vapours from the quench tower are now cooled and condensated in air conditioners followed by a shell-and-tube heat exchanger. This stream is then separated in the separations drum to form three main product streams.

- Uncondensed gas which is partially used as the internal recirculation stream, and partially as spare gas.
- Light oil, of which a large portion is recirculated to the quench tower and the net oil production, that is exported to the light oil stabilizing plant.
- Reaction water that is pumped to the water degassing plant.

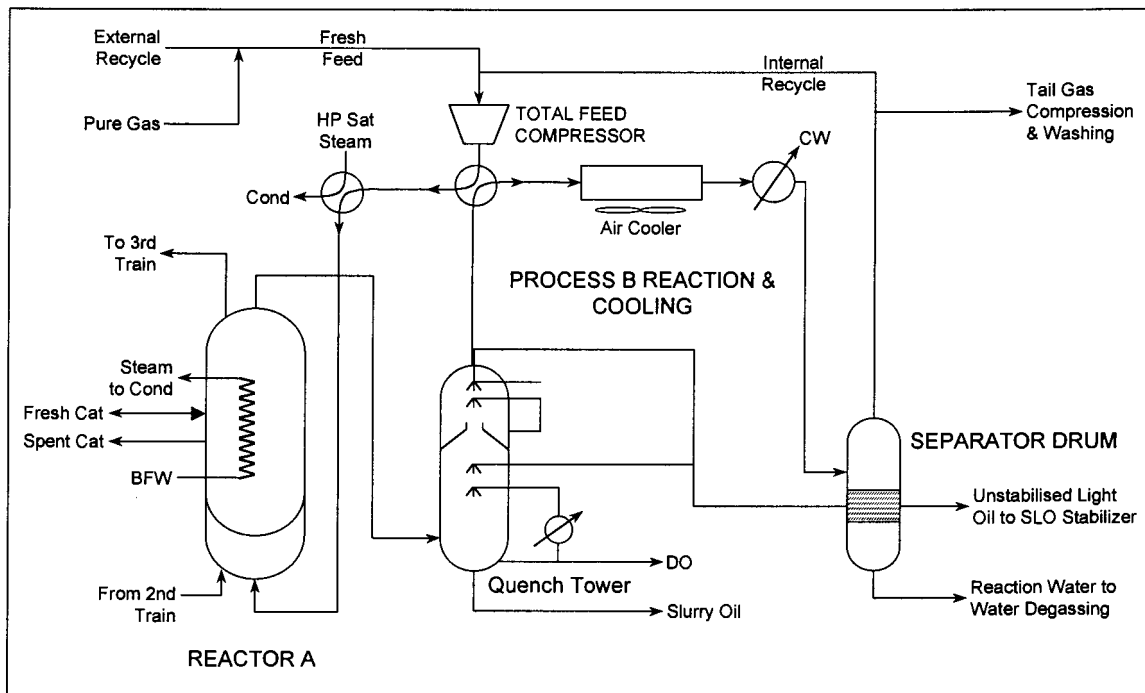


Figure 3.3. Process B reaction, cooling and recycle Loop

The flue gas is compressed by means of a centrifugal compressor and washed with water to remove the non-acid chemicals before it is transported to the plants further down.

Carbon monoxide is removed downstream from the Process A plant through the Benfield plant. The Cold Separation plant divides the gas into hydrogen-rich and methane-rich streams, as well as a C₂-rich and three condensate streams.

A large part of the hydrogen-rich gas streams is recirculated from Cold Separation to Process A. The methane-rich stream is reformed with oxygen to manufacture carbon monoxide and hydrogen. This reformed gasstream is also recirculated to Process A. The reformed gas and hydrogen-rich gas streams are called the external recirculation.

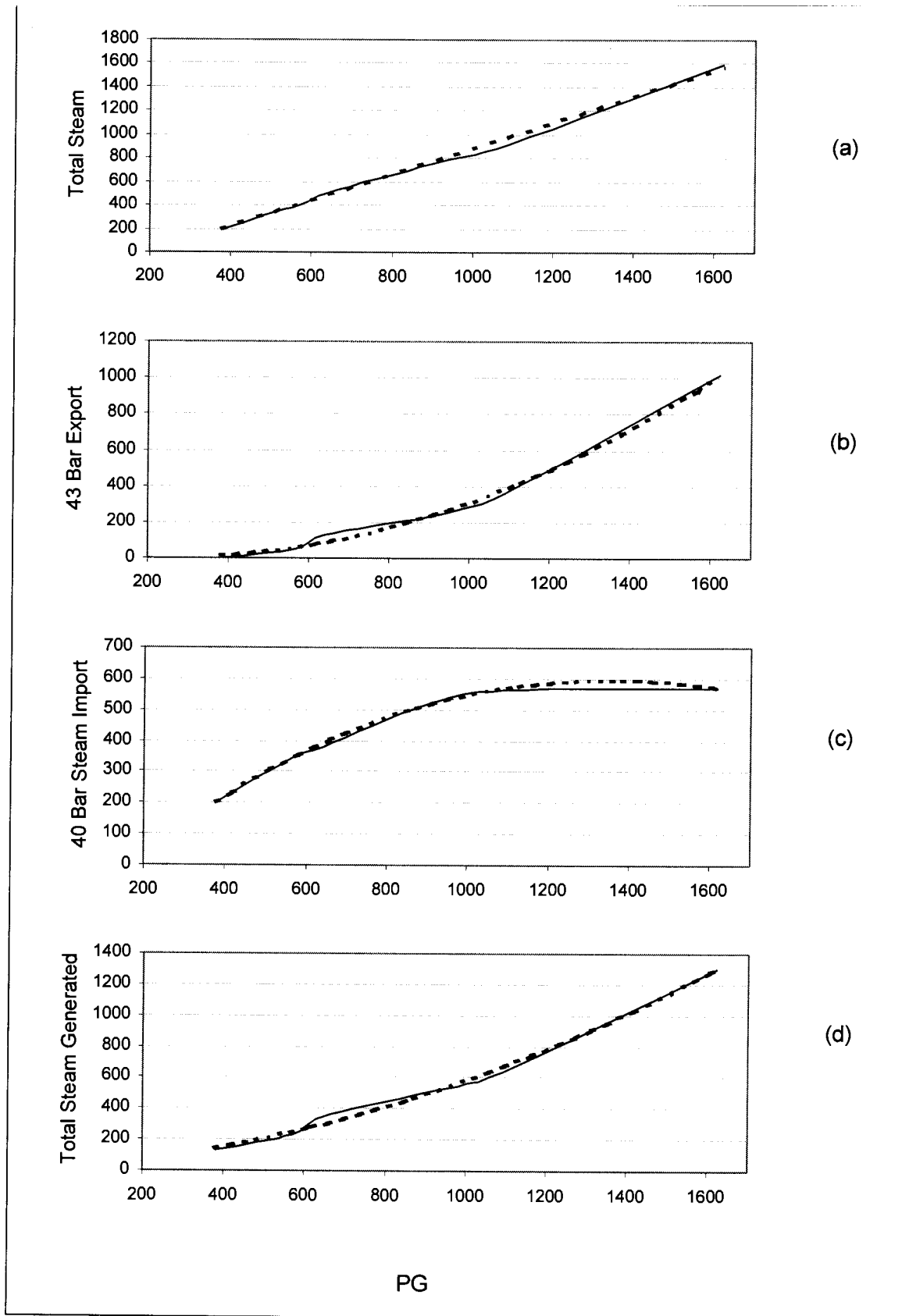


Figure 3.4 Nonlinear relationships between PG, steam production and steam export.

3.10. Steam Relationships

The figures in Figure 3.4 and Figure 3.5 will serve to illustrate the nonlinear relationships that exist between the gasloads, steam production and steam export. All the figures presented illustrate the relationship between various steam quantities and pure gas feed. The dashed lines are regression models fitted to the data for interest.

From Figure 3.4(a) we can see that there is a nonlinear relationship between the total steam being utilised in the system and the pure gas supply. This total steam is a summation of Figure 3.4(b) and Figure 3.4(c). Figure 3.4(b) clearly illustrates the nonlinear relationship between the PG-supply and the 43-bar steam export. Figure 3.4(c) illustrates the relationship between the PG-supply and the 40-bar steam import and Figure 3.4(d) gives the relationship between the PG-feed to and total steam generated by Process A. The influence on the steam export is clearly illustrated in Figure 3.5 generated from the data in Figure 3.4. From Figure 3.5(a) we can see that a 10% reduction in PG feed will cause a 24% reduction in 43-bar steam export and a 12% reduction in the total steam export.

A 50% reduction in PG feed will cause an 88% reduction in 43-bar steam export and a reduction of 40% in the total steam export.

From this it is evident that the problem lies with the 43-bar steam export. Thus, a small upset in the PG feed to Process A can have a huge effect on the 43-bar steam export, which in turn can have a major influence on the rest of the system since so many plants are dependent on the steam supply.

3.11. Process Variables

A list of all the process variables appear in Appendix C together with the calculated variables used in the investigation. From this list eight variables were selected that most accurately represent the system under investigation and is listed in Table 3.1. More detail will be provided in Chapter 5.

Table 3.1. Process variables used in the investigation (See Appendix C)

Variable number	Variable Description	Variable number	Variable Description
1	Total Rectisol Feed	5	Total Tail gas
2	Total Pure Gas feed	6	Total Steam Consumers
3	Total Reformed Gas Feed	7	Total Steam Letdown
4	Total Fresh Feed	8	Total Steam Export (Measured)

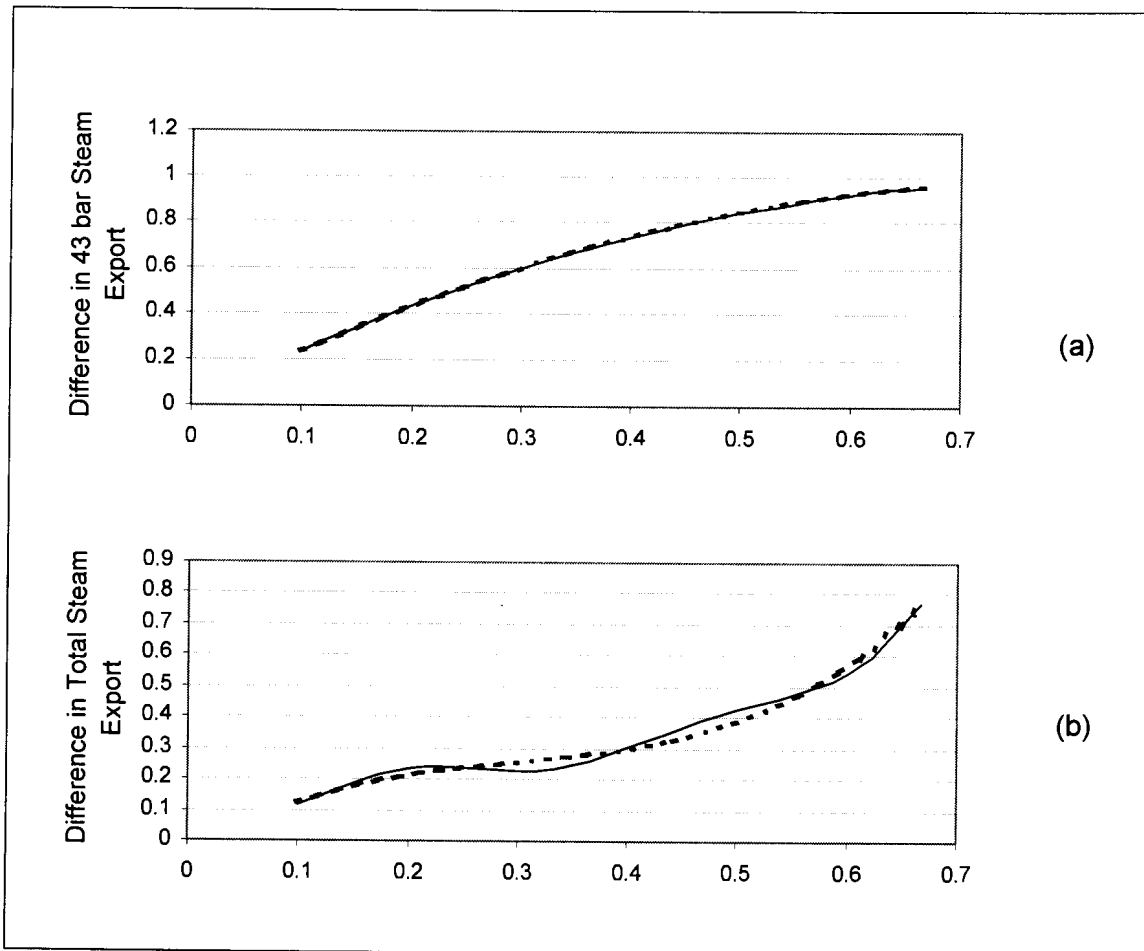


Figure 3.5. Differential relationships between PG feed and steam export

4.1. Introduction

The software used in this research was developed using Matlab. The full functionality of Matlab was implemented in order to create a toolbox that provides as much user-friendliness as is currently possible with Matlab. The creation of a separate complete NLMSPCA toolbox can be justified due to the following reasons:

- There is currently no software available for NLMSPCA since it is a new concept.
- Matlab does include a Wavelet Analysis Toolbox, but it does not contain any boundary corrected wavelet filters, does not incorporate the necessary threshold methods and cannot be used online.
- Matlab also includes a Neural Network Toolbox, but this toolbox does not allow input training or modification of algorithms.
- Using Matlab's toolbox functions makes it difficult to understand the mathematics and concepts.
- The structure of the toolboxes is such that it is very difficult to make alterations to the current software.
- It also would have been difficult to link the different toolboxes in order to form a complete functional step-by-step procedure.

The significance of the toolbox lies in the fact that:

- It operates independently from other toolboxes;
- It is understandable so that modifications or alternative ideas can easily be incorporated or linked to the current software;
- It is user-friendly;
- It automates the whole process allowing a step-by-step procedure for NLMSPCA;
- The complete toolbox can be used and accessed via user-interface;
- The flowcharting method used makes the various steps easy to follow;
- Help and background information are provided for quick reference.

The documentation provided here gives a thorough description of the toolbox and Appendix B provides extra information on the setup of the programs for someone who wishes to make alterations or use some of the applied methods in their own software development.

4.2. To Get Started

The toolbox is very easy to set up. Execute the following steps:

1. Simply copy the \Monitor directory located on the supplied cd to an appropriate directory on your hard drive for example c:\. For this example the directory c:\Monitor will then exist on you hard drive.

2. Start Matlab

3. At the Matlab command prompt set the path to the \Monitor directory for example

» cd c:\monitor

4. At the command prompt execute the following command

» startupm

5. The following message will appear

=====

This will set the search path to include all

the essential Monitor subdirectories

=====

Specify the path to the ...\monitor directory :

6. Enter the path to the \monitor directory for example

c:\monitor

and press enter. The setup may take a few seconds. While the setup is in progress the message in Figure 4.1 will appear.

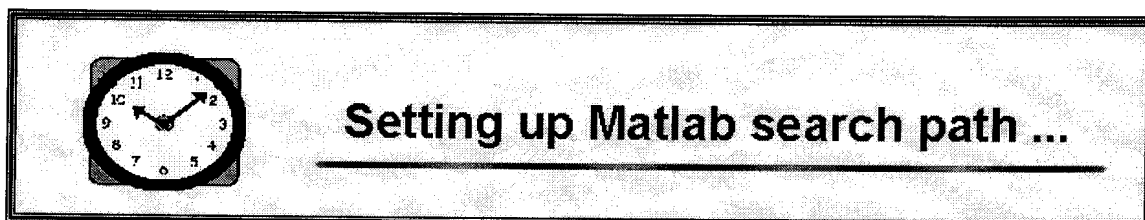


Figure 4.1 Setup progress display

7. The path setup will be acknowledged by displaying Figure 4.2.

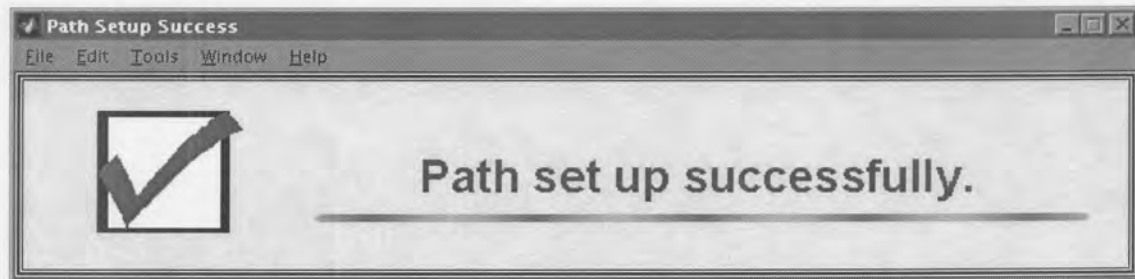


Figure 4.2. Path setup success display

8. Start the Matlab Path Browser from the Matlab workspace menu and save the path as illustrated in Figure 4.3.

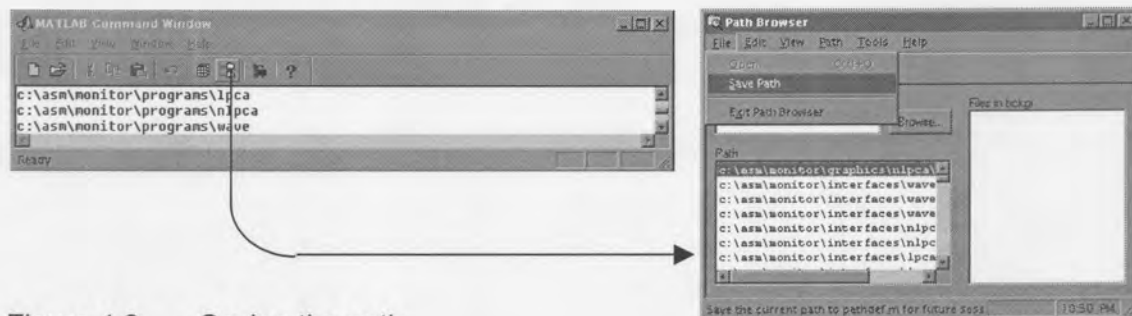


Figure 4.3. Saving the path

9. The setup is now complete. The setup will advance to the Database Setup, which is discussed in the next section.

4.3. The Program

4.3.1. DATABASE

The application makes use of a database in which all the necessary variables are stored. Each time the application is run, the variables in the database is updated. The database :

1. ensures that data is not lost while the training phase is in progress since it can take up to a few hours to generate this data,
2. saves the information generated during the training phase so that it is available for the application phase,
3. ensures that the data is available for further processing, comparisons and independent plotting.

The database is a Matlab mat-file. The system contains a default database called *data_base.mat*. This database resides in the ...\monitor\database directory. Appendix C contains a list of all the variables that are contained in the database. As discussed in the previous section, after completion of the Path Setup, the setup will advance to the Database Setup in by displaying the Database Setup Interface in Figure 4.4. It contains the name of the default database. However, a new database can be created by changing the name of the specified database.

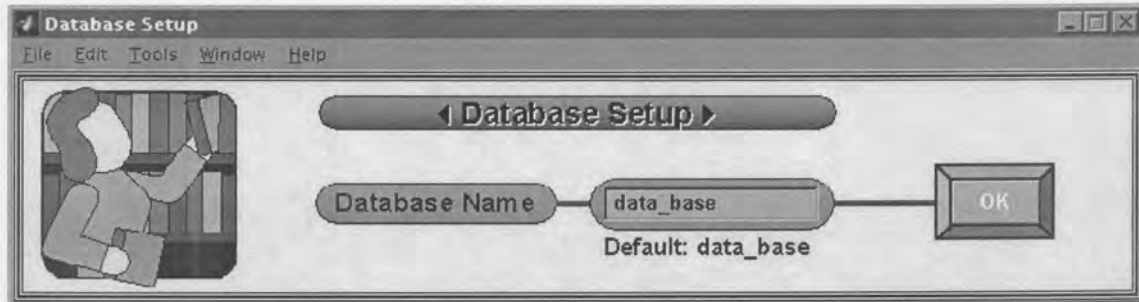


Figure 4.4. Database Setup display

The default database can be selected by clicking on the OK-button. If the default database is used the current variables that reside in the database can be used as default or can be overwritten by choosing the *Retain* option where available. If a new database is created no default options exist for the first time this database is used. The results in different databases can be compared with each other. Note however that the variables in the different databases will be the same so that if variables from two different databases need to be compared with each other the variables in the first database first need to be renamed before loading the second database. If not, the variables from the second database will overwrite the variables in the workspace loaded from the first database. After clicking the OK-button, creation of the database will be acknowledged by displaying Figure 4.5.

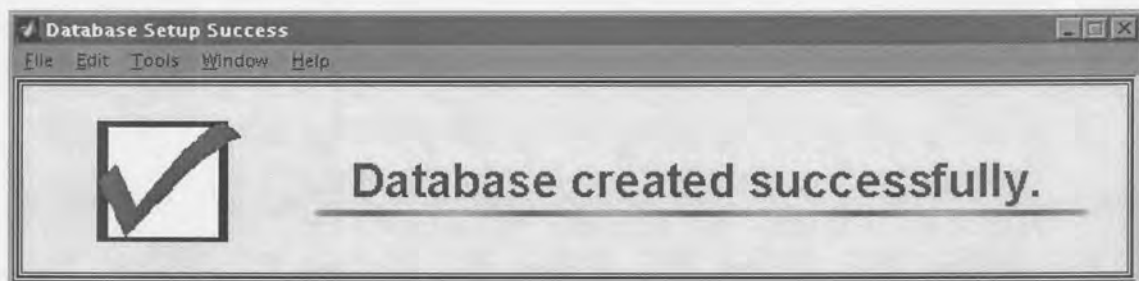


Figure 4.5. Database creation success display

4.3.2. INTRODUCTION DISPLAY

After creation of the database has been acknowledged the introductory window in Figure 4.6 will be displayed:

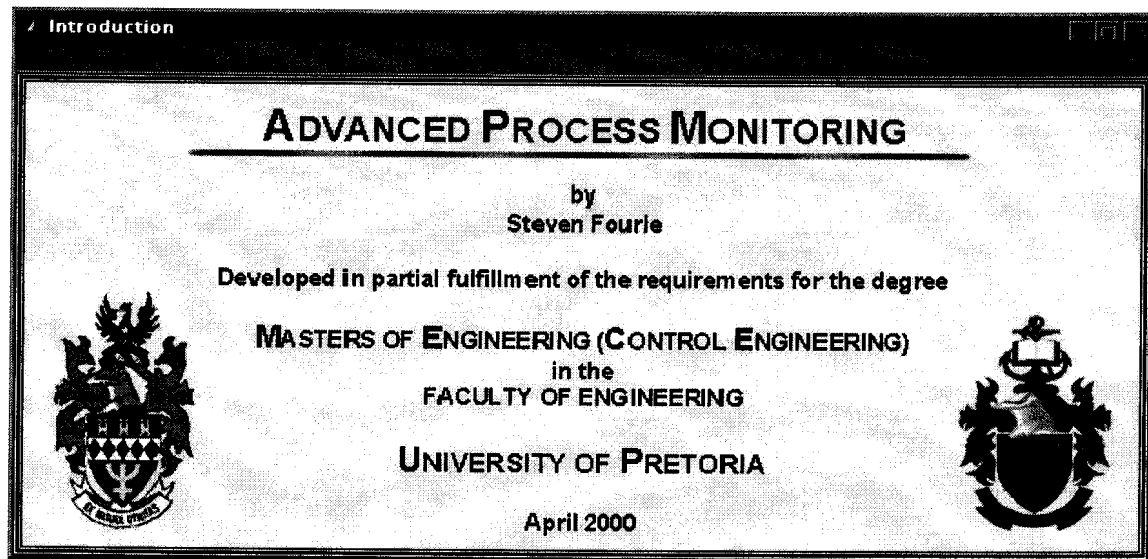


Figure 4.6. Introductory display

4.3.3. MAIN INTERFACE

Figure 4.7 Tags:

1. Go back to previous page.
2. Provides help on the current window.
3. Provides background information on the topic addressed in the current window.
4. Exit the program.
5. Exit the current window and advance to the following window.
6. Data selection and setup.

After displaying the introductory interface the setup will advance to the main interface in Figure 4.7. The main interface is a shortcut interface to all the main processing steps in the process monitoring setup which includes the following:

- a. Data selection and setup
- b. Wavelet analysis
- c. Linear principal component analysis

- d. Nonlinear principal component analysis
- e. Demapping
- f. Bivariate plot setup
- g. SPE setup
- h. Monitoring

(a) to (f) form part of the setup process which uses the normal operating process data. (h) is the actual monitoring process with new data. Selecting button 6 will take you to the first step in the process monitoring setup sequence. If you want to use current data from the database, you can jump to any other step in the setup sequence by selecting the appropriate button from the main menu. Thus, it is not necessary to start the whole process all over again if you were unable to complete the whole NLMSPCA setup process.

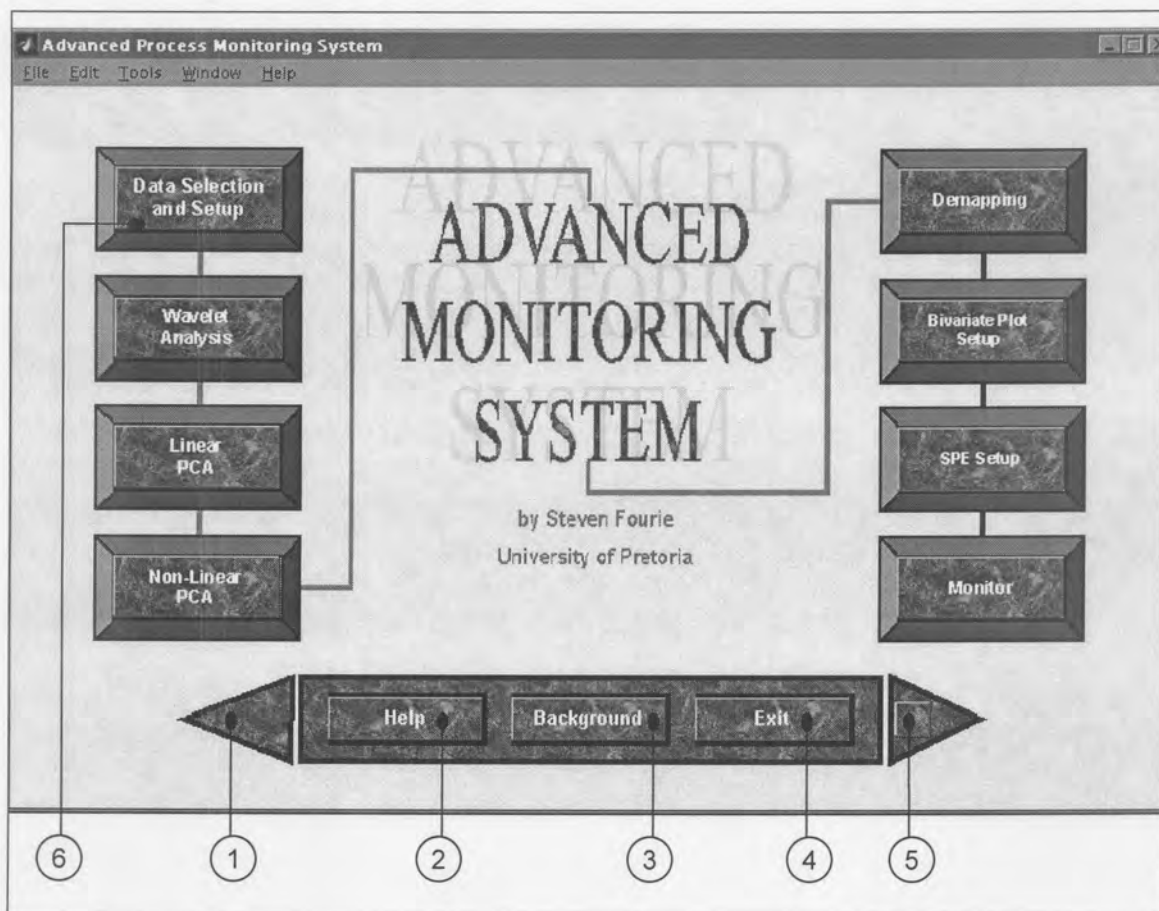


Figure 4.7. Main Interface

5.1. Introduction

The joint implementation of multiresolution analysis, wavelet filtering and non-linear PCA for process performance monitoring and fault detection is illustrated by application to a nonlinear industrial process which, in this case, is applied to the process data from case study 1 in Chapter 3 which is representative of that widely seen in the chemicals manufacturing industries.. Details of the process, except the background information provided in Chapter 3, are withheld for commercial confidentiality reasons. For the same reason the data was standardized prior to illustration. The data setup procedure is the first step in the process monitoring setup sequence and is accessed from the main menu as discussed in chapter 4.

5.2. Data Features

It has been pointed out several times in the recent literature that chemical processes are becoming more heavily instrumented and the data is recorded more frequently (Wise et. al, 1990; Kresta et. al, 1991). This is creating a data overload, and the result is that a good deal of the data is 'wasted', i.e. no useful information is obtained from it. The problem is one of both compression and extraction. Generally, there is a great deal of correlated or redundant information in process measurements. This information must be compressed in a manner that retains the essential information and is more easily displayed than each of the process variables individually. Also, often essential information lies not in any individual process variable but in how the variables change with respect to one another, i.e. how they co-vary. In this case the information must be extracted from the data. Furthermore, in the presence of large amounts of noise, it would be desirable to take advantage of some sort of signal denoising. These concepts will be discussed in more detail in subsequent chapters.

5.3. Data Setup Interface

This interface is used to load the various data sets into the workspace and save it to the database for further processing.

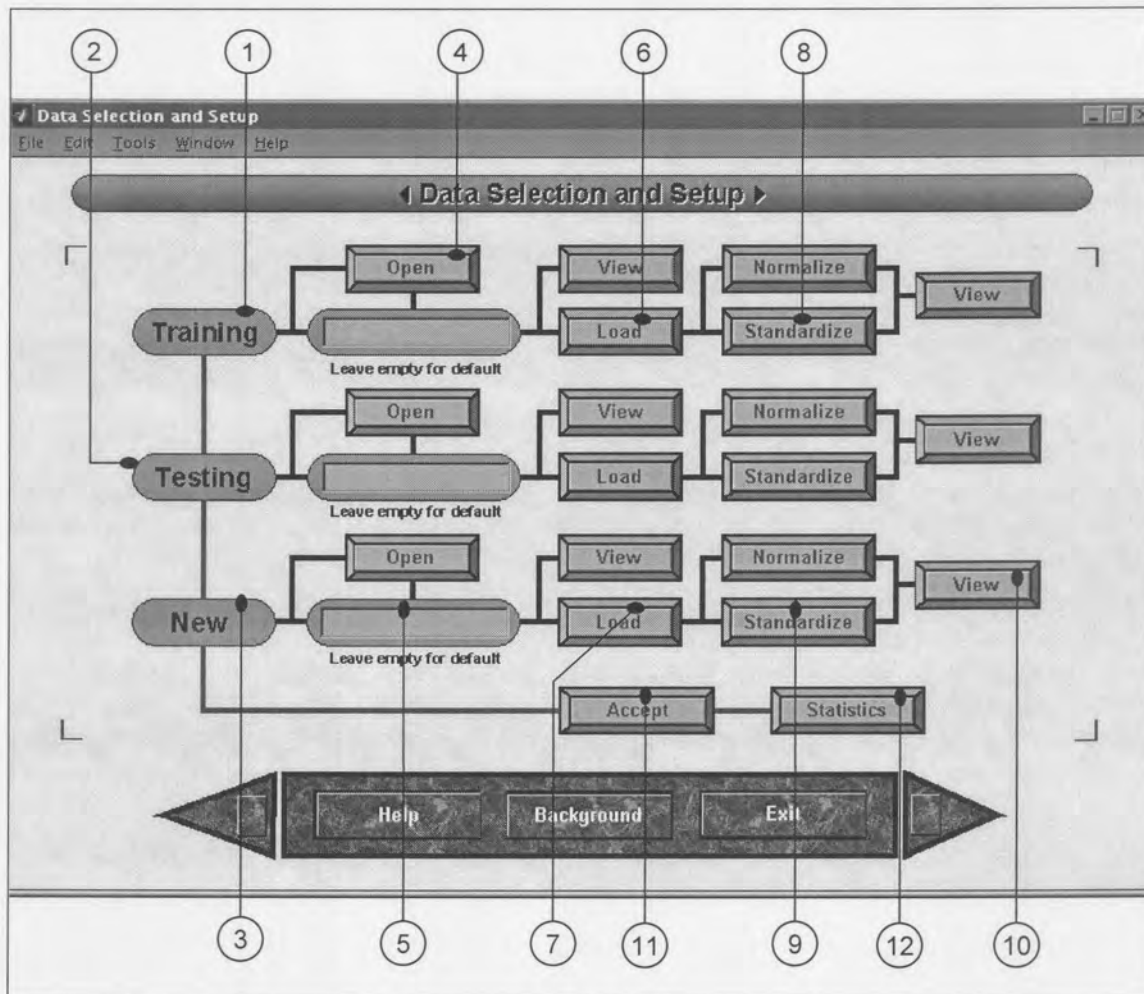


Figure 5.1. Data setup interface

Figure 5.1 Tags:

1. Training data set - this data is used for the actual training of the monitor system.
2. Testing data set - this data can be used for validation purposes when working with neural networks.
3. New data set - this data can be used to test the models in every section
4. Open existing mat data file
5. Display name of mat file or workspace variable
6. View original data
7. Load data into workspace
8. Normalise data
9. Standardise data

10. View normalised or standardised data
11. Write data to database
12. View some data statistics

The option exists to load the data from a mat-file by using button 4, a workspace variable or from the database. After the data has been chosen it can be loaded using button 7.

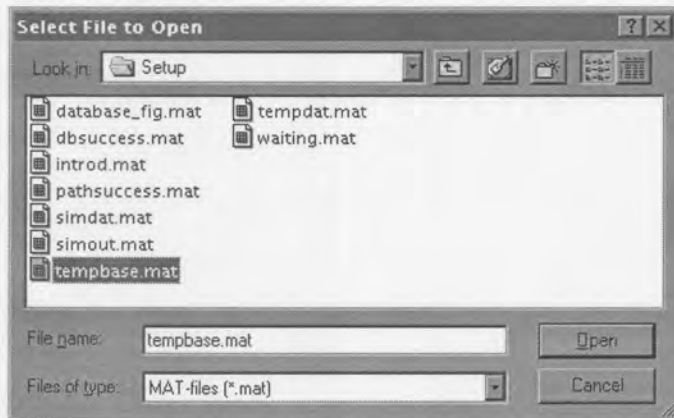


Figure 5.2. Open File interface

If no option is chosen (edit box 5 is left empty) then the data is loaded from the default database.

5.3. Data Viewer

The original data can then be viewed prior to normalization or standardization using button 6. This interface plots each individual variable separately as illustrated in Figure 5.3 and can be used to plot other variables in the workspace by changing the variable name.

Figure 5.3 Tags:

13. Variable number slider
14. Variable number display
15. Toggle between adding and removing the grid from the plot.
16. Toggle between hold and unhold. Use this if you need to plot more than one variable on the same graph.

17. Variable name. When the display window is opened it displays the default variable, in this case the variable traindata.

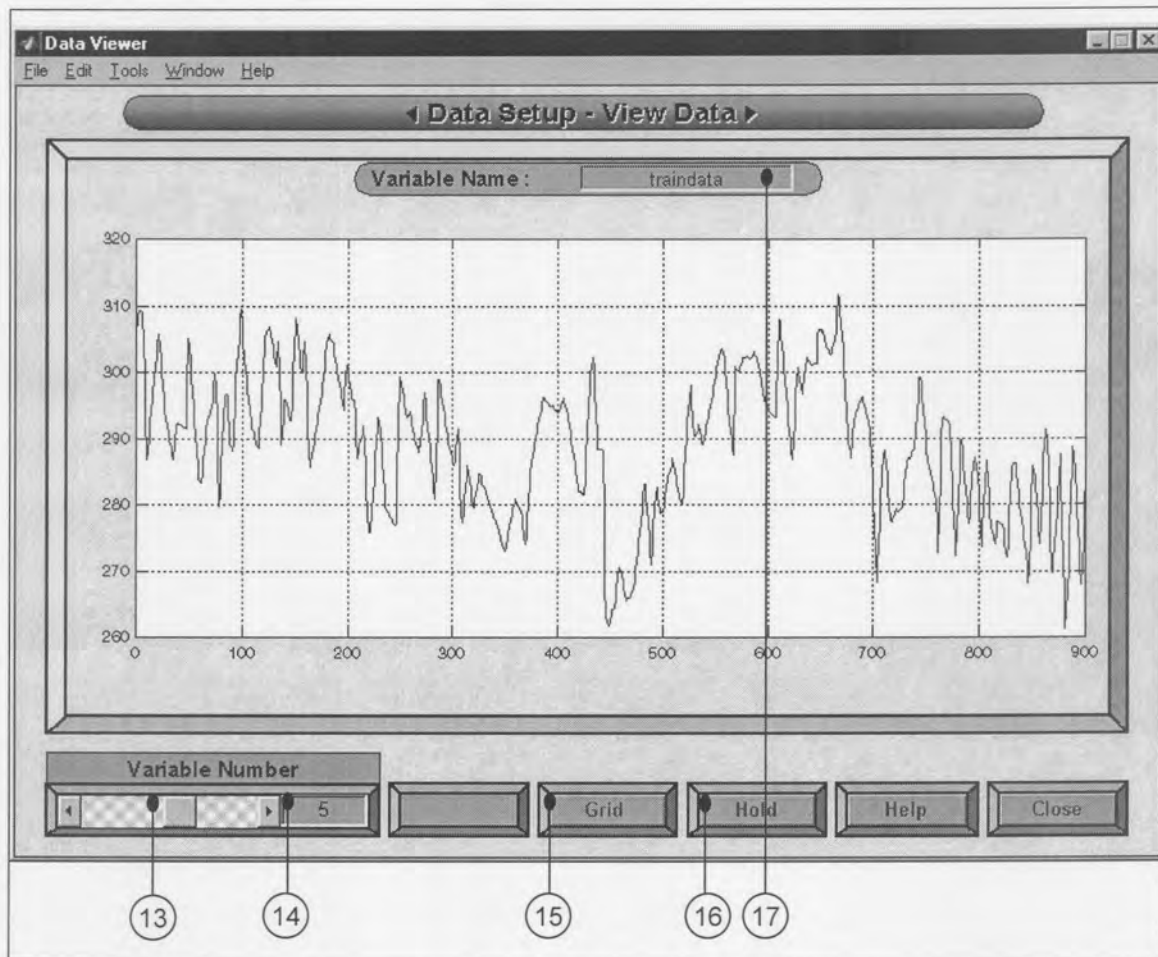


Figure 5.3. Data Viewer interface

5.5. Normalisation and Standardisation

The option exists to either normalise or standardise the data. The necessity for this becomes more apparent when the issue of principal component analysis is addressed. The normalised or standardised data can be viewed in a similar way as the original data using button 10.

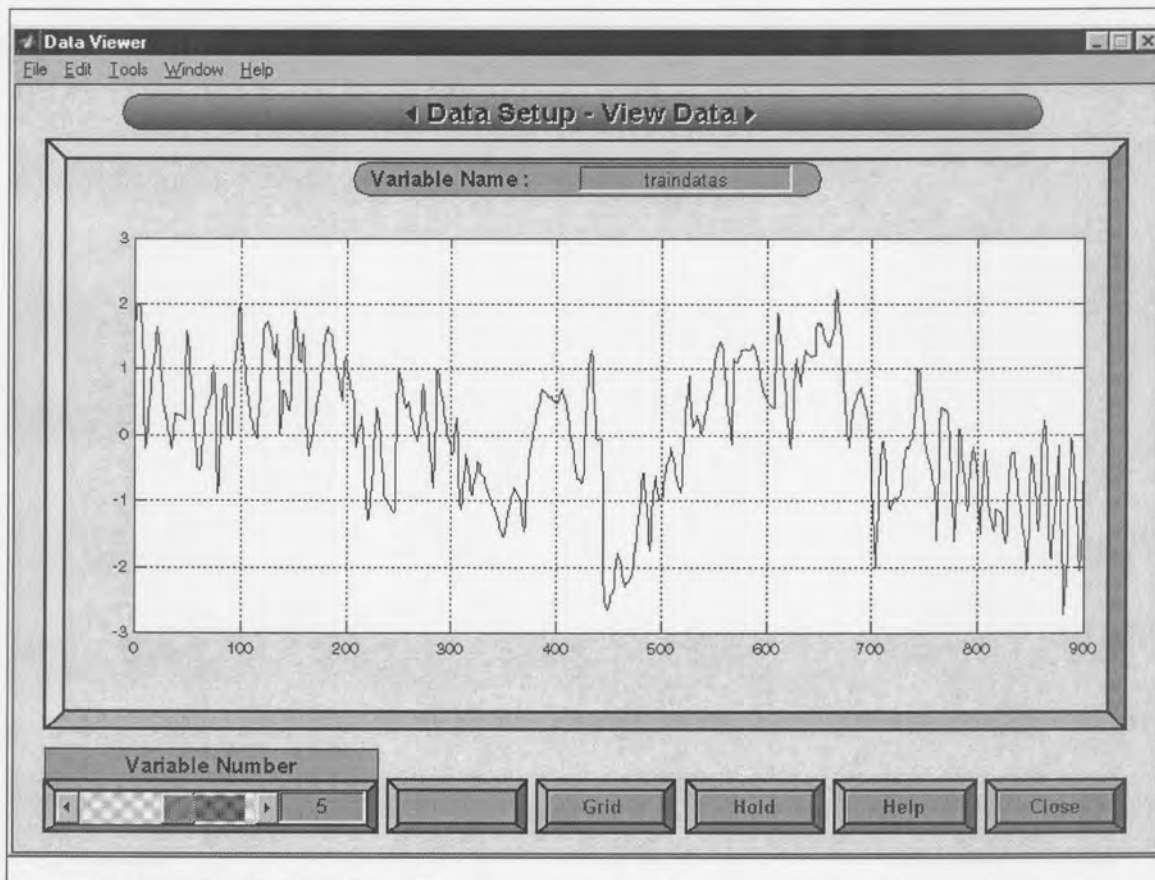


Figure 5.4. Standardized data

5.3. Statistics Viewer

The statistics viewer is accessed using button 12. Currently only the correlation coefficient can be calculated. However, other empty buttons are provided should the necessity of more statistic calculations be required.

Figure 5.5 Tags:

18. Choose from which data set the statistics are required.
19. Calculate the correlation coefficient of the specific data set.



Figure 5.5. Statistics Viewer: Correlation coefficients for training data set

5.5. Application

The first stage of the analysis was to manually carry out data pre-screening to identify and handle outliers, in-fill missing data, etc. Time-series plots of the process variables indicated that many of the measurements were corrupted by noise with some variables exhibiting sharp spikes. The sharp spikes were treated as outliers and were assumed to be due to missing data and faulty measurements. They were removed in Excel and replaced with the average of the five preceding values and five values following the outliers. Without appropriate pre-treatment of the data, the construction of a robust nominal process model for process performance monitoring is problematical and potentially worthless. Figure 5.7 is a plot of all eight variables on the same axes in order to show their relative values after removal of the outliers. Figure 5.8 is a plot of the same variables, but standardized. Both plots represent normal operation.



Figure 5.6. Statistics Viewer: Correlation coefficients for testing/validation data set

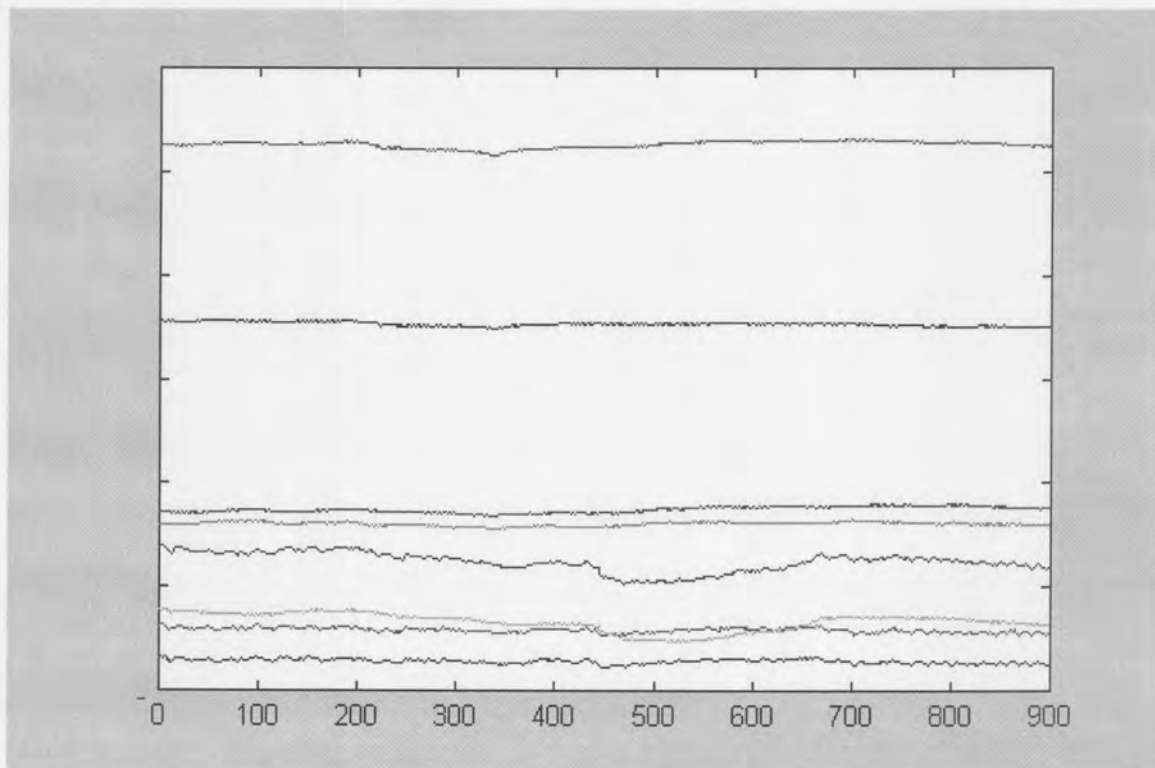


Figure 5.7. Plot of variables representing normal operation on the same axes.

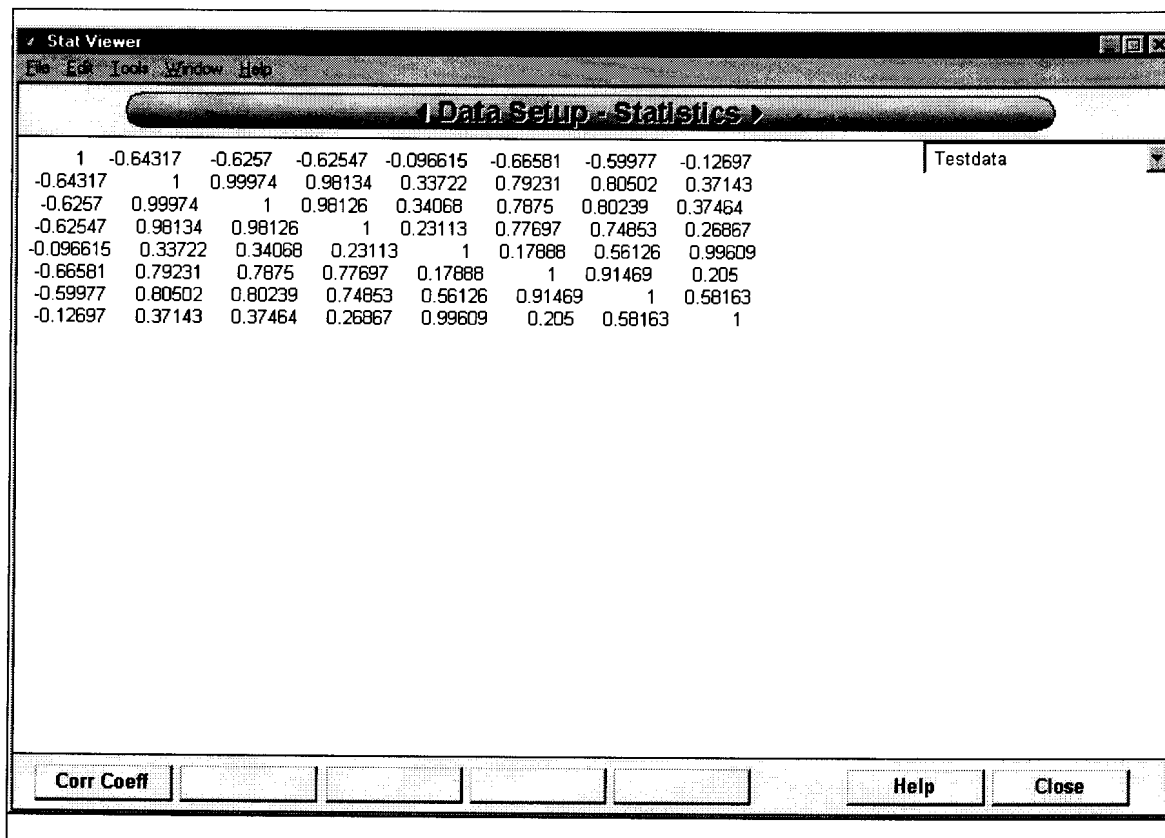


Figure 5.6. Statistics Viewer: Correlation coefficients for testing/validation data set

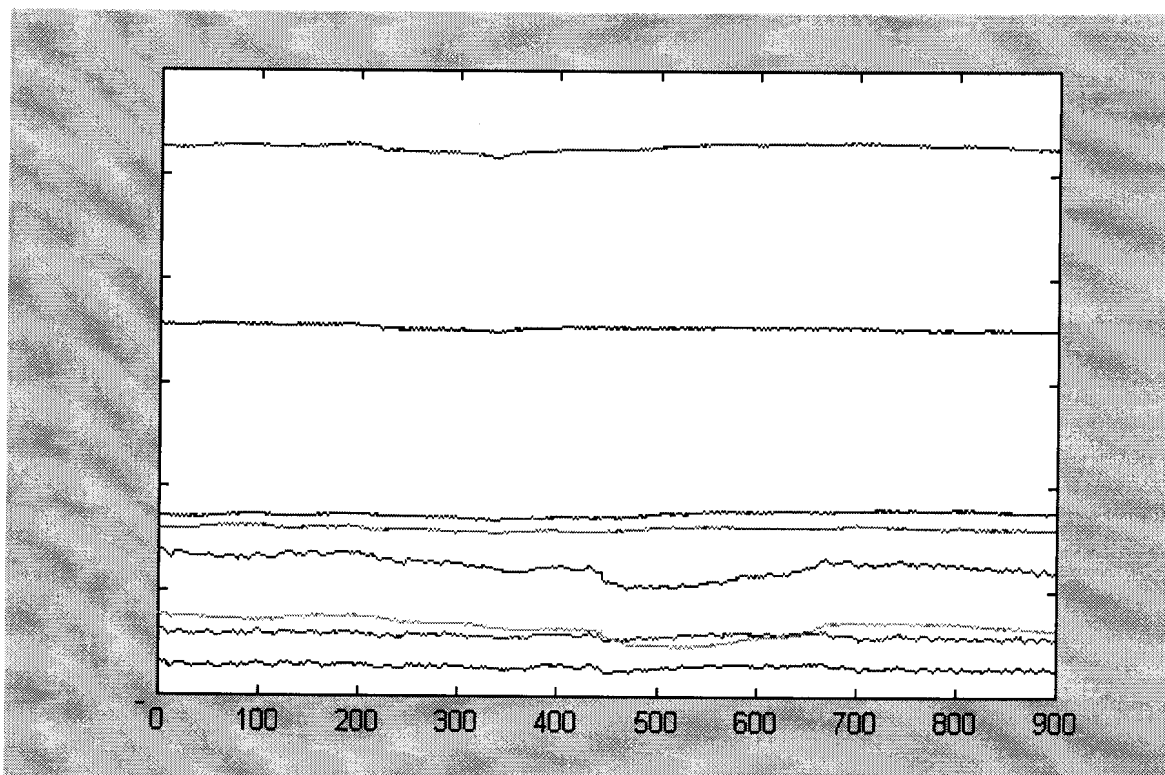


Figure 5.7. Plot of variables representing normal operation on the same axes.

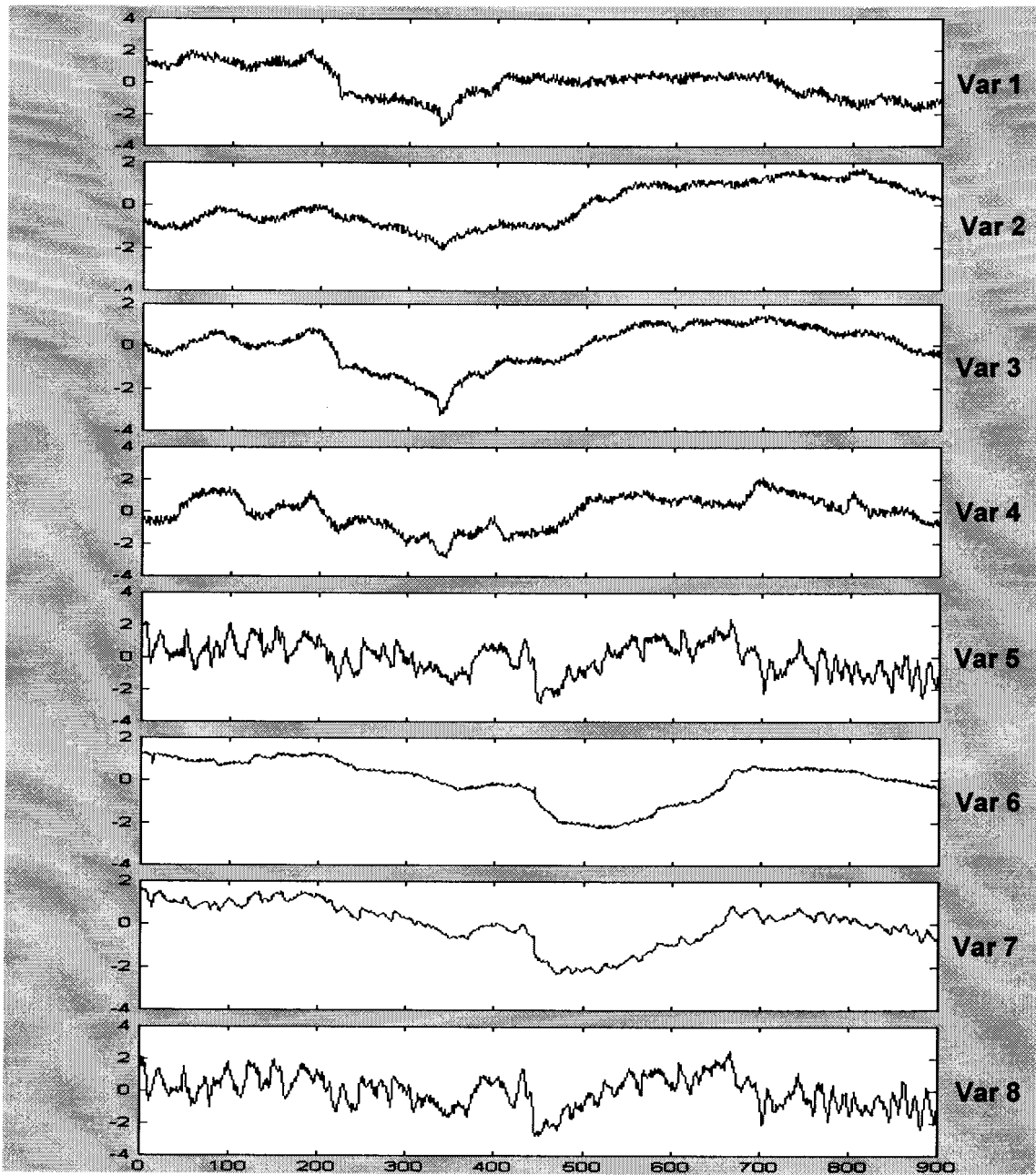


Figure 5.8. Standardized plot of variables representing normal operation

6.1. Introduction

Industrial data is synonymous with process measurement “noise”. Noise associated with the process measurements is known to have impact upon the robustness of the process model. It is therefore desirable to extract the “true” signal from the noise-corrupted data prior to carrying out any detailed statistical analysis. The most widely used forms of filtering algorithm found in the process industries include exponential and polynomial filters and the median filter. For data exhibiting small signal-to-noise ratios, heavy filtering can result in significant phase-shifts in the signal. A further limitation of some filters is that they cannot handle signal spikes efficiently or effectively. Finally, to implement some filtering algorithms it is necessary to have future values, e.g. in the median filter. In this respect they are unsuitable for on-line application. The wavelet transform addresses some of these limitations. In particular, through the application of wavelet de-noising, high-frequency noise as well as sharp spikes in the data can be removed without smoothing out the important features in the process data. The discrete wavelet transform is also an effective tool for reducing the amount of data.

6.2. Previous work on feature extraction of dynamic transients

This section briefly reviews some of the previous work on feature extraction. Feature extraction is basically a transformation of the data composing a dynamic trend to a lower dimensionality. An important property of such a transformation is that it is information preserving, that is, data is reduced by removing redundant components while preserving, in some optimal sense, information which is crucial for pattern discrimination (Chen et al., 1999).

Some researchers have adapted the episode representation technique originated by William (1986) to qualitative interpretation of transient signals. Janusz and Venkatasubramanian (1991) developed an episode approach that uses nine primitives to represent any plots of a function. Each primitive consists of the signs and the first and second derivatives of the function. Therefore, each primitive possesses the information about whether the function is positive or negative, increasing, decreasing, or not changing and the concavity. An episode is an interval described by only one primitive and the time interval the episode spans. A trend is a series of episodes that when grouped together can completely describe the dynamic feature. The approach automatically converts on-line sensor data to qualitative classification trees. Cheung

and Stephanopoulos (1990) developed the triangular-episode that uses seven triangle components to describe a dynamic trend. Bakshi and Stephanopoulos (1994, 1996) used wavelet decomposition of functions in different scales and zero-crossing of wavelet derivatives to find the inflections of decomposition. In this way, episodes can be identified automatically by computers. Based on episode analysis, dynamic trends can be interpreted as symbolic representations. The main idea of dynamic trend interpretation using episode approaches is to classify a trend such as increasing or decreasing pieces. This interpretation is sometimes not enough and inadequate in process analysis. Furthermore, there is no noise filtering in any of the episode based approaches, which significantly limits the trend representation and identification capability.

Whiteley and Davis (1992) applied back-propagation neural networks (BPNN) to convert numerical sensor data into symbolic abstractions. The major limitation of this approach is that it requires training data to train the model first.

The best known technique for signal analysis is probably the Fourier transform and it is therefore necessary to mention it here.

For a continuous function of period $2P$, the Fourier series is given by;

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left\{ a_n \cos\left(\frac{n\pi x}{P}\right) + b_n \sin\left(\frac{n\pi x}{P}\right) \right\} \quad (6.1)$$

where the Fourier coefficients are calculated by,

$$a_n = \frac{1}{P} \int_C^{C+2P} F(x) \cos\left(\frac{n\pi x}{P}\right) dx \quad (6.2)$$

$$b_n = \frac{1}{P} \int_C^{C+2P} F(x) \sin\left(\frac{n\pi x}{P}\right) dx \quad (6.3)$$

Fourier transform uses sine and cosine functions as its building blocks to decompose a function into a sum of frequency components. However, Fourier transform does not show how frequency varies with time, therefore it is not able to detect when a particular event took place. It means that the non-stationary feature of the signal is not captured. The short-time Fourier transform is able to overcome this limitation by sliding a window over the signal in time. However in time-frequency analysis of a non-stationary signal, there are two conflicting requirements. The window width must be long enough to give the desired frequency resolution but must also be short enough to lose track of time dependent events. While it is possible to optimise the design of window shapes, or

trade-off time and frequency resolution, there is a fundamental limitation on what can be achieved, for a given fixed window width (Dai, Joseph & Motard, 1994).

6.3. What is a wavelet?

Only a very brief introduction to wavelet transformation for signal processing will be presented. Only the main mathematical issues will be addressed to give some background to its calculation since it is too broad to cover here and won't facilitate a better understanding for this purpose.

According to Chen et al. (1999), wavelets can be viewed as an extension to Fourier analysis that is well-suited and designed to address the problem of non-stationary signals. Such signals are not well represented in time and frequency by the Fourier transform methods. One major advantage afforded by wavelets is the ability to perform local analysis — that is, to analyze a localized area of a larger signal. Wavelet analysis is capable of revealing aspects of data that other signal analysis techniques miss, aspects like trends, breakdown points, discontinuities in higher derivatives, and self-similarity. Further, because it affords a different view of data than those presented by traditional techniques, wavelet analysis can often compress or de-noise a signal without appreciable degradation. Wavelets offer a technique to localise events in both time and frequency and they can be applied to continuous and discrete-time problems and to two-dimensional, and in principle, to higher-dimensional data.

Another useful property of wavelets is that although they are not known to be exact eigenfunctions or principal components of any operators, they are approximate eigenfunctions of a large variety of operators (Wornell, 1990; Dickerman and Majumdar, 1994). Consequently, the wavelet coefficients of most stochastic processes are approximately decorrelated. The variance of the wavelet coefficients at different scales represents the energy of the stochastic process in the corresponding range of frequencies, and corresponds to its power spectrum. Thus, for an uncorrelated Gaussian stochastic process or white noise, the variance of the wavelet coefficients is constant at all scales, whereas for coloured noise, the variance decreases at finer scales.

A wavelet is a waveform of effectively limited duration that has an average value of zero. Compare wavelets with sine waves, which are the basis of Fourier analysis. Sinusoids do not have limited duration — they extend from minus to plus infinity. And where sinusoids are smooth and predictable, wavelets tend to be irregular and asymmetric as illustrated in Figure 6.1.

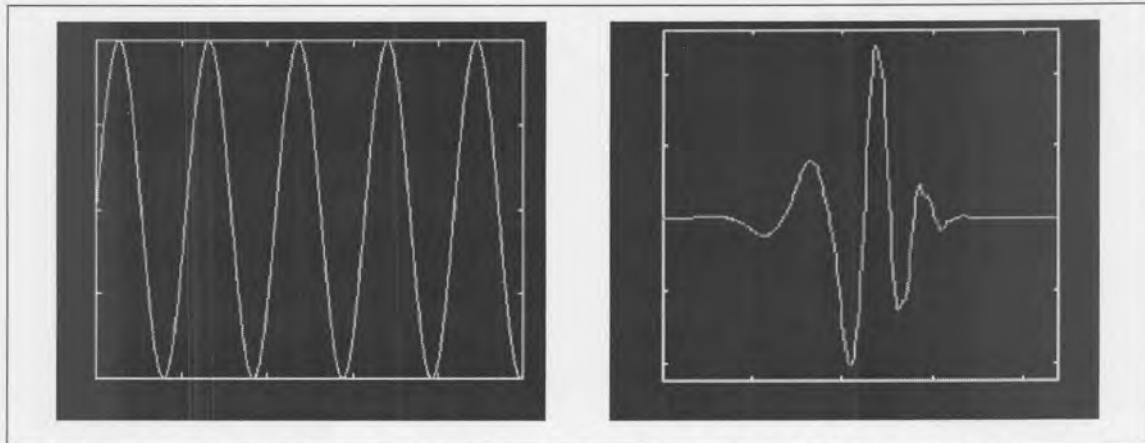


Figure 6.1. A comparison of the sine wave and the daubechies 5 wavelet

Fourier analysis consists of breaking up a signal into sine waves of various frequencies. Similarly, wavelet analysis involves the breaking up of a signal or time function into simple, fixed building blocks, termed wavelets (Rioul & Vetterli, 1991; Motard & Joseph, 1994; Chui, 1992). These building blocks are actually a family of functions which are derived from a single generating function called the mother wavelet by translation and dilation operations. Dilation, also known as scaling, compresses or stretches the mother wavelet and translation shifts it along the time axis. That is, the signal is mapped to a time-scale plane, as illustrated in Figure 6.2, that is analogous to the time-frequency plane used in the short-time Fourier transform.

The mother wavelet satisfies

$$\int_{-\infty}^{+\infty} \psi(t) dt = 0 \quad (6.4)$$

and the translation and scaling operations on $\psi(t)$ create a family of functions,

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) \quad (6.5)$$

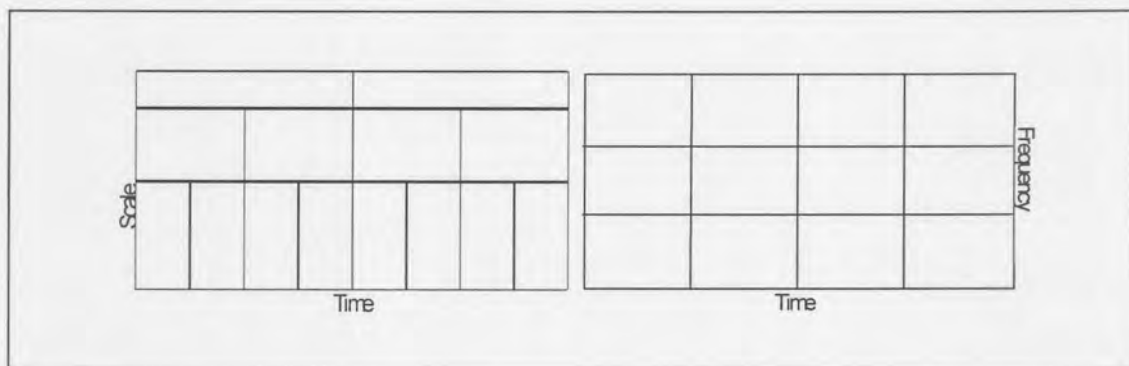


Figure 6.2. Time-scale characteristic of Wavelets (left) and STFT (right)

The parameter a is a scaling factor and stretches (or compresses) the mother wavelet. The parameter b is a translation along the time axis and simply shifts a wavelet and so delays or advances the time at which it is activated. Mathematically delaying a function $f(t)$ by t_d is represented by $f(t-t_d)$. The factor $1/\sqrt{a}$ is used to ensure the energy of the scaled and translated versions are the same as the mother wavelet.

The stretched and compressed wavelets through scaling operation are used to capture the different frequency components of the function being analysed. The translation operation, on the other hand, involves shifting of the mother wavelet along the time axis to capture the time information of the function to be analysed at a different position. In this way, a family of scaled and translated wavelets can be created using scaling and translation parameters a and b . This allows signals occurring at different times and having different frequencies to be analysed. In contrast to the short-time Fourier transform, which uses a single analysis window function, the wavelet transform can use short windows at high frequencies or long windows at low frequencies. Thus wavelet transform is capable of zooming in on short-lived high frequency phenomena and zooming-out on sustained low frequency phenomena. This is the main advantage of the wavelet over the short-time Fourier transform.

6.4. Wavelet Analysis Methodology

In the introduction the effect of noise was mentioned. Noise is a phenomenon that affects all frequencies and appears in different forms such as high-frequency measurement noise and spikes due to process filters being purged and other process operations. However, the “true” signal tends to dominate the low-frequency area, especially in chemical processes. The traditional approach to filtering is to remove the high-frequency components above a certain level since they are associated with noise. Small wavelet coefficients at low scales (high-frequency area) are usually expected to be mainly due to noise components. The procedure for wavelet de-noising is as follows:

- Apply the wavelet transform to a noisy signal and obtain the noisy wavelet coefficients,
- Threshold those elements in the wavelet coefficients that are believed to be attributed to noise,
- Apply the inverse wavelet transform to the thresholded wavelet coefficients to obtain a de-noised signal.

Each of these issues will be addressed in more detail in subsequent sections.

6.5. The Discrete Wavelet Transform

6.5.1. INTRODUCTION

Wavelet transforms can be categorized into continuous and discrete. Continuous, in the context of wavelet transform, implies that the scaling and translation parameters a and b change continuously. However, calculating wavelet coefficients for every possible scale can represent a considerable effort and result in a vast amount of data. Therefore a discrete parameter wavelet transform is often used where we choose only a subset of scales and positions at which to make our calculations. The discrete parameter wavelet transform (DWT) uses scale and position values based on powers of two (so-called dyadic scales and positions) and makes the analysis much more efficient, whilst remaining accurate. To do this, the scale and time parameters are discretised as follows,

$$a = a_0^{-m/2}, \quad b = nb_0 a_0^n \quad (6.6)$$

The family of wavelets $\{\psi_{m,n}(t)\}$ is given by

$$\psi_{m,n}(t) = a_0^{-m/2} \psi(a_0^{-m} t - nb_0) \quad (6.7)$$

resulting in a discrete wavelet transform (DWT) having the form

$$\begin{aligned} DWT_f(m,n) &= \langle f, \psi_{m,n} \rangle \\ &= a_0^{-m/2} \int_{-\infty}^{+\infty} f(t) \psi(a_0^{-m} t - nb_0) \end{aligned} \quad (6.8)$$

An efficient way to implement this scheme using filters was developed in Mallat (1989). This very practical filtering algorithm yields a fast wavelet transform — a box into which a signal passes, and out of which wavelet coefficients quickly emerge.

6.5.2. ONE-STAGE FILTERING: APPROXIMATIONS AND DETAILS

For many signals, the low frequency content is the most important part that gives a signal its identity. The high frequency content, on the other hand provides flavour or nuance. In wavelet analysis the high-scale, low frequency content is called the *approximation* and the low-scale, high frequency content is called the *detail*. The filtering process uses *lowpass* and *highpass* filters to decompose an original signal into the *approximation* and *detail* parts. The filtering process at its most basic level, which is a single-level decomposition, is illustrated in Figure 6.3 where the original signal,

$s = f(t)$, passes through two complementary high- and lowpass filters and emerges as two signals.

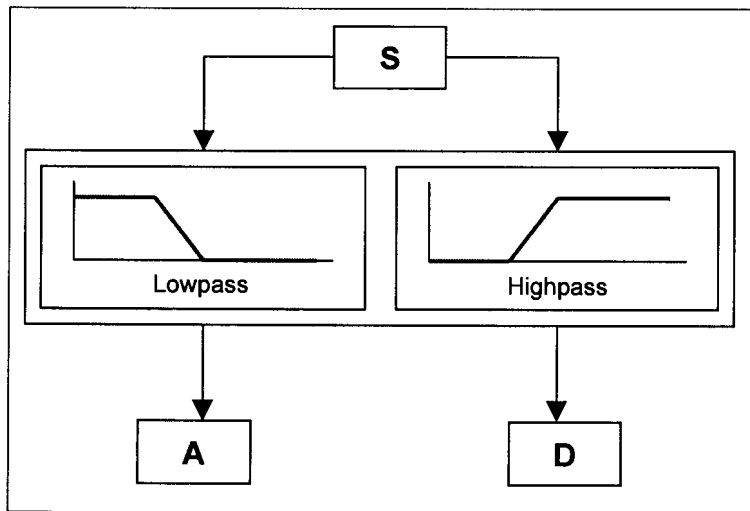


Figure 6.3. Basic discrete wavelet filtering

Unfortunately, if we actually perform this operation on a real digital signal, we end up with twice as much data as we started with. Suppose, for instance, that the original signal s consists of 1000 samples of data. Then the approximation and the detail will each have 1000 samples, for a total of 2000. However, it is not necessary to preserve all the outputs from the filters and therefore, to correct this problem, we introduce the notion of downsampling where we keep only the even components of the *lowpass* and *highpass* filter outputs and throw away every second data point. While doing this introduces aliasing, which is a type of error (Strang and Nguyen, 1995), in the signal components, it turns out we can account for this later on in the process. This procedure is illustrated in Figure 6.4(a) and (b).

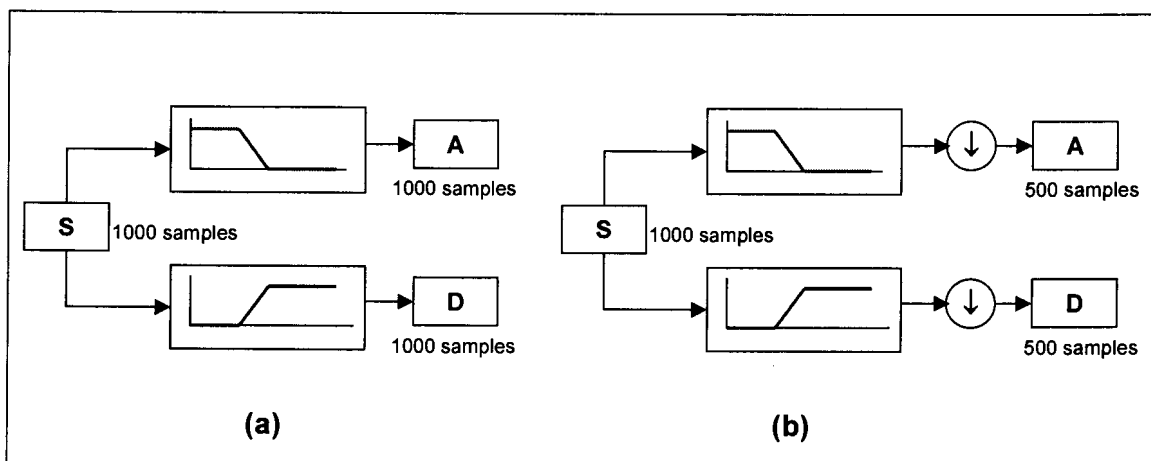


Figure 6.4(a) Wavelet decomposition without downsampling, and (b) with downsampling.

The process in Figure 6.4(b), which includes downsampling, produces discrete wavelet transform (DWT) coefficients. The detail coefficients will consist mainly of the high-frequency noise, while the approximation coefficients will contain much less noise than does the original signal.

The actual lengths of the detail as well as the approximation coefficient vectors will be slightly more than half the length of the original signal. This has to do with the filtering process, which is implemented by convolving the signal with a filter. The convolution “smears” the signal, introducing several extra samples into the result.

6.5.3. MULTIPLE-LEVEL DECOMPOSITION

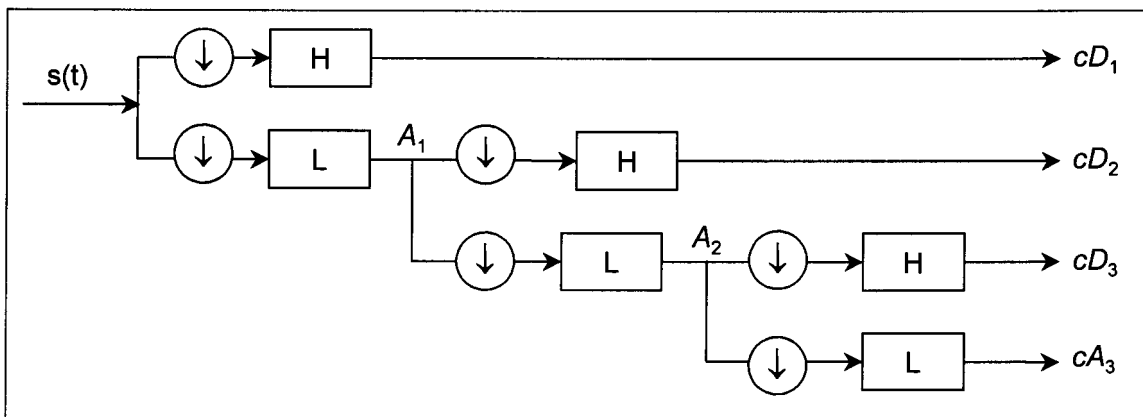


Figure 6.5 Multilevel decomposition tree (An octave band non-subsampled filter bank.)

The decomposition process can be iterated, with successive approximations being decomposed in turn, so that one signal is broken down into many lower-resolution components. This is called the wavelet decomposition tree, illustrated in Figure 6.5, which can yield valuable information.

Since the analysis process is iterative, in theory it can be continued indefinitely. In reality, the decomposition can proceed only until the individual details consist of a single sample or pixel. In practice, you’ll select a suitable number of levels based on the nature of the signal, or on a suitable criterion such as entropy.

After calculating the wavelet coefficients, these coefficients can be thresholded to remove noise prior to reconstruction. Wavelet thresholding is discussed in more detail in Section 6.6, but it is worth noting that this step is applied after calculating the wavelet coefficients.

6.5.4. WAVELET RECONSTRUCTION

The process of assembling the components back into the original signal with no loss of information is called reconstruction, or synthesis. The mathematical manipulation that affects synthesis is called the inverse discrete wavelet transform (IDWT). Where wavelet analysis involves filtering and downsampling, the wavelet reconstruction process consists of upsampling and filtering. Upsampling is the process of lengthening a signal component by inserting zeros between samples.

The filtering part of the reconstruction process is crucial since achieving perfect reconstruction of the original signal depends on the choice of filters. In the case of a discrete wavelet transform, reconstruction of the original signal is not guaranteed. Recall that the downsampling of the signal components performed during the decomposition phase introduces a distortion called aliasing. It turns out that by carefully choosing filters for the decomposition and reconstruction phases that are closely related (but not identical), we can “cancel out” the effects of aliasing. This was the breakthrough made possible by the work of Daubechies (1992) who developed conditions under which $\{\psi_{m,n}\}$ forms an orthonormal basis. A technical discussion of how to design these filters can be found in p. 347 of the book *Wavelets and Filter Banks*, by Strang and Nguyen (1995). Usually, $a_0 = 2$ and $b_0 = 1$ are used, although any values can be used. In this case, both the transform and reconstruction are complete because the family of wavelets form an orthonormal basis. The low- and highpass decomposition filters (L and H), together with their associated reconstruction filters (L' and H'), form a system of what is called quadrature mirror filters.

6.5.5. RECONSTRUCTING APPROXIMATIONS AND DETAILS

So it is possible to reconstruct the original signal from the coefficients of the approximations and details. It is also possible to reconstruct the approximations and details themselves from their coefficient vectors. As an example, let's consider how we would reconstruct the first-level approximation A_1 from the coefficient vector cA_1 . We pass the coefficient vector cA_1 through the same process we used to reconstruct the original signal. However, instead of combining it with the level-one detail cD_1 , we feed in a vector of zeros in place of the details as in Figure 6.6.

The process yields a reconstructed approximation A_1 , which has the same length as the original signal s and which is a real approximation of it. Similarly, we can reconstruct the first-level detail D_1 , using the analogous process illustrated in Figure 6.7.

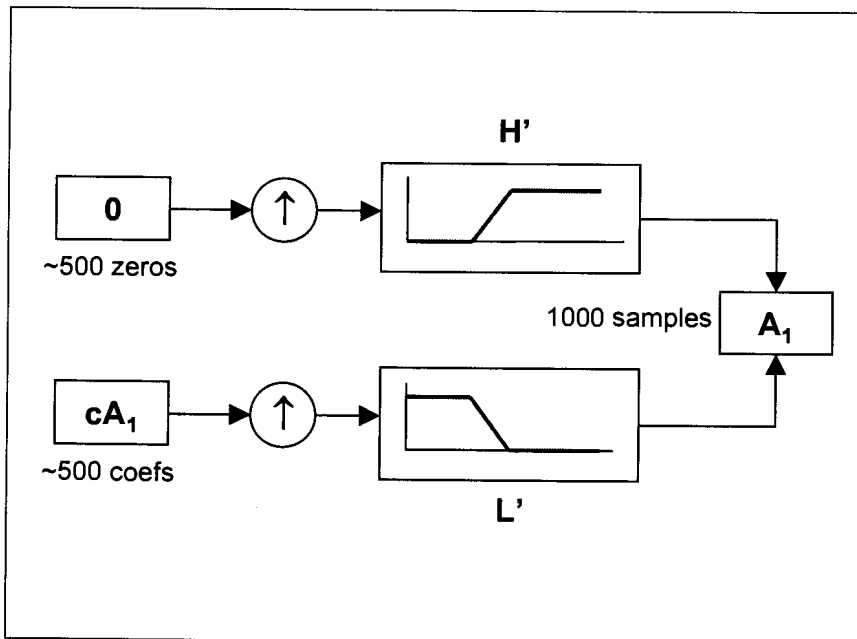


Figure 6.6. First-level reconstruction of approximation

The reconstructed details and approximations are true constituents of the original signal. In fact, we find when we combine them that:

$$A_1 + D_1 = S$$

Note that the coefficient vectors cA_1 and cD_1 — because they were produced by downsampling, contain aliasing distortion, and are only half the length of the original signal — cannot directly be combined to reproduce the signal. It is necessary to reconstruct the approximations and details before combining them.

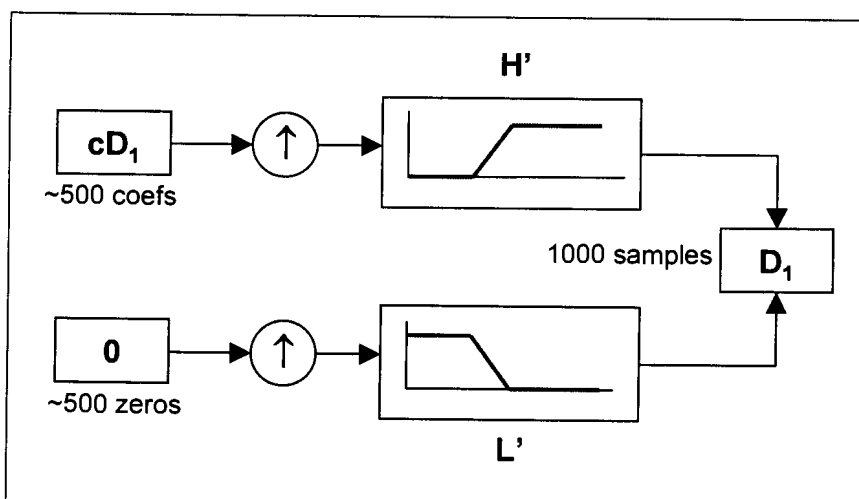


Figure 6.7. First-level reconstruction of detail

Extending this technique to the components of a multi-level analysis, we find that similar relationships hold for all the reconstructed signal constituents. That is, there are several ways to reassemble the original signal:

$$S = A_1 + D_1$$

$$S = A_2 + D_2 + D_1$$

$$S = A_3 + D_3 + D_2 + D_1$$

6.5.6. FILTERS USED TO CALCULATE THE DWT AND IDWT

For an orthogonal wavelet, in the multiresolution framework, we start with the scaling function ϕ and the wavelet function ψ . One of the fundamental relations is the twin-scale relation (dilation equation or refinement equation):

$$\frac{1}{2}\phi\left(\frac{x}{2}\right) = \sum_{n \in \mathbb{Z}} w_n \phi(x-n) \quad (6.9)$$

All the filters used in DWT and IDWT are intimately related to the sequence $(w_n)_{n \in \mathbb{Z}}$. Clearly if ϕ is compactly supported, the sequence (w_n) is finite and can be viewed as a filter. The filter W , which is called the scaling filter (non-normalized), is:

- Finite Impulse Response (FIR)
- of length $2N$
- of sum 1
- of norm $\frac{1}{\sqrt{2}}$
- a low-pass filter

From filter W , we define four FIR filters, of length $2N$ and of norm 1, organized as in Table 6.1.

The four filters are computed using the scheme in Figure 6.8 where qmf is such that H' and L' are quadrature mirror filters (i.e., $H'(k) = (-1)^k L'(2N-1-k)$). Note that $wrev$ flips the filter coefficients so H and L are also quadrature mirror filters.

Table 6.1. Filter representation

Filters	Low-pass	High-pass
Decomposition	L	H
Reconstruction	L'	H'

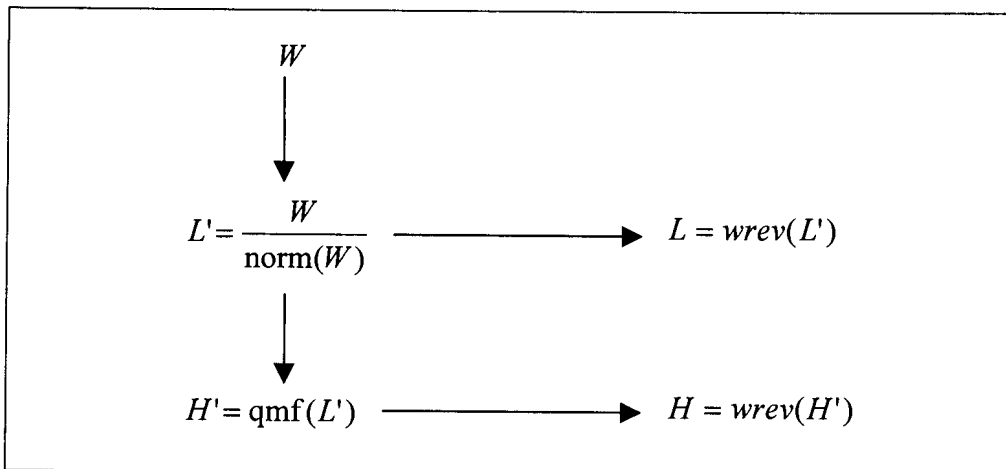


Figure 6.8. Wavelet filter computing scheme

6.6. Wavelet Denoising through Thresholding

To address the issue of noise, wavelet de-noising can be applied where the wavelet coefficients cA_i and cD_i are thresholded in order to remove noise components contained in the signal and thus also in the wavelet coefficients.

Multiscale rectification using wavelets is based on the observation that random errors in a signal are present over all the coefficients, while deterministic changes get captured in a small number of relatively large coefficients. Thus, stationary Gaussian noise may be removed by suppressing coefficients smaller than a selected value (Donoho et al., 1995).

Donoho and coworkers have studied the statistical properties of wavelet thresholding and have shown that for a noisy signal of length n , the rectified signal will have an error of order $\log n$ of the error between the error-free signal and the signal rectified with a-priori knowledge about the smoothness of the underlying signal (Donoho and Johnstone, 1994).

Generally speaking, wavelet thresholding can be divided into two categories: *global thresholding* and *level-dependent thresholding*. If the threshold value is denoted as λ , then in global thresholding a single value of λ is selected and is applied globally to all empirical wavelet coefficients above a certain frequency level. For *leveldependent thresholding*, a different threshold value λ_j can be selected for the wavelet coefficient at level j . This approach is necessary when the noise in the data is non-stationary and/or correlated and is the approach used in this study.

Selecting the proper value of the threshold is a critical step in the rectification process and a number of different methods for selecting appropriate threshold values for wavelet denoising have been proposed in the literature (e.g. Donoho and Johnstone, 1994,1995; Donoho, 1995; Donoho et al., 1996; Hall et al., 1996; Hall and Patil, 1996; Nason, 1996).

Generally, wavelet denoising methods are based on either a hard or a soft thresholding approach. If the threshold value is denoted as λ , hard thresholding is given by Equation 6.10, whilst soft thresholding is given by Equation 6.11. Soft thresholding shrinks the value of the wavelet coefficients towards zero (eliminates coefficients) if they are above a certain threshold and hard thresholding if they are smaller.

$$\delta_{\lambda}^H(x) = \begin{cases} x & \text{if } |x| > \lambda \\ 0 & \text{otherwise} \end{cases} \quad (6.10)$$

$$\delta_{\lambda}^S(x) = \begin{cases} x - \lambda & \text{if } x > \lambda \\ 0 & \text{if } |x| \leq \lambda \\ x + \lambda & \text{if } x < -\lambda \end{cases} \quad (6.11)$$

Hard thresholding can lead to better reproduction of peak heights and discontinuities, but at the price of occasional artifacts that can roughen the appearance of the rectified signal, while soft thresholding usually gives better visual quality of rectification and less artifacts. An artifact, which is not present in the original signal, is created in the reconstructed signal when the wavelet function used to represent a feature in the signal and the feature itself does not align. Such artifacts are due to a localized Gibbs phenomenon which is caused by the lack of translational invariance in orthonormal wavelet decomposition.

Two factors that can influence the performance of wavelet thresholding are considered in the selection of the threshold values, these are the sample size N and the noise level σ . For good visual quality of the rectified signal, the VisuShrink method determines the threshold as

$$t_j = \sigma_j \sqrt{2 \log N} \quad (6.12)$$

where N is the signal length and σ_j is the standard deviation of the errors at scale j .

In practice, the value of the standard deviation of the noise in the data, σ , is unknown and is replaced by an estimate $\hat{\sigma}$. Donoho and Johnstone (1995) proposed the use of the median of the absolute deviation (MAD) of the wavelet coefficients at the finest level (level=1):

$$\hat{\sigma} = \text{median}(|w_{1,i}|) / 0.6745 \quad (6.13)$$

where $i = 0, \dots, 2^{J-1} - 1$, $J = \log_2(N)$. The median absolute deviation of the coefficients is a robust estimate of σ . When coloured noise is suspected, the noise level σ needs to be estimated level-by-level using a similar kind of strategy and the threshold values also need to be modified according to the level-dependent estimation of the noise.

Wavelet de-noising is able to remove as much noise as required but not at the expense of smoothing out any real fine-scale features (Ogden, 1997). The advantage of spatially adaptive methods such as wavelet de-noising is that they perform close to the optimum across the whole range of noise levels, no matter the smoothness of the signal. On the other hand, the best performing median filter is almost as efficient as the wavelet de-noising methods at relatively high signal to noise ratios, if the window size is selected appropriately. However, for low signal-to-noise ratios, phase-shift may result. Moreover, future values are needed to apply the median filtering algorithm, thus making it unsuitable for on-line application and therefore wavelet de-noising remains a better alternative.

Wavelet-based multiscale rectification is a very effective approach for denoising signals contaminated by white, as well as correlated Gaussian noise. If the traditional wavelet decomposition algorithm is applied to a signal with non-Gaussian errors, outliers will be present at multiple scales in both the scaled and detailed signals, and large coefficients corresponding to outliers get confused with those corresponding to important features. Thus, wavelet thresholding is not effective in eliminating non-Gaussian errors. This limitation may be overcome by combining wavelet thresholding with multiscale median filtering as in the robust multiscale rectification technique (Bruce et al., 1994).

6.7. Algorithms

This section takes you through the most important steps of the wavelet analysis and de-noising algorithms in view of the actual implementation. It considers in more detail the magnitude and nature of the different calculated values and signals.

Starting out with a signal s of length N , the DWT consists of $\log_2 N$ stages at most. The first step produces, starting from s , two sets of coefficients: approximation coefficients cA_1 and detail coefficients cD_1 . These vectors are obtained by convolving s with the low-pass filter L for approximation, and with the high-pass filter H for detail, followed by dyadic decimation. The first step is illustrated by Figure 6.9.

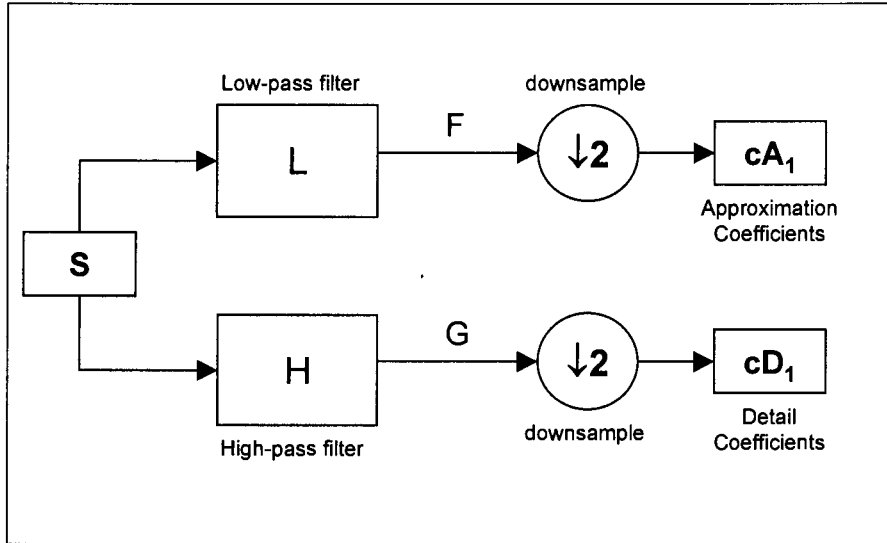


Figure 6.9. First step in the wavelet analysis algorithm

The length of each filter is equal to $2N$. If $n = \text{length}(s)$, the signals F and G , are of length $N + 2N - 1$ and then the coefficients cA_1 and cD_1 are of length

$$\text{floor}\left(\frac{n-1}{2}\right) + N.$$

The next step splits the approximation coefficients cA_1 in two parts using the same scheme, replacing s by cA_1 , and producing cA_2 and cD_2 , and so on as illustrated in Figure 6.5. So the wavelet decomposition of the signal s analyzed at level j has the following structure: $[cA_j, cD_j, \dots, cD_1]$.

The next step involves applying level-dependent thresholding to the coefficients so that the wavelet decomposition of the signal s analyzed at level j now has the following structure: $[cA'_j, cD'_j, \dots, cD'_1]$, where cA'_j and cD'_j are the thresholded approximation and detail wavelet coefficients.

Conversely, starting from cA'_j and cD'_j , the IDWT reconstructs A'_{j-1} , the reconstructed approximation signal, inverting the decomposition step by inserting zeros and

convolving the results with the reconstruction filters as depicted in Figure 6.10 where *wkeep* means taking the central part of *U* with the convenient length.

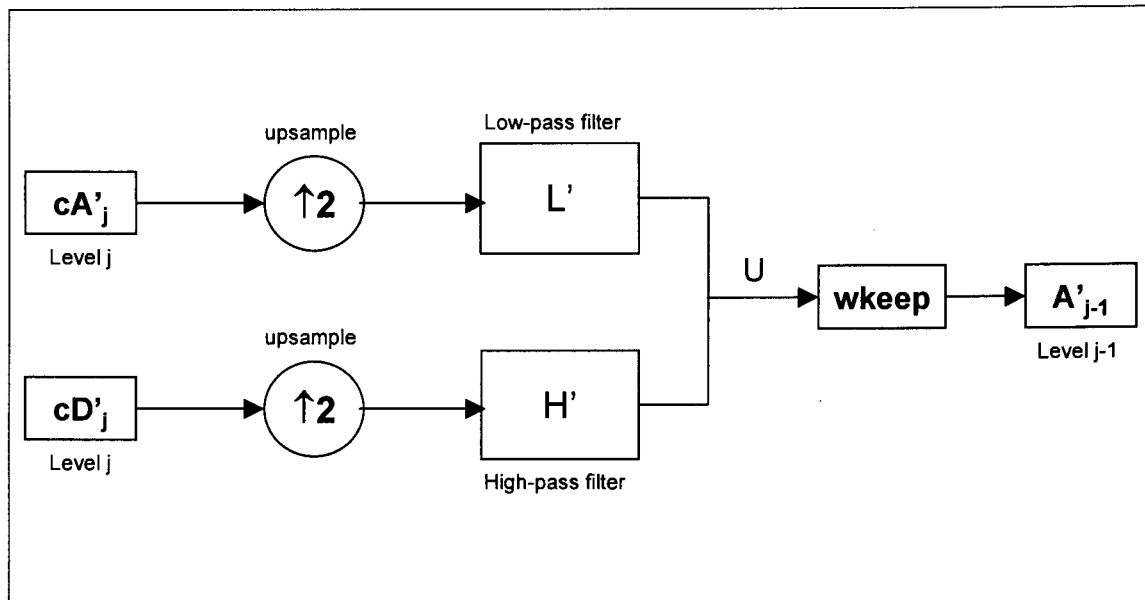


Figure 6.10. Reconstruction of the wavelet coefficients using reconstruction filters

6.8. On-Line Multiscale Rectification

Existing nonlinear rectification techniques do perform better than linear filters for a broad variety of signals. However, a significant disadvantage of these nonlinear multiscale methods is that they cannot be implemented online. In general wavelet filters are noncausal in nature and require future measured data for calculating the current wavelet coefficient. This introduces a time delay in the computation that increases at coarser scales and smoother filters. This time delay may be overcome in a rigorous manner by using special wavelets at edges that eliminate boundary errors while being orthonormal to the other wavelets (Cohen, et al., 1993). These boundary corrected filters are causal and require no information about the future to compute wavelet coefficients at the signal end points. Another reason for restricting the wavelet-based methods to off-line use is the dyadic discretization of the wavelet parameters, which requires a signal of dyadic length for the wavelet decomposition.

A signal containing a dyadic number of measurements can be decomposed as shown in Figure 6.11(a). In contrast, if the number of measurements is odd, the last point cannot be decomposed without a time delay as shown in Figure 6.11(b). In many applications such a time delay is unacceptable. Consequently, this section describes an online method for multiscale rectification (OLMS), where absolutely no time delay is allowed.

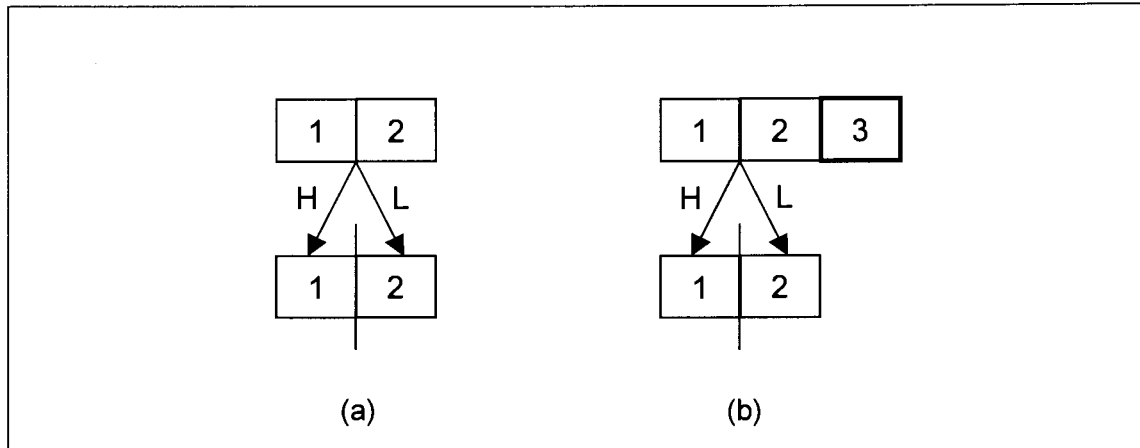


Figure 6.11. Time delay introduced due to dyadic length requirement in wavelet decomposition

On-line multiscale rectification is based on multiscale rectification of data in a moving window of dyadic length, as shown in Figure 6.12. The OLMS methodology can be summarized as follows:

- (1) Decompose the measured data within a window of dyadic length using a causal boundary corrected wavelet filter.
- (2) Threshold the wavelet coefficients and reconstruct the rectified signal.
- (3) Retain only the last data point of the reconstructed signal for on-line use.
- (4) When new measured data are available, move the window in time to include the most recent measurement while maintaining the maximum dyadic window length.

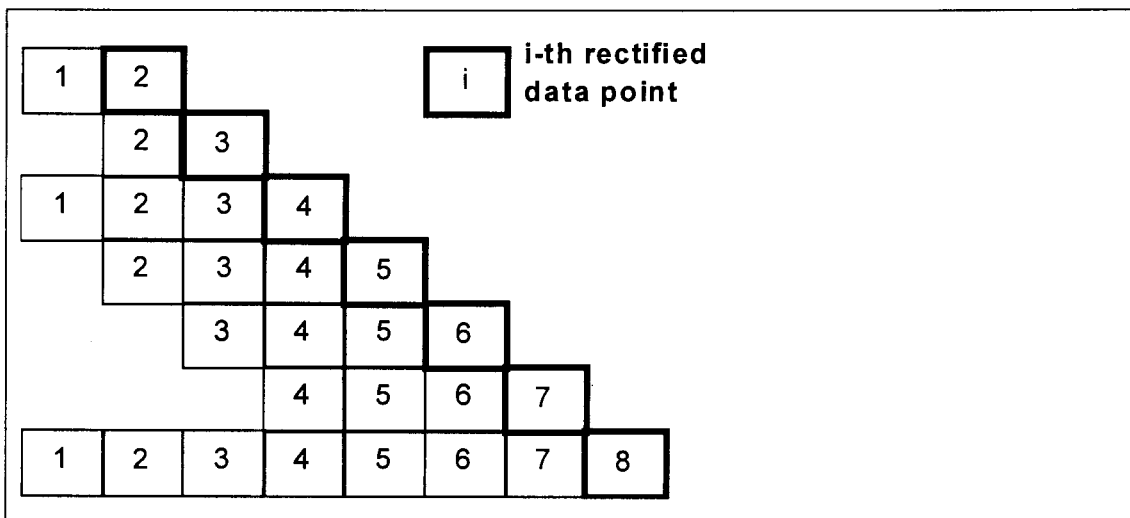


Figure 6.12. OLMS rectification

The measurements in each window are rectified by the wavelet thresholding approach of Donoho et al. (1995) described in the previous section. This simple approach is very effective compared to the single-scale techniques and retains the benefits of the wavelet decomposition in each moving window, while allowing each measurement to be rectified on-line.

6.9. Practical issues of OLMS

Any filtering method requires typical data or information about the underlying signal and noise for selecting the filter parameters. In OLMS rectification the filter tuning parameters are the value of the threshold and the maximum depth of the wavelet decomposition. Other practical issues include selecting a wavelet and the maximum length of the moving window.

6.9.1. VALUE OF THRESHOLD

The threshold may be estimated by applying the Visushrink method (Donoho et al., 1995; Nason, 1996; Nounou and Bakshi, 1999) to the available measurements. For data corrupted by stationary errors, the threshold value stops changing much after an adequate number of measurements are available. Consequently, for stationary noise, the threshold may be estimated from the measurements until the change is below a user-specified value. This approach for estimating the threshold cannot be performed recursively due to the median operator used in Equation 6.9 and will require storage of a large number of measurements.

6.9.2. DEPTH OF DECOMPOSITION

Thresholding wavelet coefficients at very coarse scales may result in the elimination of important features, whereas thresholding only at very fine scales may not eliminate enough noise. Therefore, the depth of wavelet decomposition needs to be selected to optimize the quality of the rectified signal. Empirical evidence suggests that a good initial guess for the decomposition depth is about half of the maximum possible depth, that is $(\log_2(n))/2$ where n is the moving window length. However, a smaller depth might be more appropriate in OLMS rectification if a long boundary corrected filter with a large support is used in the decomposition since the filters at the two edges might overlap at very coarse scales. The depth may also be determined by cross-validation.

6.9.3. SELECTED WAVELET FILTER

The type, length, and nature of the wavelet filter used in OLMS affect the quality of the rectification. Since the OLMS rectification uses only the last rectified data point from each translated signal, it is crucial that only boundary corrected causal wavelet filters are used. If boundary corrected filters are not used, then the last point is among the least accurate ones due to the end effect errors. OLMS rectification using Daubechies second-order boundary corrected filters was used and results in smaller mean-square error than OLMS rectification using other simpler wavelet filters like the Haar wavelet.

6.10. Application

6.10.1. SOFTWARE SETUP

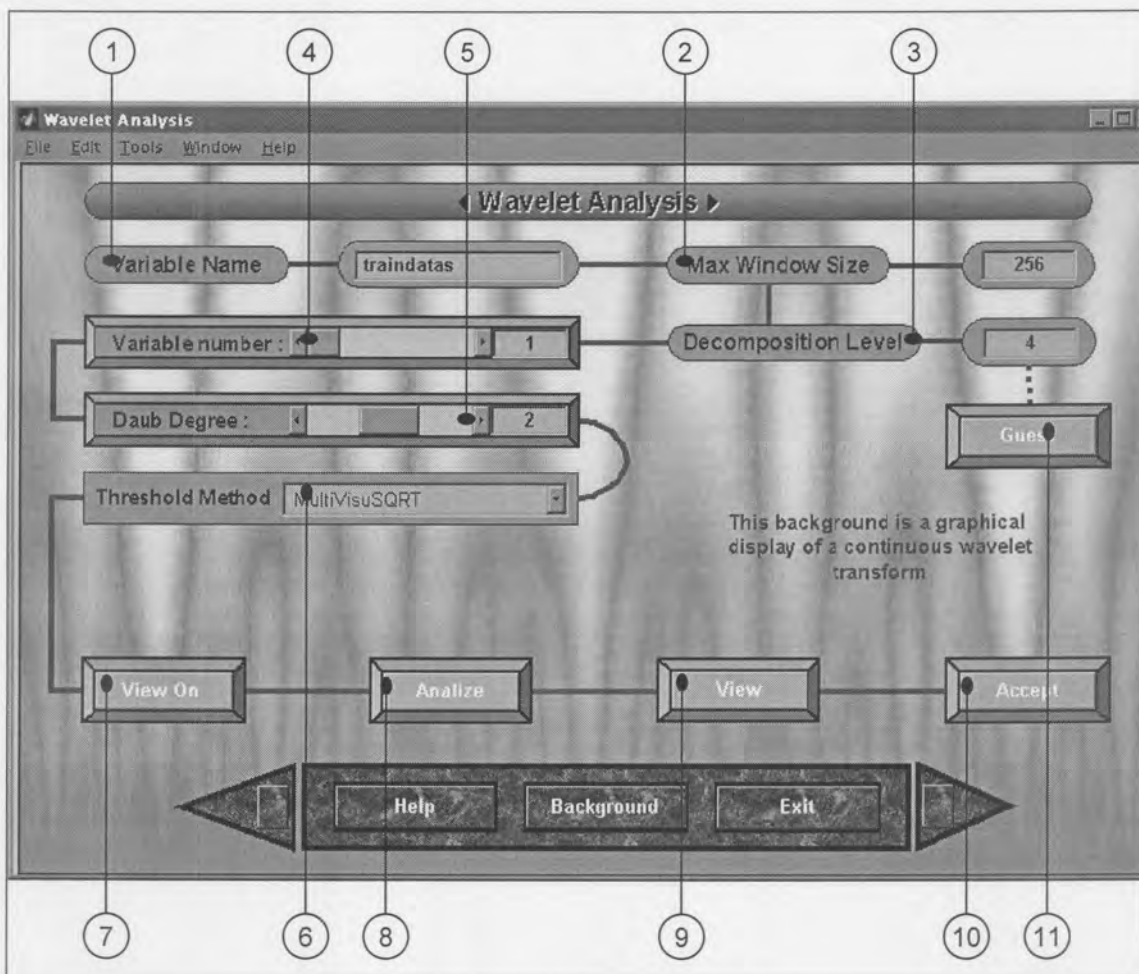


Figure 6.13. Wavelet analysis main interface

The wavelet analysis user interface is displayed by using the *Next* button on the data setup interface or it can be accessed via the main user interface. The options are related to the

theory discussed in the previous sections. Normally one would have to play around with different combinations of the parameters in order to select the best combination since there are no definite rules.

Figure 6.13 Tags:

1. Name of variable to which the wavelet transform should be applied. By default *traindatas* from the database is used. Any other variable name may be specified. However, it is important to know that the data should be normalized or standardized prior to applying the wavelet analysis.
2. Maximum dyadic (power of two) window size. For this application a maximum window size of 256 (2^8) was used.
3. Multiresolution decomposition level.
4. Variable number. The wavelet analysis is applied to one variable at a time. The specific variable is specified via its column number in the data matrix.
5. Daubechies degree. In most cases the second (2) degree works the best.
6. Type of threshold to apply to the wavelet coefficients. Those methods with *multi* as prefix refer to level dependent thresholding.
7. Toggle between real-time viewer on an off. If the viewer is on, one is able to view the coefficients and reconstructed approximations and details as they are calculated as illustrated in Figure 6.14 and Figure 6.15.
8. Apply wavelet analysis. This starts the wavelet analysis process.
9. View reconstructed multiresolution data.
10. Write results to database.
11. Initial guess for decomposition level.

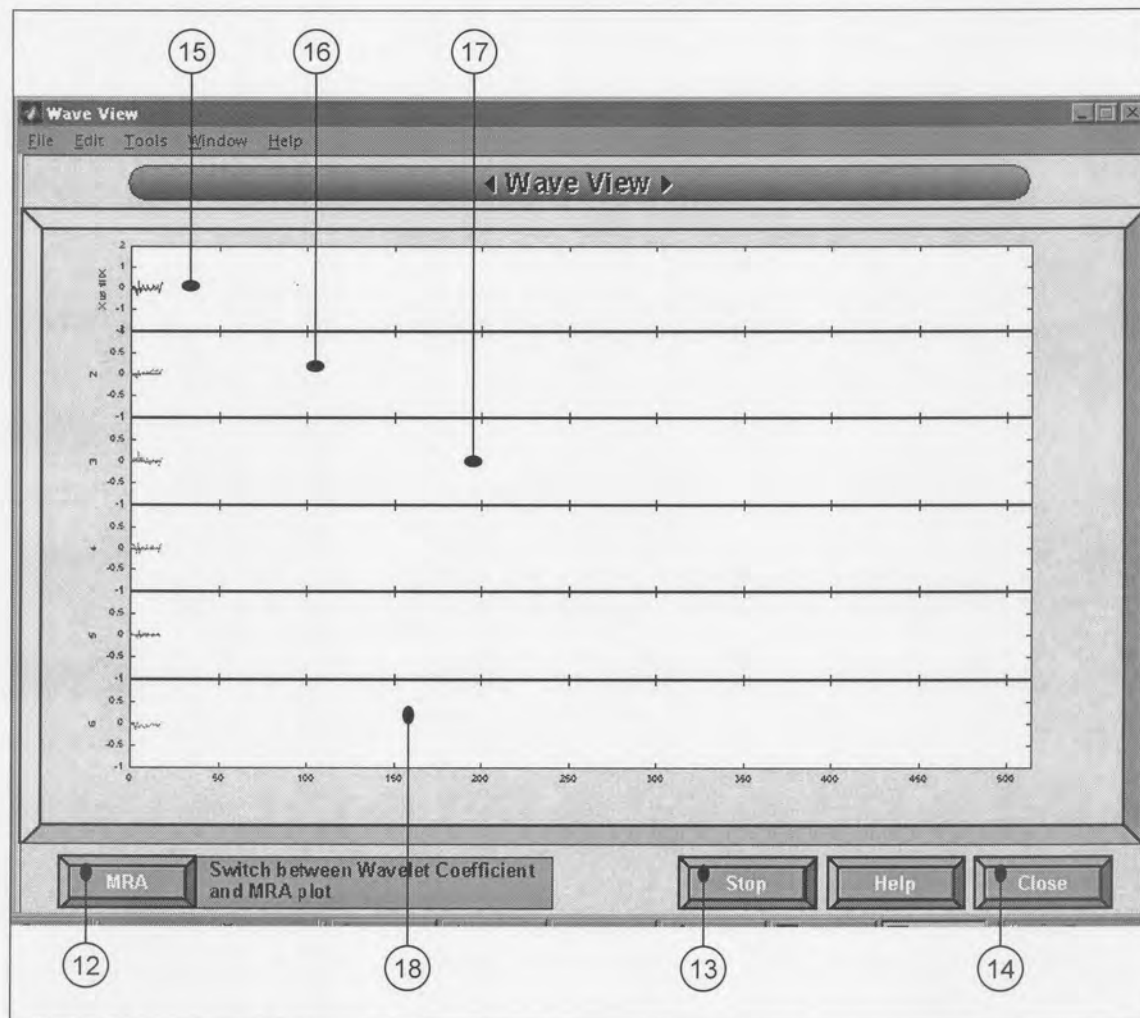


Figure 6.14. Wavelet analysis viewer – Reconstructed details and approximation

Figure 6.14 Tags:

12. Switch between multiresolution analysis (Figure 6.14) and wavelet coefficient (Figure 6.15) plot.
13. Stop multiresolution analysis.
14. Close the plotting window.
15. Original (black) and reconstructed approximation/filtered (red) data plot.
16. Reconstructed detail signal at finest scale/level before (cyan) and after (blue) thresholding
17. Reconstructed detail signal at second scale/level before (cyan) and after (blue) thresholding
18. Reconstructed detail signal at coarsest scale/level before (cyan) and after (blue) thresholding

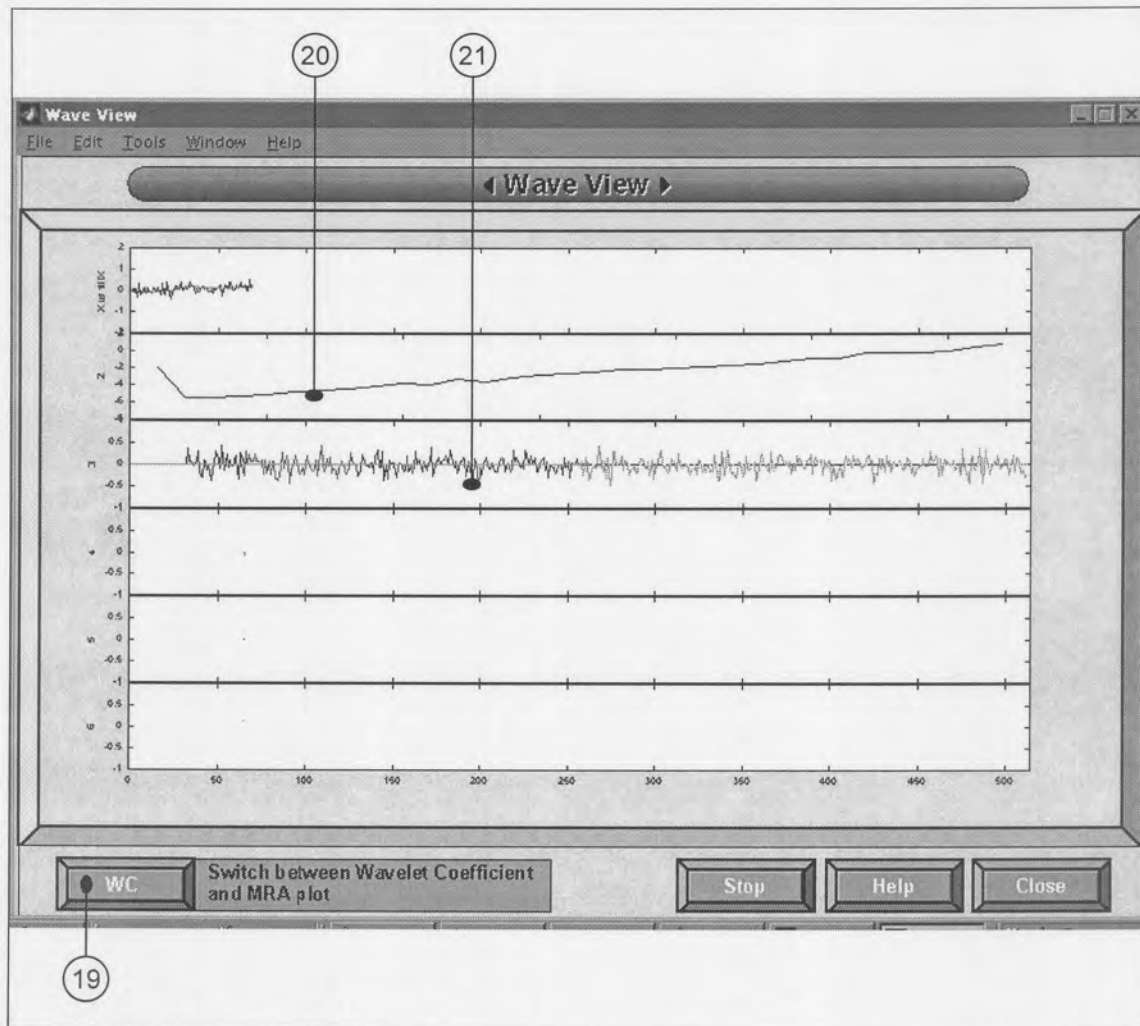


Figure 6.15. Wavelet analysis viewer – Wavelet coefficients

Figure 6.15 Tags:

- 19. Switch back to multiresolution analysis plot
- 20. Approximation coefficients of dyadic window after each time interval
- 21. Detail wavelet coefficient plot of dyadic window after each time interval.

Figure 6.16 is a completed version of Figure 6.14. This was the multiresolution wavelet analysis of variable one.

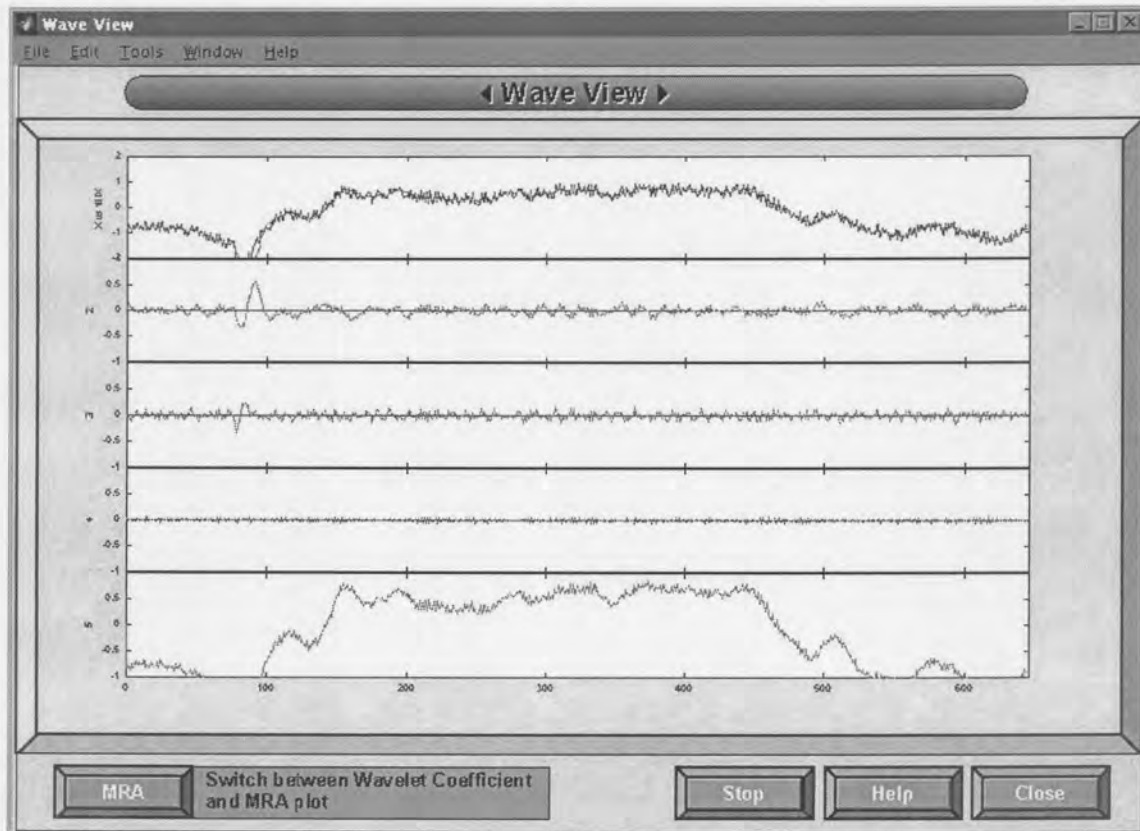


Figure 6.16. Completed multiresolution analysis of variable one.

Figure 6.17 Tags:

22. The name of the variable containing the thresholded multiresolution data for a specific level. Here *thmra_level_1* refers to the thresholded multiresolution data of level one which is a data matrix containing the first detail level of all the variables each in a separate column.
23. The nonthresholded (black) and thresholded (red) reconstructed detail level of the level specified by tag 22 and variable number specified by tag 24. Here the nonthresholded and thresholded detail of detail level one of variable one can be viewed.
24. Variable number of which the information is required.
25. Here the effect of removing or adding the specific nonthresholded detail level can be viewed. The black plot represents the original nonthresholded reconstructed signal and the red plot the thresholded signal with the added effect of removal or adding of a nonthresholded detail level. This is used if one wishes to override the thresholding of a specific level. During thresholding a specific detail level may be zeroed (removed) as in this example. The user may however decide that the specific level is significant and that it contains important information. In such a case the detail level may be replaced and the effect on the final signal can be viewed.

Thresholding may also retain some information in the detail levels that the user may decide is insignificant, in which case it can be removed.

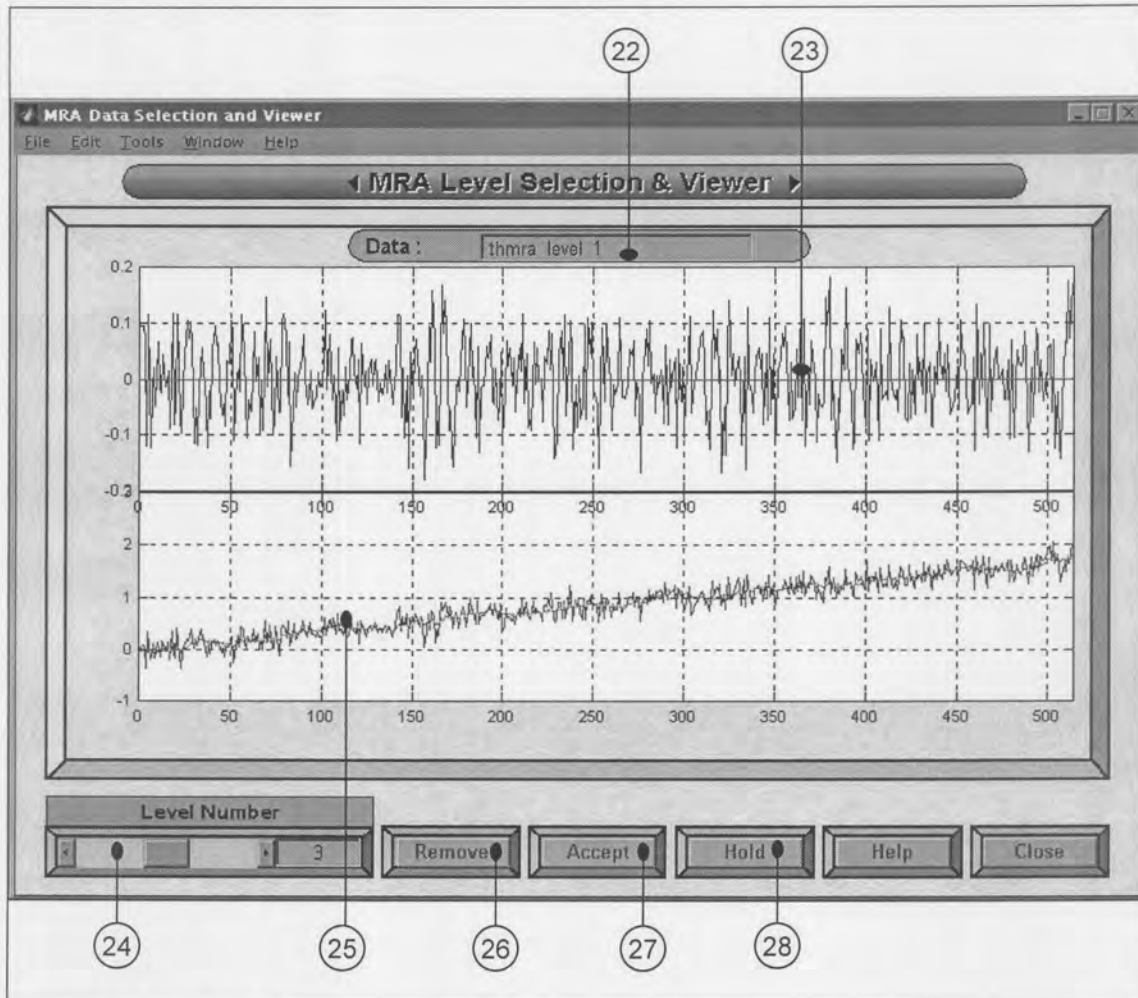


Figure 6.17. MRA data selection and viewer

- 26. Toggle between removing and adding a specific detail level.
- 27. Accept the configuration and save it to the database.
- 28. Use this if more than one detail level at a time needs to be viewed.

6.10.2. EXPERIMENTAL

6.10.2.1. Wavelet Analysis

The eight standardized variables from chapter 5 were decomposed into their contributions in different regions of the time-frequency space by projection on the corresponding wavelet basis function, as depicted in Figure 6.18 for variable one.

Figure 6.18 represents a moving window width of 256 data samples at a given time instance.

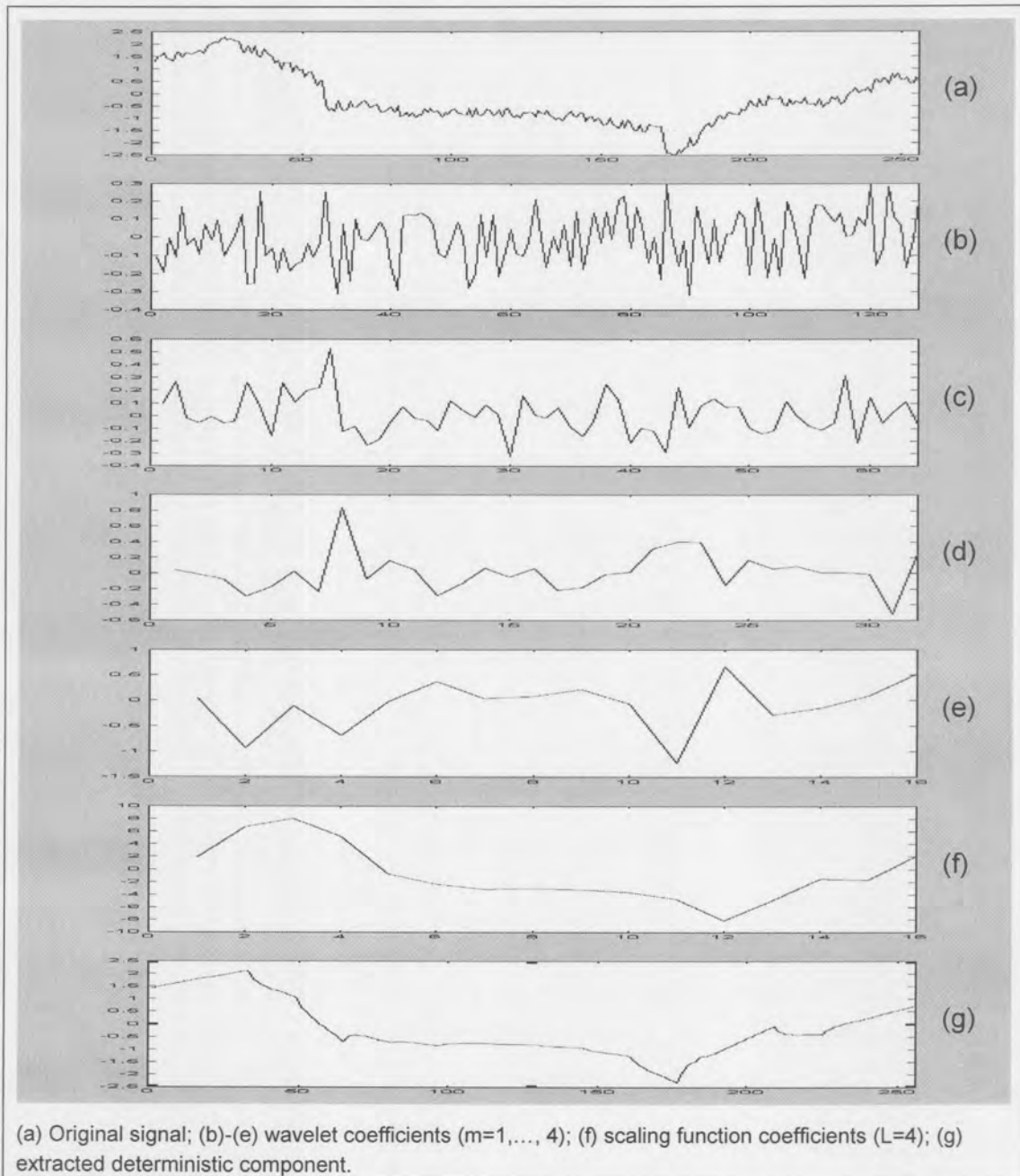


Figure 6.18. Wavelet decomposition and separation of stochastic and deterministic components.

The high-scale, low frequency content (approximation) of variable one is represented on a set of scaling functions, as depicted in Figure 6.18(f). The low-scale, high frequency content (detail) of variable one is illustrated by Figure 6.18(b)-(e).

The same process was repeated for each variable during each time interval.

6.10.2.2. Multiresolution decomposition

Multiresolution decomposition based on wavelets was carried out for each variable to observe both the general trend and the detailed features of the process data. The discrete fast wavelet transform using a boundary-corrected Daubechies second order filter at level $L = 4$, which is half the maximum length, was used.

Decomposition at level 4: $s = A_4 + D_4 + D_3 + D_2 + D_1$

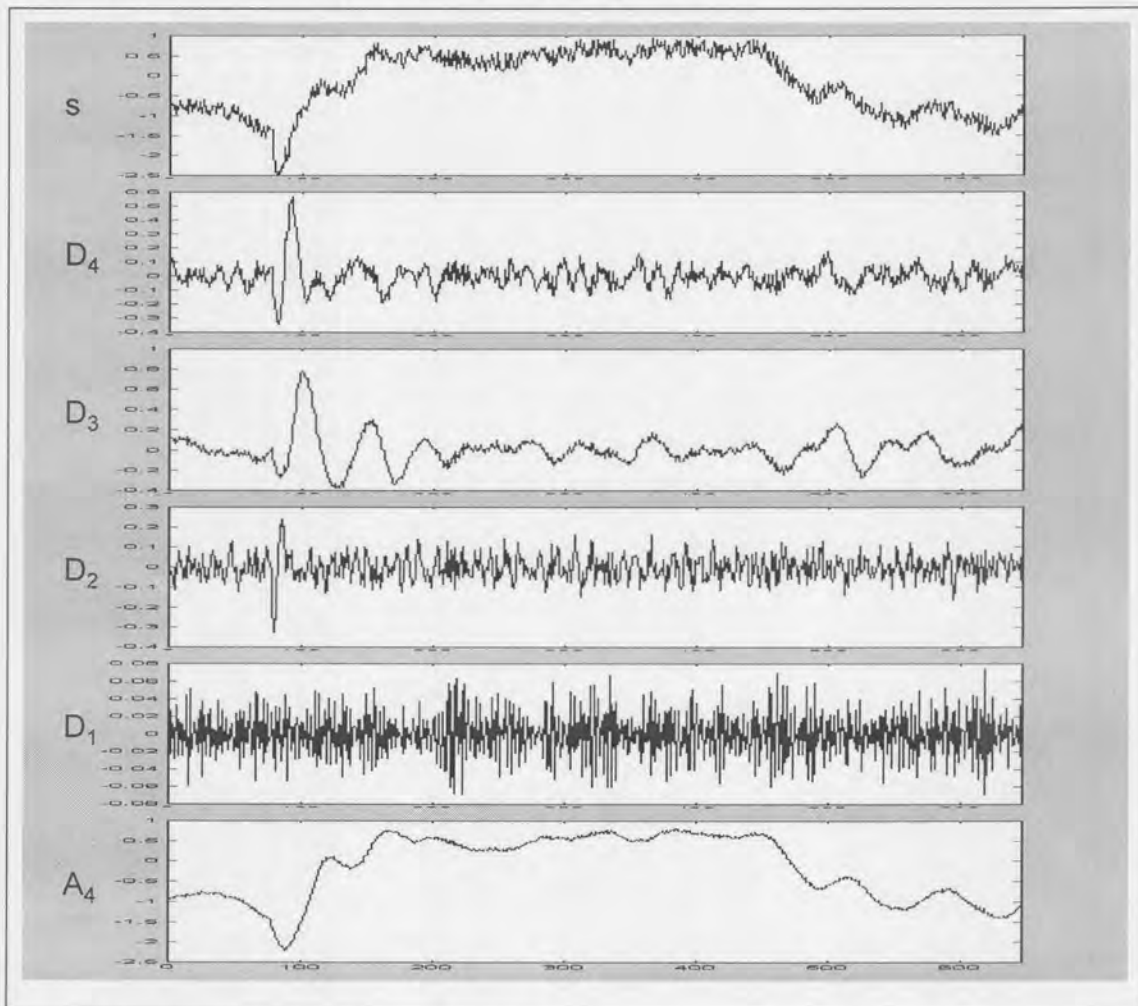


Figure 6.19. Multiresolution analysis plot

The length of the dyadic window was chosen as 256 which is a power of two (2^8) since classically the discrete wavelet transform is defined for sequences with length of some power of two. Note that this initial window length is not a restriction. Theoretically, OLMS rectification can start with any dyadic set of measurements, starting at two. However, since the threshold is estimated from the data in the moving window, the threshold estimate improves as the moving window length increases. When the noise is assumed to be stationary, the threshold stops changing after a large set of measurements are collected, and, thus, the moving window length can be held constant. Figure 6.19 shows the multiresolution analysis plot for process variable one. Approximation coefficients at scale 4 (A_4) represent the underlying trend of the signal(s)

whilst wavelet coefficients ($D_4 \sim D_1$) show the high-frequency details. Examining the multiresolution analysis results for all eight process variables (see Paragraph 3.11 of Chapter 3), the level of noise corruption was found to be different for each variable necessitating level dependent thresholding.

This was repeated for each of the eight variables and provided similar results for each of the variables.

6.10.2.3. Wavelet thresholding

Wavelet thresholding based on hard thresholding was then used to remove the high-frequency noise as well as the spikes known to be outliers. Level-dependent threshold values were derived from the VisuShrink threshold strategy. In this manner both noise and spikes were removed from the signal without affecting the underlying process trends. The thresholding zeroed all the detail coefficients indicating that all the detail could be attributed to noise. The approximation coefficients obtained in Figure 6.19 for variable one and all the other variables preserve the process trend well in a compact form since all high-frequency elements are omitted (Shimizu et al., 1997).

6.9.2.4. Multilevel signal reconstruction

The thresholded and non-thresholded wavelet coefficients were used to construct thresholded and non-thresholded approximations and details. The non-thresholded details and approximations were combined to form dataset 2 from which the combined principal component model was derived. At each level the thresholded details were investigated to see if they contained any significant contributions. The significant contributions were combined according to the level from which the detail principal component model for each separate level was derived. The investigation revealed that the current thresholded details did not contain any significant contributions so that no need existed to derive detail principal component models. An approximation principal component model was derived from the combined approximations (dataset 1). By removing the undesirable high-frequency elements from the nominal data, the possibility of input-training network overfitting (Chapter 8) is greatly reduced. If by chance any desirable high-frequency elements were removed, it would be accounted for in the combined model.

Figure 6.20 gives dataset 1 which contains the approximations of all eight variables. Dataset 2 contains these approximations together with all the nonthresholded details of all eight variables. The methodology explained in this section is illustrated by Figure 6.21.

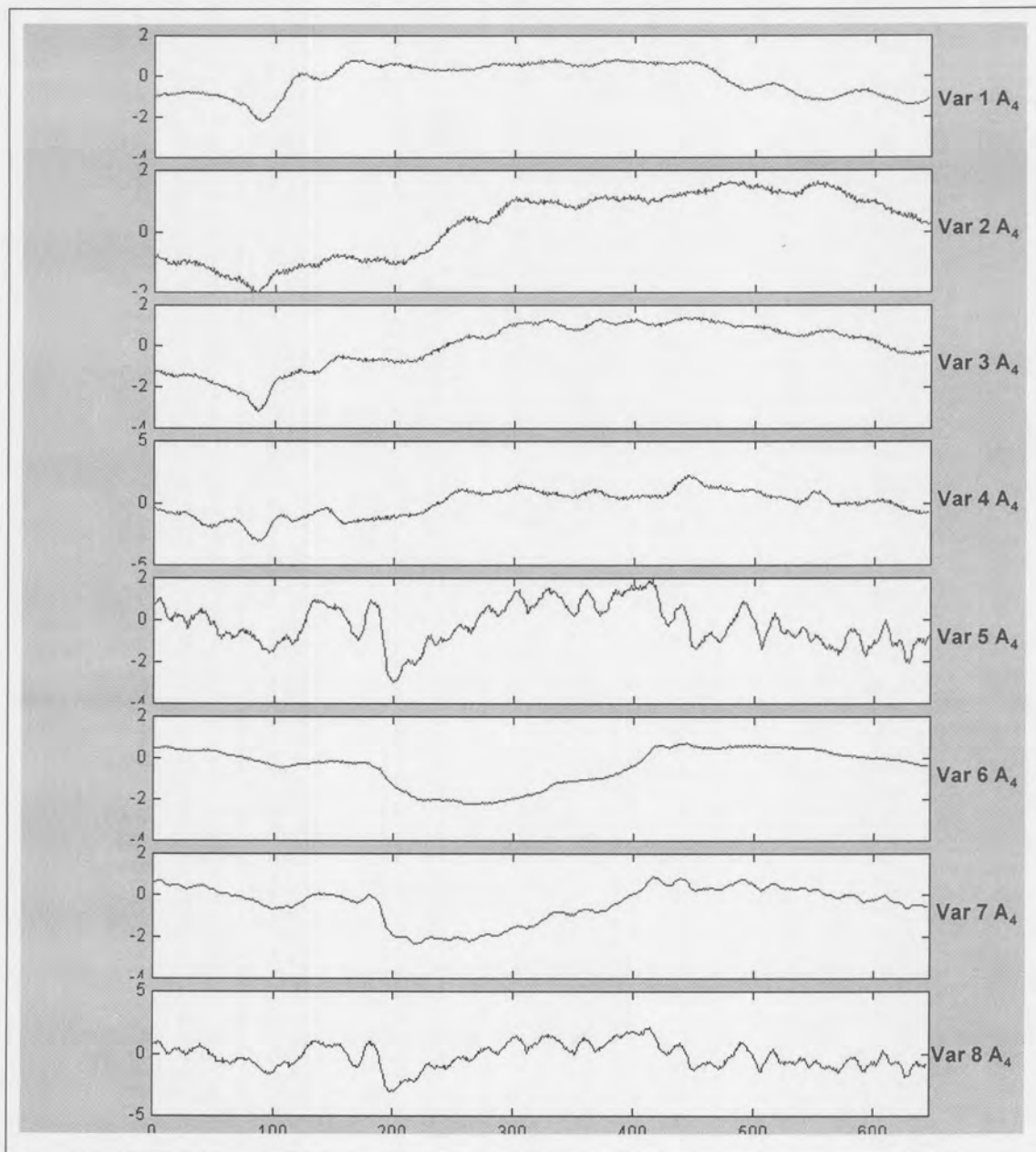


Figure 6.20. Dataset 1 containing the approximations of all eight variables.

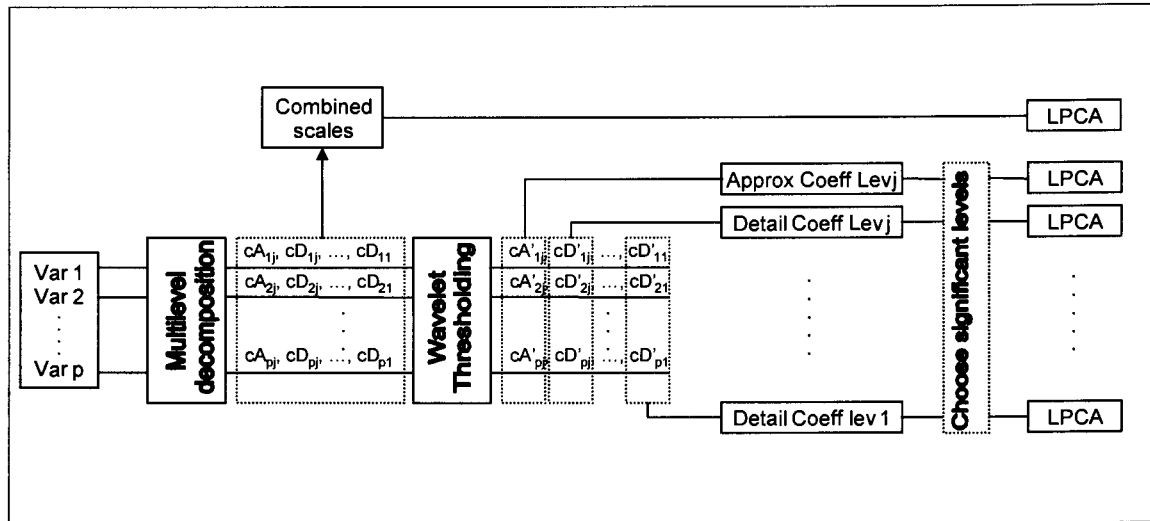


Figure 6.21. Comparison between the original signal, de-noised signal and the approximation coefficients used for model derivation

7.1. Introduction to linear principal component analysis (LPCA)

Principal component analysis is among the most popular methods for extracting information from data, which has been applied in a wide range of disciplines. In chemical process operation and control, PCA is used to solve several tasks including rectification (Kramer and Mah, 1994), gross-error detection (Tong and Crowe, 1995), disturbance detection and isolation (Ku et al., 1995), statistical process monitoring (Kresta et al., 1991; Wise et al., 1990), and fault diagnosis (MacGregor et al., 1994; Dunia et al., 1996). PCA is popular for process monitoring since it allows extension of the principles of univariate statistical process monitoring (SPM) to monitoring of multivariate processes (Jackson, 1980; Kresta et al., 1991).

Conventional PCA is best for analyzing a two-dimensional matrix of data collected from a steady-state process, containing linear relationships between the variables. Since these conditions are often not satisfied in practice, several extensions of PCA have been developed. Multiway PCA allows the analysis of a multidimensional matrix (Nomikos and MacGregor, 1994). Hierarchical or multiblock PCA permits easier modeling and interpretation of a large matrix by decomposing it into smaller matrices or blocks (Wold et al., 1996; MacGregor et al., 1994). Dynamic PCA extracts time-dependent relationships in the measurements by augmenting the data matrix by time-lagged variables (Kresta et al., 1991; Ku et al., 1995). Nonlinear PCA (Kramer, 1991; Hastie and Stuetzle, 1989; Dong and McAvoy, 1996; Tan and Mavrouniotis, 1995) extends PCA to extracting nonlinear relationships between the variables. On-line adaptive PCA updates the model parameters continuously by exponential smoothing (Wold, 1994).

The superficial dimensionality of data, or the number of individual observations constituting one measurement vector, is often much greater than the intrinsic dimensionality, the number of independent variables underlying the significant nonrandom variations in the observations (Kramer, 1991). The reduction of the data set from its superficial to intrinsic dimensions is the focus of principal component analysis.

Due to correlation, a few principal components are usually sufficient to capture the data variance (Dunia, 1999). Two kinds of abnormal conditions can be distinguished using PCA. These are:

- failure of sensor correlations: In this situation the PCA model is no longer valid and the Euclidean norm of the residual vector increases significantly.
- Excessive normal variance: The variables used to define the operating variability are out of the normal range, as suggested by the historical data.

7.2. Introduction to Multiscale PCA (MSPCA)

Modeling by PCA and its extensions is done at a single scale, that is, the model relates data represented on basis functions with the same time-frequency localization at all locations. For example, PCA of a time series of measurements is a single-scale model since it relates variables only at the scale of the sampling interval. Such a single-scale modeling approach is only appropriate if the data contains contributions at just one scale. Unfortunately, data from almost all practical processes are multiscale in nature due to:

- Events occurring at different locations and with different localization in time and frequency.
- Stochastic processes whose energy or power spectrum changes with time and/or frequency.
- Variables measured at different sampling rates or containing missing data

Consequently, conventional PCA is not ideally suited for modeling of most process data. Techniques have been developed for PCA of some types of multiscale data such as missing data, but the single-scale approach forces data at all scales to be represented at the finest scale, resulting in increased computational requirements.

Another shortcoming of conventional PCA and its extensions is that its ability to reduce the error by eliminating some components is limited, since an embedded error of magnitude proportional to the number of selected components will always contaminate the PCA model (Malinowski, 1991). This limited ability of PCA to remove the error deteriorates the quality of the underlying model captured by the retained components, and adversely affects the performance of PCA in a variety of applications. For example, in process monitoring by PCA, due to the presence of errors, detection of small deviations may not be possible and that of larger deviations may be delayed. Similarly, contamination by the embedded error also deteriorates the quality of the gross-error detection and estimation of missing data. Consequently, the performance of PCA may be improved by methods that allow better separation of the errors from the underlying signal.

A popular approach for improving the separation between the errors and the underlying signal is to pretreat the measurements for each variable by an appropriate filter. Linear filters represent the data at a single scale, and suffer from the disadvantages of single-scale PCA. Nonlinear filters are multiscale in nature, and cause less distortion of the retained features, but perform best for piecewise constant or slowly varying signals, and are often restricted to off-line use. The recent development of wavelet-based methods (Donoho et al., 1995) overcomes the disadvantages of other nonlinear filters, and can be used on-line for all types of signals (Nounou and Bakshi, 1998). Despite these advances in filtering methods, preprocessing of the measured variables is still not a good idea, since it usually

destroys the multivariate nature of the process data, which is essential for multivariate SPM and other operation tasks (MacGregor, 1994).

For reaping the benefits of reducing errors by filtering to improve process monitoring, it is necessary to use an integrated approach to both these tasks. For this reason the approach developed by Bakshi (1998) is used who combined the ability of PCA to extract the relationship between the variables and decorrelate the cross-correlation with the ability of wavelets to extract features in the measurements and approximately decorrelate the autocorrelation. This multiscale approach for modeling by PCA can also be generalized to transform other single-scale empirical modeling methods to multiscale modeling. Interesting enough, multiscale modeling has received surprisingly little attention, despite the fact that most existing modeling methods are inherently single scale in nature, whereas most data contain contributions at multiple scales.

The reconstructed signal in the time domain is generated from the large wavelet coefficients and therefore MSPCA integrates the task of monitoring with that of extracting the signal features representing abnormal operation, with minimum distortion and time delay. Consequently, there is no need for a separate step for prefiltering the measured variables (Bakshi, 1998)

7.3. Methodology of MSPCA

The MSPCA methodology consists of decomposing each variable on a selected family of wavelets according to the methods discussed in Chapter 6. The PCA model is then determined independently for the coefficients at each scale. All the scales (approximations and details) are then combined to yield the model for all scales together.

The approach of computing the PCA of the wavelet coefficients instead of the time-domain data, and its application to process monitoring has also been suggested by Kosanovich and Piovoso (1997). Their approach preprocesses the data by the univariate FMH filter and then transforms it to the wavelet domain before applying PCA to the coefficients. This approach does not fully exploit the benefits of multiscale modeling, and the univariate filtering is not integrated with the PCA. Furthermore, monitoring a process based only on its wavelet decomposition will result in too many false alarms after a process returns to normal operation.

MSPCA combines the ability of PCA to extract the cross-correlation or relationship between the variables with that of orthonormal wavelets to separate deterministic features from stochastic processes and approximately decorrelate the autocorrelation among the measurements. The steps in the MSPCA methodology are shown in Figure 6.21 in Chapter 6 and the following algorithm:

1. for each column in the data matrix,
2. compute wavelet decomposition
3. apply level dependent wavelet thresholding
4. end
5. for each scale that contains important information,
6. compute covariance matrix of wavelet coefficients at selected scale
7. compute PCA loadings and scores of wavelet coefficients
8. select appropriate number of loadings
9. end
10. for all scales together, repeat steps 6 to 8
11. reconstruct approximate data matrix from the selected and thresholded scores at each scale
12. end

Steps 11 and 12 only serve to evaluate the linear principal component model and do not serve a purpose in the monitoring scheme. Steps 11 and 12 are also evaluated for the nonlinear principal component model discussed in Chapter 9, which will form part of the monitoring methodology.

To combine the benefits of PCA and wavelets, the measurements for each variable (column) are decomposed to the column's wavelet coefficients using the same orthonormal wavelet for each variable. This results in transformation of the data matrix, \mathbf{X} , into the matrix, \mathbf{WX} , where \mathbf{W} is an $n \times n$ orthonormal wavelet transformation operator containing the filter coefficients,

$$\mathbf{W} = \begin{bmatrix} h_{L,1} & h_{L,2} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & h_{L,N} \\ g_{L,1} & g_{L,2} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & g_{L,N} \\ g_{L-1,1} & \cdot & \cdot & \cdot & g_{L-1,\frac{N}{2}} & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & \cdot & \cdot & \cdot & 0 & g_{L-1,\frac{N}{2}-1} & \cdot & \cdot & \cdot & g_{L-1,N} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ g_{1,1} & g_{1,2} & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & g_{1,N-1} & g_{1,N} \end{bmatrix} = \begin{bmatrix} \mathbf{H}_L \\ \mathbf{G}_L \\ \mathbf{G}_{L-1} \\ \cdot \\ \cdot \\ \mathbf{G}_m \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{G}_1 \end{bmatrix}, \quad (7.1)$$

where \mathbf{G}_m is the $2^{\log_2 n - m} \times n$ matrix containing wavelet filter coefficients corresponding to scale $m = 1, 2, \dots, L$, and \mathbf{H}_L is the matrix of scaling-function filter coefficients at the coarsest scale discussed in Chapter 6. The matrix, \mathbf{WX} , is the same size as the

original matrix, \mathbf{X} , but due to the wavelet decomposition, the deterministic component in each variable in \mathbf{X} is concentrated in a relatively small number of coefficients in \mathbf{WX} , while the stochastic component in each variable is approximately decorrelated in \mathbf{WX} , and is spread over all components according to its power spectrum.

The covariance of the wavelet transformed matrix, and equivalently of the original data matrix, may be written in terms of the contribution at multiple scales as

$$(\mathbf{WX})^T(\mathbf{WX}) = (\mathbf{H}_L\mathbf{X})^T(\mathbf{H}_L\mathbf{X}) + (\mathbf{G}_L\mathbf{X})^T(\mathbf{G}_L\mathbf{X}) + \dots + (\mathbf{G}_m\mathbf{X})^T(\mathbf{G}_m\mathbf{X}) + \dots + (\mathbf{G}_1\mathbf{X})^T(\mathbf{G}_1\mathbf{X}) \quad (7.2)$$

To exploit the multiscale properties of the data, the PCA of the covariance matrix of the coefficients at each scale is computed independently of the other scales. The resulting scores at each scale are not cross-correlated due to PCA, and their autocorrelation is approximately decorrelated due to the wavelet decomposition. Depending on the nature of the application, a smaller subset of the principal-component scores and wavelet coefficients may be selected at each scale. The number of principal components to be retained at each scale are not changed due to the wavelet decomposition since it does not affect the underlying relationship between the variables at any scale. Consequently, existing methods such as cross-validation may be applied to the data matrix in the time domain or to all the wavelet coefficients to select the relevant number of components. This is done to ensure that only those principal components associated with noise are discarded. A cross-validation method for selecting the relevant number of components will be discussed in Section 7.10. Applying separate thresholds at each scale as discussed in Chapter 6 allows MSPCA to be more sensitive to scale-varying signal features such as autocorrelated measurements. Thresholding of the coefficients at each scale identifies the region of the time-frequency space and scales where there is significant contribution from the deterministic features in the signal and also helps in denoising the signal.

The covariance matrix for computing the loadings and scores for all scales together is computed by combining the covariance matrices of all the approximations and details.

This MSPCA modeling method represents one way of using the PCA models at multiple scales, and other approaches may be devised, depending on the application.

Instead of the MSPCA methodology described earlier, some of the benefits of the wavelet representation may be reaped by just transforming the measured data on a selected wavelet basis and computing the PCA of \mathbf{WX} instead of \mathbf{X} . PCA of \mathbf{WX} will make it easier to separate deterministic features in a stochastic process, but this approach will be restricted to off-line use and will not fully exploit the benefits of the multiscale representation,

since it will implicitly assume that the nature of the data does not change with scale. This assumption will cause too many false alarms for autocorrelated measurements, as compared to the MSPCA approach that accounts for the scale-dependent power spectrum. False alarms will also be created for process monitoring based on the scores of \mathbf{WX} after a process returns to normal operation.

7.4. Principle of LPCA

Principal Component Analysis (PCA) is a multivariate technique in which a number of related variables are transformed to (hopefully) a smaller set of uncorrelated variables.

PCA transforms the data matrix in a statistically optimal manner by diagonalizing the covariance matrix by extracting the cross-correlation or relationship between the variables in the data matrix. If the measured variables are linearly related and contaminated by errors, the first few components capture the relationship between the variables, and the remaining components are composed only of the error. Thus, eliminating the less important components reduces the contribution of errors in the measured data and represents it in a compact manner. Applications of PCA rely on its ability to reduce the dimensionality of the data matrix while capturing the underlying variation and relationship between the variables.

7.5. Characteristic roots and vectors

The method of principal components is based on a key result from matrix algebra: A $p \times p$ symmetric, nonsingular matrix, such as the covariance matrix \mathbf{S} , may be reduced to a diagonal matrix \mathbf{L} by premultiplying and postmultiplying it by a particular orthonormal matrix \mathbf{U} such that

$$\mathbf{U}'\mathbf{S}\mathbf{U} = \mathbf{L} \quad (7.3)$$

The diagonal elements of \mathbf{L} , l_1, l_2, \dots, l_p are called the characteristic roots, latent roots or eigenvalues of \mathbf{S} . The columns of \mathbf{U} , $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_p$ are called the characteristic vectors or eigenvectors of \mathbf{S} . The characteristic roots may be obtained from the solution of the following detrimental equation, called the characteristic equation:

$$|\mathbf{S} - \mathbf{I}l| = 0 \quad (7.4)$$

where \mathbf{I} is the identity matrix. This equation produces a p th degree polynomial in l from which the values l_1, l_2, \dots, l_p are obtained.

7.6. The method of principal components

The starting point for PCA is the sample covariance matrix \mathbf{S} (or the correlation matrix). For a p -variable problem,

$$\mathbf{S} = \begin{bmatrix} s_1^2 & s_{12} & \cdots & s_{1p} \\ s_{12} & s_2^2 & \cdots & s_{2p} \\ \vdots & \vdots & & \vdots \\ s_{1p} & s_{2p} & \cdots & s_p^2 \end{bmatrix} \quad (7.5)$$

Where s_i^2 is the variance of the i th variable, x_i , and s_{ij} is the covariance between the i th and j th variables. If the covariances are not equal to zero, it indicates that a linear relationship exists between these two variables, the strength of that relationship being represented by the correlation coefficient, $r_{ij} = s_{ij} / (s_i s_j)$.

The principal component transformation will transform the p correlated variables x_1, x_2, \dots, x_p into p new uncorrelated variables z_1, z_2, \dots, z_p . The coordinate axes of these new variables are described by the characteristic vector \mathbf{u}_i which make up the matrix \mathbf{U} of the direction cosines used in the transformation:

$$\mathbf{z} = \mathbf{U}'[\mathbf{x} - \bar{\mathbf{x}}] \quad (7.6)$$

Here \mathbf{x} and $\bar{\mathbf{x}}$ are $p \times 1$ vectors of observations on the original variables and their means.

The transformed variables are called the principal components of \mathbf{x} or pc's for short. The i th principal component is

$$z_i = \mathbf{u}_i'[\mathbf{x} - \bar{\mathbf{x}}] \quad (7.5)$$

and will have mean zero and variance l , the i th characteristic root. To distinguish between the transformed variables and the transformed observations, the transformed variables will be called principal components and the individual transformed observations will be called z-scores. The distinction is made here with regard to z-scores because another normalization of these scores exists.

7.7. Some properties of principal components

7.7.1. TRANSFORMATIONS

If one wishes to transform a set of variables \mathbf{x} by a linear transformation $\mathbf{z}_i = \mathbf{u}_i'[\mathbf{x} - \bar{\mathbf{x}}]$ whether \mathbf{U} is orthonormal or not, the covariance matrix of the new variables, \mathbf{S}_z , can be determined directly from the covariance matrix of the original observations, \mathbf{S} , by the relationship

$$\mathbf{S}_z = \mathbf{U}'\mathbf{S}\mathbf{U} \quad (7.8)$$

However, the fact that \mathbf{U} is orthonormal is not a sufficient condition for the transformed variables to be uncorrelated. Only this characteristic vector solution will produce an \mathbf{S}_z that is a diagonal matrix like \mathbf{L} producing new variables that are uncorrelated.

7.7.2. INTERPRETATION OF PRINCIPAL COMPONENTS

The interpretation of principal components will be explained by means of an example. For the purpose of illustration, linear principal component analysis was applied to the first two variables from the industrial data. The coefficients of the first vector were found to be 0.7236 and 0.6902. As observed, they are nearly equal and both positive, indicating that the first pc, z_1 , is a weighted average of both variables. This is related to variability that x_1 and x_2 have in common; in the absence of correlated errors of measurement, this would be assumed to represent process variability. The coefficients of the second vector were -0.6902 and 0.7236. They are also nearly equal except for sign indicating that the second pc, z_2 , represent differences in the measurements for the two variables that would probably represent testing and measurement variability.

7.7.3. GENERALIZED MEASURES AND COMPONENTS OF VARIABILITY

In keeping with the goal of multivariate analysis of summarizing results with as few numbers as possible, there are two single-number quantities for measuring the overall variability of a set of multivariate data. These are

1. The determinant of the covariance matrix, $|\mathbf{S}|$. This is called the generalized variance. The square root of this quantity is proportional to the area or volume generated by a set of data.
2. The sum of the variances of the variables:

$$s_1^2 + s_2^2 + \dots + s_p^2 = \text{Tr}(\mathbf{S}) \quad (\text{trace of } \mathbf{S}) \quad (7.9)$$

Conceivably, there are other measures of generalized variability that may have certain desirable properties but these two are the ones that have found general acceptance among practitioners.

A useful property of PCA is that the sum of the original variances is equal to the sum of the characteristic roots. For the two variable case,

$$\begin{aligned} s_1^2 + s_2^2 &= 0.7986 + 0.7343 = 1.5329 \\ &= 1.4465 + 0.0864 = l_1 + l_2 \end{aligned}$$

This identity is particularly useful because it shows that the characteristic roots, which are the variances of the principal components, may be treated as variance components. The ratio of each characteristic root to the total will indicate the proportion of the total variability accounted for by each pc. For z_1 , $1.4456/1.5329 = 0.944$ and for z_2 , $0.0864/1.5329 = 0.056$. This says that roughly 94% of the total variability of these data (as represented by $\text{Tr}(\mathbf{S})$) is associated with, accounted for or “explained by” the variability of the process and 6% due to the variability related to testing and measurement. Since the characteristic roots are sample estimates, these proportions are also sample estimates.

7.5.4. CORRELATION OF PRINCIPAL COMPONENTS AND ORIGINAL VARIABLES

It is also possible to determine the correlation of each pc with each of the original variables, which may be useful for diagnostic purposes. The correlation of the i th pc, z_i , and the j th original variable, x_j , is equal to

$$r_{zx} = \frac{u_{ji} \sqrt{l_i}}{s_j} \quad (7.10)$$

For instance, the correlation between z_1 and x_1 is

$$\frac{u_{11} \sqrt{l_1}}{s_1} = \frac{0.7236 \sqrt{1.4465}}{\sqrt{0.7986}} = 0.974$$

The first pc is more highly correlated with the original variables than the second. This is to be expected because the first pc accounts for more variability than the second. Note that the sum of squares of each row is equal to 1.0.

7.8. Scaling of Data

7.8.1. INTRODUCTION

There are two ways of scaling principal components, one by rescaling the original variables, which will be discussed here, and the other by rescaling the characteristic vectors, which won't be discussed here. There is no significant reason for choosing the one method above the other and in the end remains a personal choice. To me, scaling the original variables made more sense.

The results obtained by scaling the original data will depend on the method employed. Once the method of scaling is selected, the PCA operations will proceed, for the most part, as described earlier but there will be some modifications unique to each method. The main effect of this choice will be on the matrix from which the characteristic vectors are obtained.

Specifically, the following three methods will be considered:

1. No scaling at all. The final variate vector is \mathbf{x} .
2. Scaling the data such that each variable has zero mean (i.e., in terms of deviation from the mean). The final variate vector is $\mathbf{x} - \bar{\mathbf{x}}$.
3. Scaling the data such that each variable is in standard units. (i.e., has zero mean and unit standard deviation). Each variable is expressed as $(x_i - \bar{x}_i) / s_i$.

As stated above, the choice of scale will determine the dispersion matrix used to obtain the characteristic vectors. If no scaling is employed, the resultant matrix will be the product or second moment matrix; if the mean is subtracted, it will be the covariance matrix; if the data are in standard units, it will be a correlation matrix.

7.8.2. DATA AS DEVIATIONS FROM THE MEAN: COVARIANCE MATRICES

Here it is not necessary to actually subtract the variable means from the data; the operations required to obtain the covariance matrix will take care of it.

If one were to subtract the means from the data and use these deviations as a data set, say $\mathbf{x}_d = \mathbf{x} - \bar{\mathbf{x}}$ with the resulting $n \times p$ data matrix \mathbf{X}_d , the covariance matrix would be $\mathbf{X}_d' \mathbf{X}_d / (n - 1)$. The characteristic vectors \mathbf{U} , \mathbf{V} and \mathbf{W} would stay the same. For large problems in terms of sample size or large number of digits for the original data, this option may be preferable, numerically, to obtaining \mathbf{S} from the raw data directly.

There are many occasions when one cannot use the covariance matrix. There are two reasons for this:

1. The original variables are in different units. In this case, the operations involving the trace of the covariance matrix have no meaning. For instance, if a variable is expressed in centimeters, its variance is 100 times what it would be if it were expressed in millimeters (variance of variable in cm divided by variance of variable in mm). The variable would now exert considerable more influence on the shaping of the pc's since PCA is concerned with explaining variability. When the units are different, the solution is to make the variances the same (i.e., use standard units), which makes the covariance matrix into a correlation matrix.
2. Even if the original variables are in the same units, the variances may differ widely, often because they are related to their means. If this gives undue weight to certain variables, the correlation matrix should be employed here also (unless, possibly, taking logs of the variables or the use of some other variance-stabilizing transformation will suffice).

Nevertheless, when the variables are in the same units and do have the same amount of variability, there are some advantages in using covariance matrices. This is particularly true in physical applications where PCA is used in building physical models. Using the covariance matrix should also help with diagnostics since the V -vectors are in the original units of the variables.

It is important to note that there is no one-to-one correspondence between the pc's obtained from a correlation matrix and those obtained from a covariance matrix. The more heterogeneous variances are, the larger the difference will be between the two sets of vectors. If the covariance matrix has $(p - k)$ zero roots, then the correlation matrix will also have $(p - k)$ zero roots. However, if the covariance matrix has $(p - k)$ equal roots, the correlation matrix will not necessarily have the same number.

7.9. Using Principal Components in Quality Control

7.9.1. TYPE I ERRORS

When one uses two or more control charts (i.e. time-series plot of the squared prediction error, SPE) simultaneously, some problems arise with the type I error. This is the probability of a sample result being outside the control limits when the process is at the mean or the standard established for that process. If one would consider first the two control charts for x_1 and x_2 which is variable one and variable two of the training

data (see Paragraph 3.11 of Chapter 3), the probability that each of them will be in control if the process is on standard is 0.95. If these two variables were uncorrelated (which they are not in this case), the probability that both of them would be in control is $0.95^2 = 0.9025$ so the effective Type I error is roughly $\alpha = 0.10$, not 0.05. For 8 uncorrelated variables, the Type 1 error would be $1 - (0.95^8) = 0.37$. Thus if one was attempting to control 8 independent variables, at least one or more of these variables would indicate an out-of-control condition over one-third of the time.

The problem becomes more complicated when the variables are correlated as they are here. If they were perfectly correlated, the Type I error would remain 0.05. However, anything less than that, such as the present case, would leave one with some involved computations to find out what the Type I error really was. The use of principal component control charts resolves some of this problem because the pc's are uncorrelated; hence, the Type I error may be computed directly. This may still leave one with a sinking feeling about looking for trouble that does not exist.

7.9.2. GOALS OF MULTIVARIATE QUALITY CONTROL

Any multivariate quality control procedure, whether or not PCA is employed, should fulfill four conditions.

1. A single answer should be available to answer the question: "Is the process in control?"
2. An overall Type I error should be specified.
3. The procedure should take into account the relationships among the variables.
4. Procedures should be available to answer the question: "If the process is out-of-control, what is the problem?"

Condition 4 is much more difficult than the other three, particularly as the number of variables increases since it needs expert information which is more than just the mathematical or statistical information needed in the first three conditions. There is usually no easy way to this, although the use of PCA may help.

7.10. Selecting the number of principal components

7.10.1. INTRODUCTION

One of the greatest uses of PCA is its potential ability to adequately represent a p -variable data set in $k < p$ dimensions. The question becomes: "What is k ?"

Obviously, the larger k is, the better the fit of the PCA model; the smaller k is, the more simple the model will be. Somewhere, there is an optimal value of k ; what is it? To determine k , there must be a criterion for optimality.

One method for determining the optimum number of pc's is the cross-validation approach by Wold (1976, 1978), Eastment and Krzanowski (1982) and Krzanowski (1983, 1987). This approach is recommended when the initial intention of a study is to construct a PCA model with which future sets of data will be evaluated as in this case.

In PCA it consists of randomly dividing the sample into g groups. The first group is deleted from the sample and a PCA is performed on the remaining sample. The vectors obtained from that reduced sample are used to obtain pc's and Q-statistics (explained in more detail in Chapter 10) for the deleted group. That group is returned to the sample, the next group is deleted, and the procedure is repeated g times. The grand average of the Q-statistic, divided by p , is called the PRESS-statistic (PREdiction Sum of Squares). Its primary use in PCA is as a stopping rule. It differs from other stopping rules in that it is based on the Q-statistic rather than the characteristic roots. Krzanowski pointed out that it is possible to have different data sets produce the same covariance or correlation matrix but would probably produce different PRESS-statistics although their characteristic roots would be the same. Cross-validation also differs from other stopping rules in requiring the original data while other procedures work directly from the covariance or correlation matrix. Although the procedure described here is not a significance test, it is more quantitative than most other stopping rules.

7.10.2. A SIMPLE CROSS-VALIDATION PROCEDURE

The principle of cross-validation as a stopping rule will be illustrated using dataset 1 (refer to Chapter 6 paragraph 6.9.2.4.), where $p = 8$ and $n = 645$. The transformed data matrix will be denoted by the 645×8 matrix \mathbf{X} . For this case the data set will be divided into $g = 5$ groups of 20 observations each so that the first group will be observations 1-20, the second group 21-40, and so on. The procedure is as follows:

1. Delete the first group from the sample. Perform a PCA on the remaining observations (i.e., 21-645). Obtain all eight vectors. This example used a correlation matrix, the data will be in standard units.
2. For the deleted sample, obtain all eight z-scores for each observation using the vectors obtained in step 1.

3. Using, in turn, the first pc, the first two pc's, and so on, obtain the predicted values of the deleted sample. \bar{x} will be equal to zero.
4. For each observation in the deleted sample, obtain Q . For the first observation, $Q = 1.054$ for one pc, $Q = 0.639$ for two pc's, and so on.
5. Return the deleted group to the sample and remove the second group. Repeat steps 1-4. Do the same for the other three groups. This concluded, there will now be 645 values of Q for a one-pc model, another 645 for a two-pc model, and so on.
6. For each pc model, add up the 645 Q -statistics and divide each sum by $np = 5160$. These are called PRESS-statistics and be designated by PRESS(1), PRESS(2), and so on. It will also be necessary to obtain PRESS(0), the sum of squares of the original data, again assuming a mean of zero.
7. To determine whether the addition of another pc, say the k th pc, to the model is warranted, form the statistic

$$W = \frac{[PRESS(k-1) - PRESS(k)] / D_M}{PRESS(k) / D_R} \quad (7.11)$$

where

$$D_M = n + p - 2k \quad (7.12)$$

$$D_R = p(n-1) - \sum_{i=1}^k (n + p - 2i) \quad (7.13)$$

If $W > 1$, then retain the k th pc in the model and test the $(k+1)$ st. for example, to test whether the first pc should be included, one would form

$$W = \frac{[PRESS(0) - PRESS(1)] / 651}{PRESS(1) / 4501} = \frac{0.0077}{0.00066303} = 11.62$$

and the first pc would be included in the model. For the process data the process terminated with the inclusion of the third principal component.

In practice, if one had a large number of variables and was confident that only a small number of pc's would be retained, a different strategy might be employed, in which each characteristic vector is obtained and tested sequentially before obtaining the next and, in that way, only one unwanted vector is obtained.

Table 7.1. PRESS values for selecting the number of pc's to retain

k	$PRESS(k)$	D_M	D_R	W
0	8.0000			
1	2.9843	651	4501	11.6200
2	2.2097	649	3852	2.0806
3	1.1594	647	3205	4.4873
4	0.9813	645	2560	0.7204 < 1
5	0.9634	643	1917	0.0555
6	0.8834	641	1276	0.1804
7	0.8811	639	637	0.0025

It is possible that if one were to continue this process beyond the first occurrence where $W < 1$, later values of k might produce one or more occurrences of $W > 1$. This may be due to the presence of outliers.

7.10.3. ENHANCEMENTS

In the previous section \bar{x} was assumed to be zero since it was equal to zero for the entire example. However, it may be that the mean is not equal to zero and the cross-validation technique for it is more complicated, involving the deletion of variables. Furthermore, both \mathbf{U} and \mathbf{z} are considered estimates and, as we now know, may be estimated simultaneously using singular value decomposition. This, of course, is not possible here because the \mathbf{z} -scores are obtained for the observations not included in the sample from which \mathbf{U} is obtained. The solution to this problem is to use all n observations in each subsample but randomly delete elements from each data vector. The good way to do this is to randomly order the observations and use a cyclic deletion pattern given by Wold (1987). The estimation procedure will, of necessity, require SVD but the SVD algorithm employed must be able to handle missing data.

7.11. Application

7.11.1. SOFTWARE SETUP

Figure 7.1. displays the LPCA interface which can be used to apply LPCA to the data. This interface can either be displayed from the main interface or by using the Next button from the wavelet analysis interface.

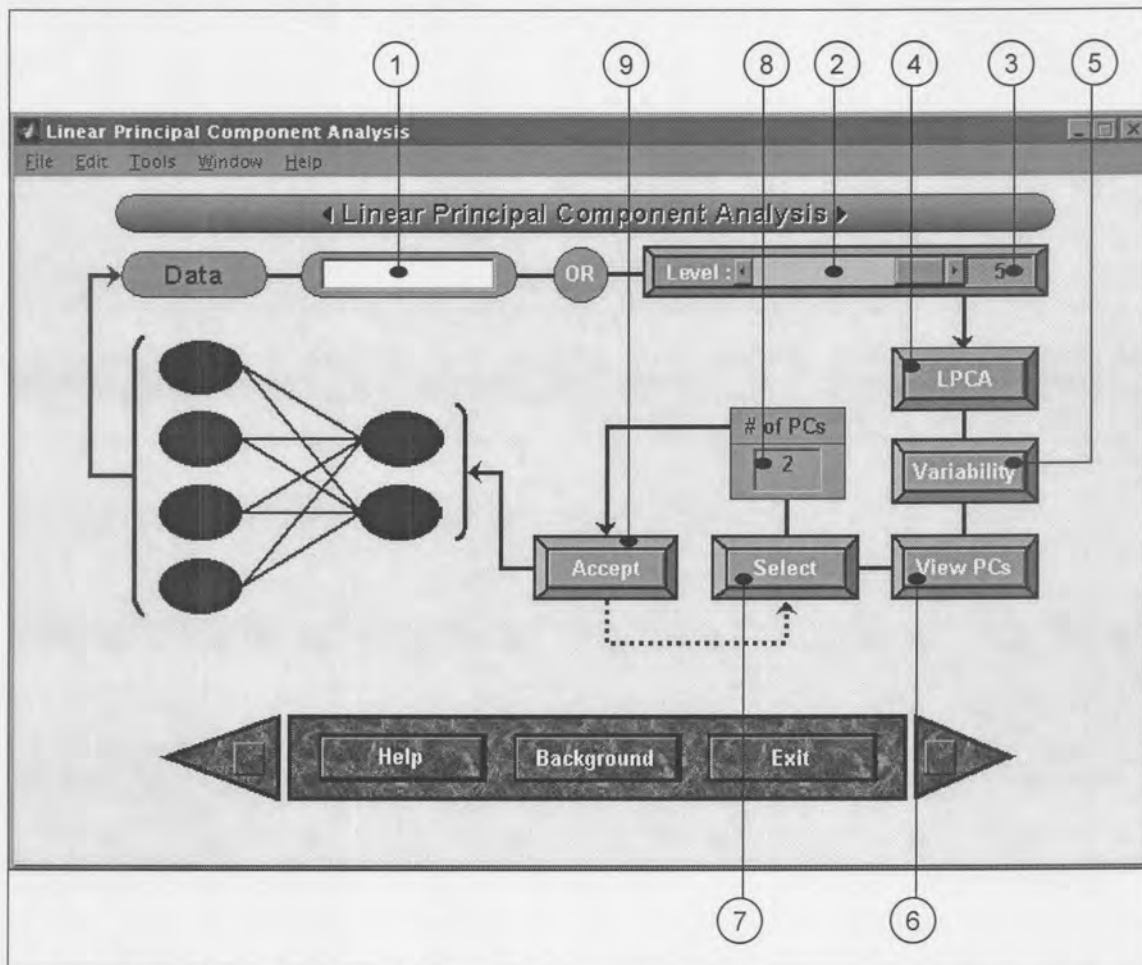


Figure 7.1. Linear PCA main interface

Figure 7.1 Tags:

1. Name of the variable containing the data to which linear PCA needs to be applied. If this space is left blank the default variable from the database will be used. The variable must contain more than one column of data.
2. Level selection slider. This slider is used if the default data from the database is used. Since LPCA is applied to each level separately, the level to which LPCA needs to be applied can be selected using this slider. It will automatically detect the number of levels contained in the database.

3. Level number display.
4. LPCA application button. Using this button will apply PCA to the named variable or the specified level in the database.
5. Variability interface used to view the variability of the principal components. This button will open Figure 7.2.
6. Principal component viewer interface used to view each principal component separately. This button will open Figure 7.3.
7. Number of principal components selection interface. This button will open Figure 7.4.
8. Specify the final number of principal components to select. This choice is based on the results obtained from the principal components selection interface in Figure 7.4.
9. Accept the number of principal components specified in 8. This reduces the number of principal components to the number specified and saves the results to the database.

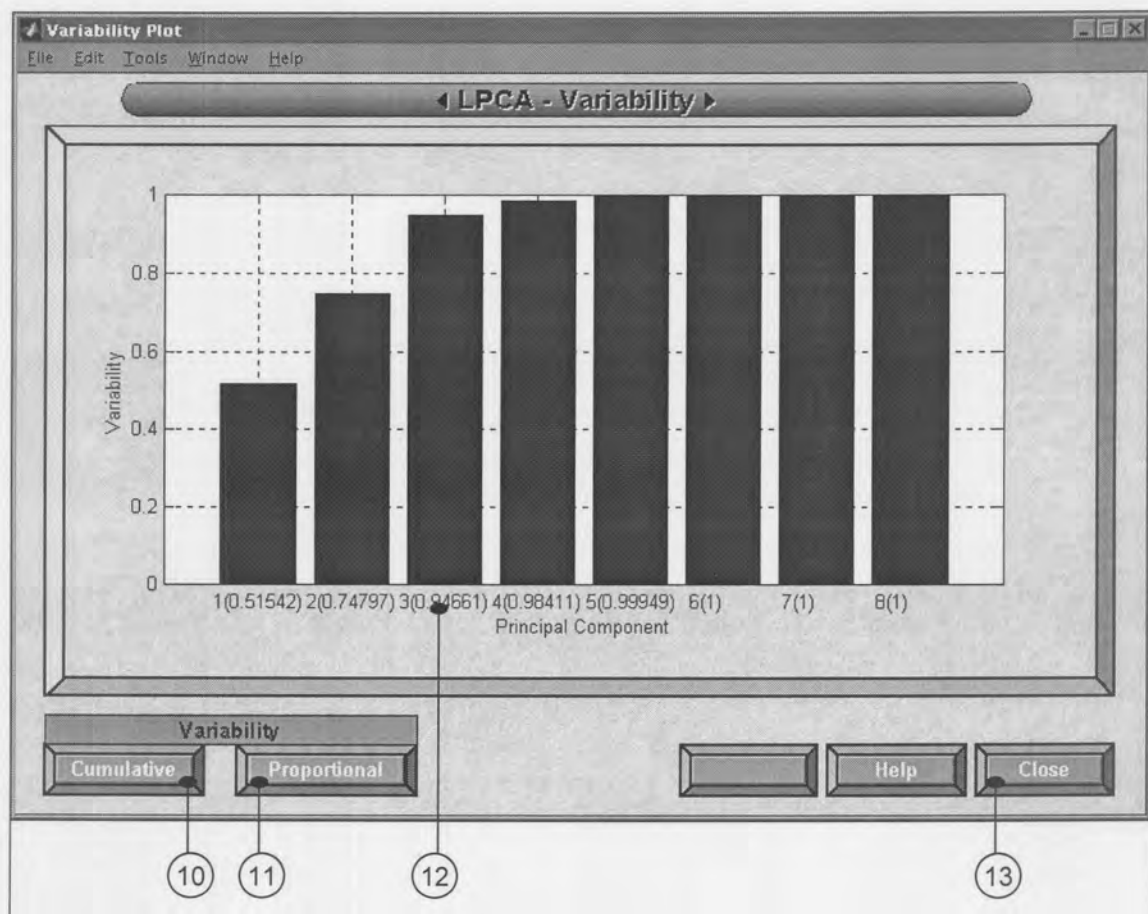


Figure 7.2. Cumulative variability plot

Figure 7.2 Tags:

10. Cumulative variability. This button displays Figure 7.2.
11. Proportional variability. This button displays Figure 7.3.
12. Display of the individual contribution of each variable to the total variability.
13. Close the current interface and return to the LPCA interface.

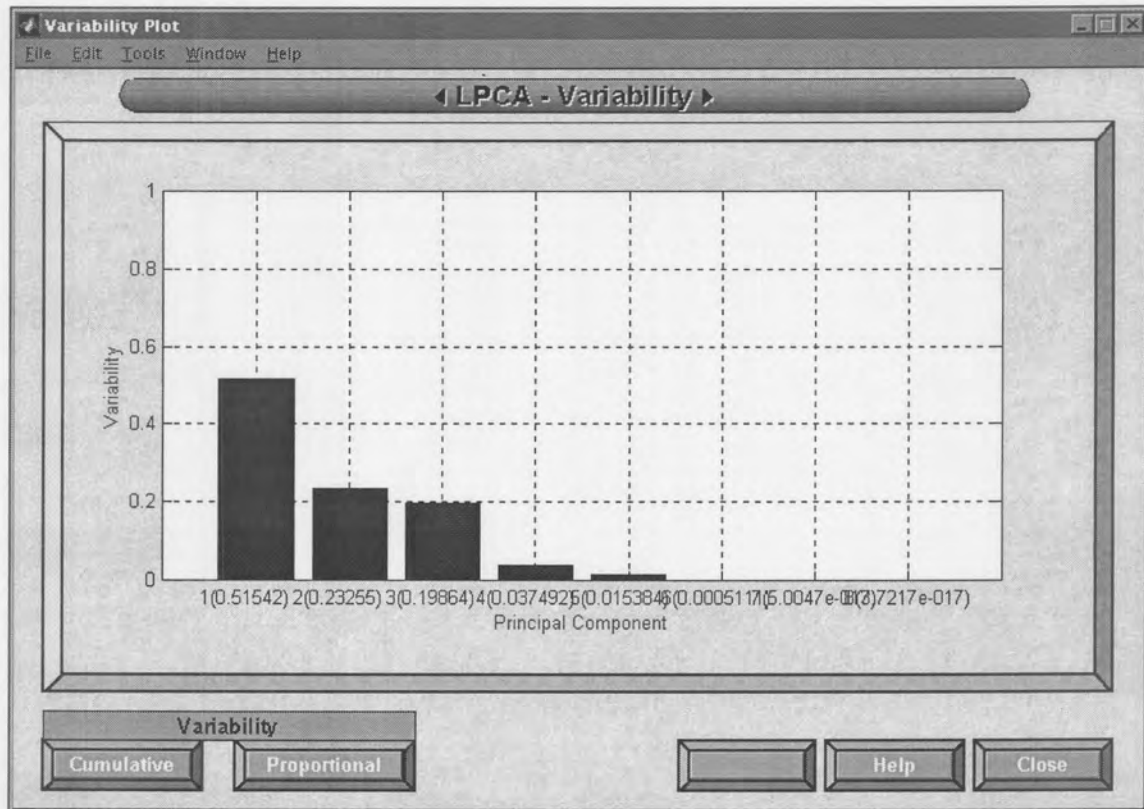


Figure 7.3. Proportional variability plot

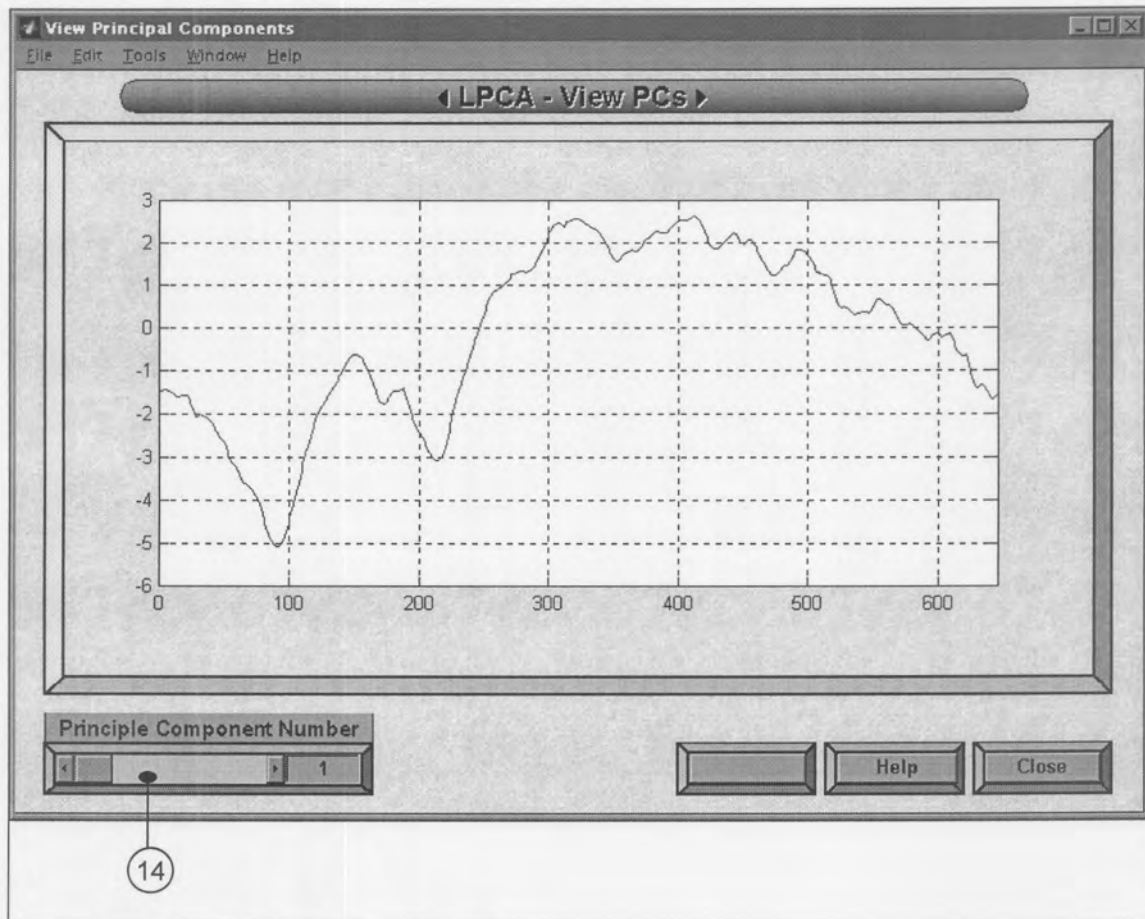


Figure 7.4. Principal Component plot

Figure 7.4 Tags:

14. Principal component number to display.

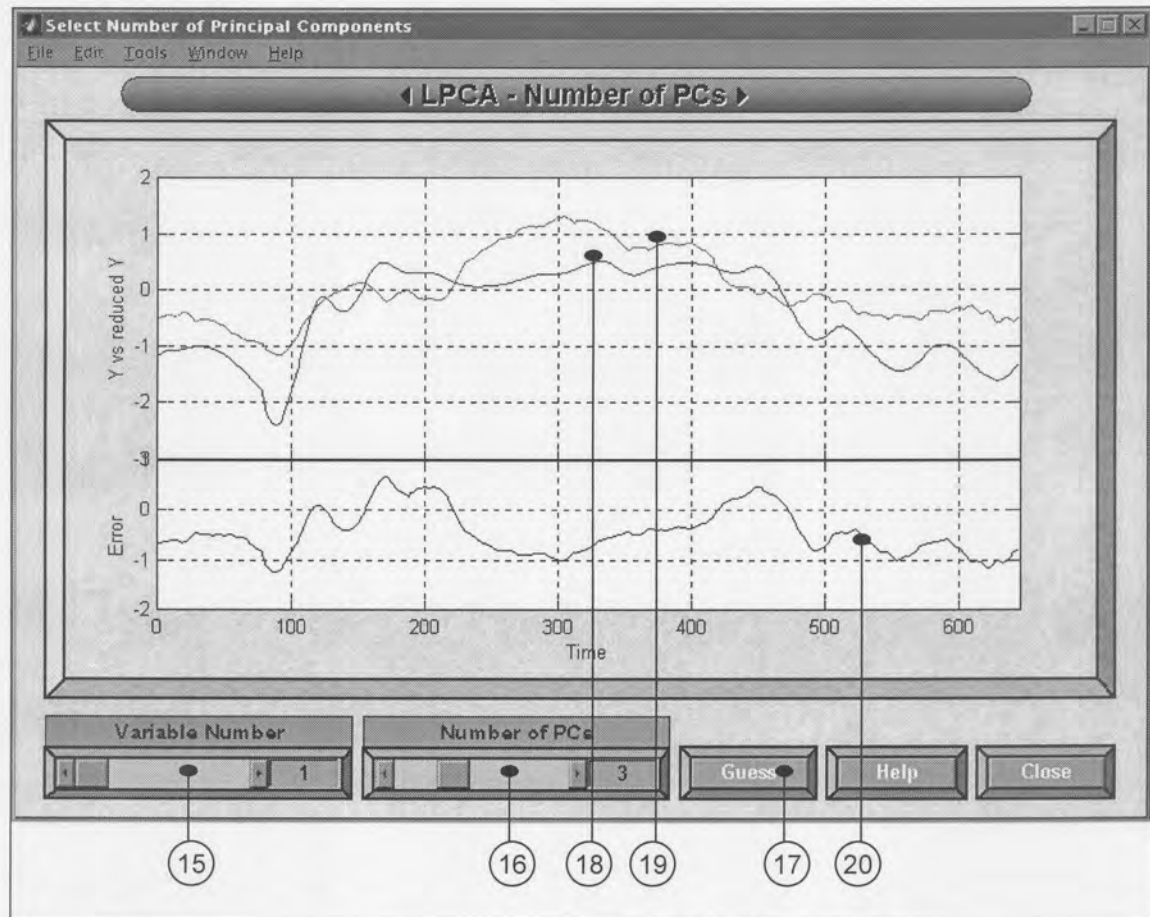


Figure 7.5. Principal component selection interface

Figure 7.5 Tags:

15. Select the original variable number from the current level to display.
16. Reconstruct the original variable in 15 using the number of principal components specified here.
17. As an option you can choose the number of principal components to retain using the Krzanowski cross-validation method based on PRESS values discussed in Section 7.8.
18. Original variable
19. Reconstruction of the original variable
20. Error plot between the original and reconstructed variable using the number of principal components specified.

7.11.2. EXPERIMENTAL

Once the data matrix of the combined non-thresholded details and approximations (dataset 2) and matrix of approximations (dataset 1) from Chapter 6 had been obtained, the next step was to remove any data points which did not correspond to nominal process operation. By visual inspection ten coefficients from dataset 1 and 30 from dataset 2 (refer to Chapter 6 paragraph 6.9.2.4.) were identified as not being representative of normal operation. Dataset 2 contained 40 columns and Dataset 1 eight columns. Linear PCA was applied to the resultant data sets. The cross-validation method using Krzanowski's PRESS-statistic (Krzanowski, 1987) was used to select the appropriate number of principal components. This technique indicated that three and four linear principal components were adequate to explain the underlying variability in dataset 1 and dataset 2 respectively. For dataset 2 this meant four out of a possible 40 and for dataset 1, three out of a possible eight, indicating a high degree of correlation. A total of 94.47% and 93.5% of the total explained variance was captured by the two sets of principal components respectively.

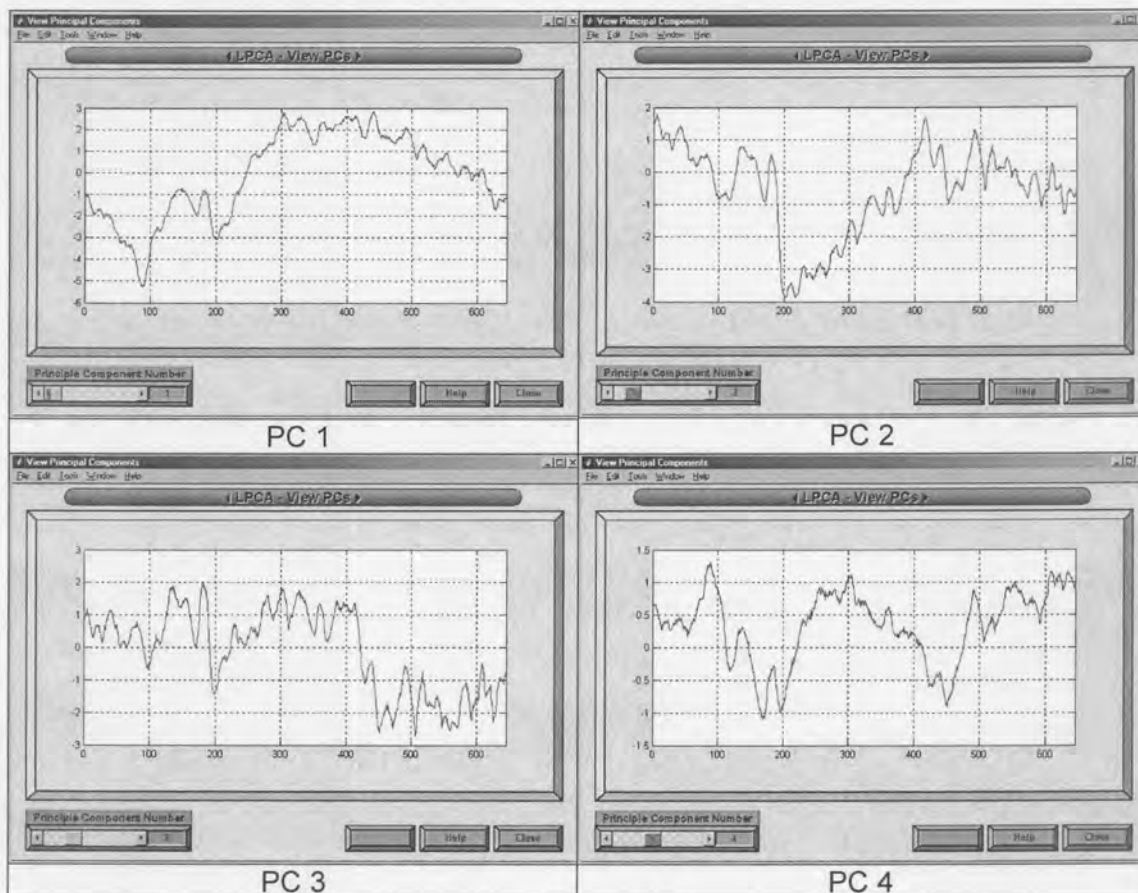


Figure 7.6. First four principal components of dataset 1.

Figure 7.6 gives the first four linear principal components of dataset 1 of which the first three were retained. Table 7.2 gives a summary of the cumulative variability of the eight and first eight principal components of dataset 1 and dataset 2 respectively.

Table 7.2. Cumulative variability for pc's of dataset 1 and dataset 2

# Principal Components	Cumulative Variability For Dataset 1	Cumulative Variability For Dataset 2
1	0.50957	0.48375
2	0.73978	0.70285
3	0.94467	0.89768
4	0.98321	0.93503
5	0.99796	0.96529
6	0.99934	0.97937
7	0.99994	0.98790
8	1	0.99089

8.1. Introduction

Many industrial processes such as the case under consideration exhibit significant nonlinear behavior. In these cases the application of PCA is not strictly appropriate. A non-linear PCA methodology is proposed to take account of the non-linearities inherent within the process data.

The non-linear PCA method proposed in this study is based upon the input-training neural network approach (Tan and Mavrovouniotis, 1995). The advantage of this approach is that it enables both the second-order and higher-order correlations to be extracted separately. This is achieved by first applying linear PCA as discussed in Chapter 7 to compress the data prior to implementing non-linear compression. By adopting this procedure, a more parsimonious description of process behaviour is achieved. The methodology is investigated for non-linear process performance monitoring in Chapter 9.

In chapter 7 the concept of multiscale linear principal component analysis (MSLPCA) was introduced as means of dimensionality reduction. In this chapter it will be extended to nonlinear principal component analysis (NLPCA) and in chapter 9 this will be combined with the LPCA from chapter 7 and multiscaling from chapter 6 to form multiscale nonlinear principal component analysis (MSNLPCA). Since NLPCA uses a type of neural network referred to as an input training neural network this concept will first be introduced separately before being applied to MSNLPCA.

Apart from input training neural networks, dimensionality reduction can also be performed by autoassociative neural networks, which are feedforward neural nets trained to perform the identity mapping between network inputs and outputs. Although IT-nets are an improvement over autoassociative neural networks (AANN), AANN's will be briefly discussed in order to realize the similarities and differences between the two.

Figure 8.1 gives a schematic of an AANN. With AANN's dimensionality reduction is achieved through a bottleneck, that is a hidden layer with a small number of nodes. Most previous work focused on single-hidden-layer networks (Ackley et al., 1985; Cottrel et al., 1987; Abbas and Fahmy, 1993). Kramer (1991) pointed out that the single-hidden-layer architecture was unable to model nonlinear relationships between observed variables and latent variables, and consequently offered no significant improvement over conventional PCA. He then established a three-hidden-layer architecture for autoassociative networks to capture nonlinear correlations. The three-hidden-layer autoassociative networks can be used to perform various data screening

tasks, such as data noise filtering, missing measurement replacement, and gross error detection and correction (Kramer, 1992).

An autoassociative network is composed of a mapping subnet and a demapping subnet, each of which is a single-hidden-layer network by itself. Dong and McAvoy (1993) proposed to train the two subnets separately. The method they proposed involves three steps:

- (1) find principal curves by successively applying the algorithm of Hastie and Stuetzle (1989) to observed data and residuals;
- (2) train a network that maps the original data to principal curves;
- (3) train another network that maps principal curves to the original data.

Autoassociative networks are typically trained through backpropagation. In general, the performance of backpropagation deteriorates as the number of hidden layers gets larger (Hertz et al., 1991). The poor performance of backpropagation in training the mapping subnet is attributable to the large number of layers. In the process of backpropagation learning, modifications of weights are based on errors propagated backward from the output layer. After several layers of error propagation, the searching direction for the weights in the mapping subnet may deviate from the direction that minimizes the output error function. This effect becomes even more pronounced for an autoassociative network due to its bottleneck layer.

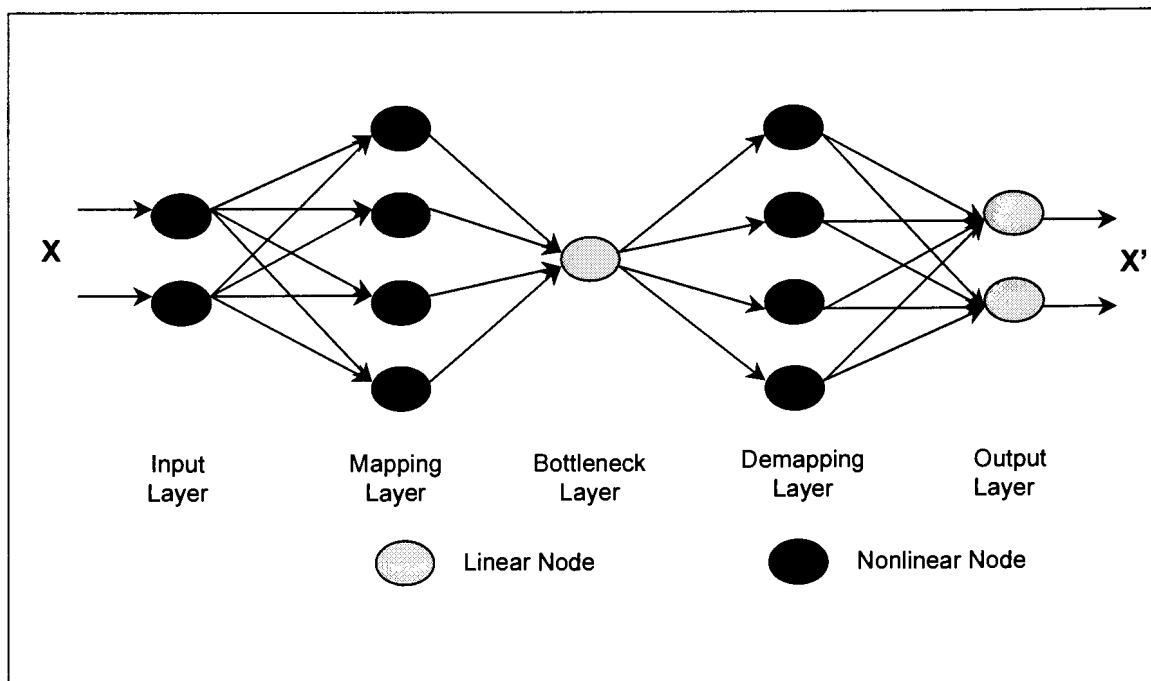


Figure 8.1. A 2-4-1-4-2 autoassociative network

Thus, as an alternative, the new method proposed by Tan and Mavrovouniotis (1995) was used. In their work they also used neural networks as nonlinear models for observed variables and latent variables and additionally used a concept called input training (IT). With this method, only one single-hidden-layer network is needed for dimensionality reduction of a given data set. The method proposed by them however, uses backpropagation to train the network, which is not an optimized training method and tends to take long to converge to the performance goal. In order to overcome this, this work extends the backpropagation training algorithm and combines it with the Levenberg-Marquardt (LM) training algorithm in an effort to facilitate enhanced speed and better convergence. This new enhanced training algorithm forms a significant contribution to the process of NLPCA.

In section 8.2. a background to Backpropagation is given and in section 8.3. background is provided to the Levenberg-Marquardt training algorithm. Section 8.4 explains the concept of input training. Section 8.5. extends section 8.3 and 8.4 to develop an enhanced training algorithm for input training neural networks (IT-nets).

8.2. The Backpropagation algorithm

8.2.1. GENERAL BACKPROPAGATION

There are many variations of the backpropagation algorithm. The simplest implementation of backpropagation learning updates the network weights and biases in the direction in which the performance function decreases most rapidly - the negative of the gradient.

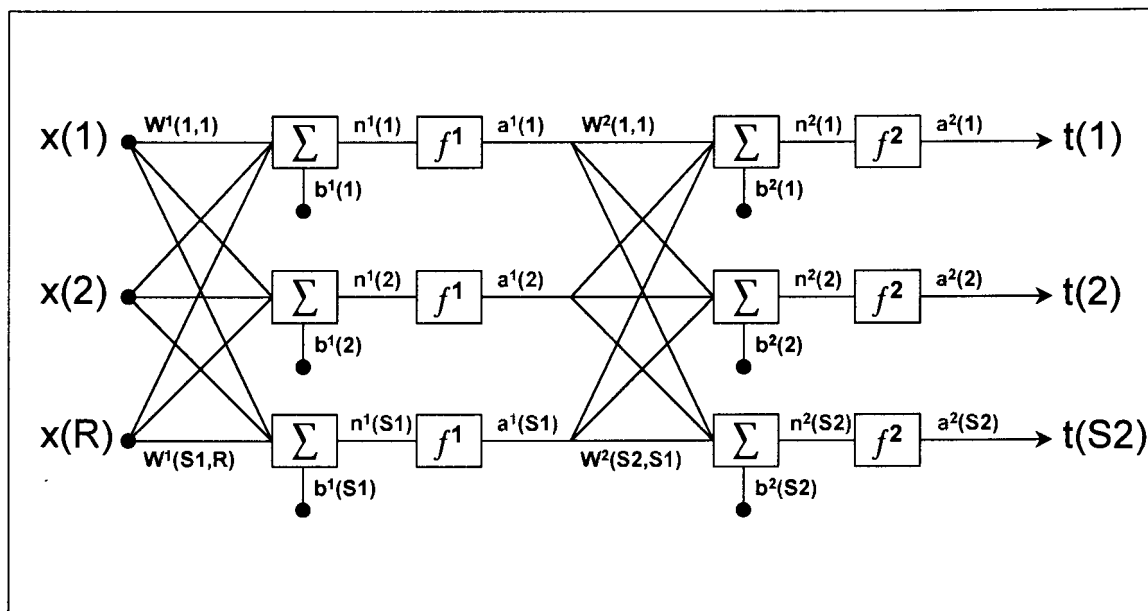


Figure 8.2. two-layer feedforward network

Consider a multilayer feedforward network, such as the two-layer network of Figure 8.2. The net input to unit i in layer $k + 1$ is

$$n^{k+1}(i) = \sum_{j=1}^{S_k} w^{k+1}(i, j)a^k(j) + b^{k+1}(i) \quad (8.1)$$

The output of unit i will be

$$a^{k+1}(i) = f^{k+1}(n^{k+1}(i)) \quad (8.2)$$

For an M layer network the system equations in matrix form are given by

$$\mathbf{a}^0 = \mathbf{x} \quad (8.3)$$

$$\mathbf{a}^{k+1} = \mathbf{f}^{k+1}(W^{k+1}\mathbf{a}^{k+1} + \mathbf{b}^{k+1}) \quad k = 0, 1, \dots, M - 1 \quad (8.4)$$

The task of the network is to learn associations between a specific set of input-output pairs $\{(x_1, t_1), (x_2, t_2), \dots, (x_Q, t_Q)\}$.

The performance index for the network is

$$E = \frac{1}{2} \sum_{q=1}^Q (\mathbf{t}_q - \mathbf{a}_q^M)^T (\mathbf{t}_q - \mathbf{a}_q^M) = \frac{1}{2} \sum_{q=1}^Q \mathbf{e}_q^T \mathbf{e}_q \quad (8.5)$$

where \mathbf{a}_q^M is the output of the network when the q th input, \mathbf{x}_q , is presented and \mathbf{e}_q is the error for the q th input. For standard backpropagation algorithm, as in this case, we use an approximate steepest descent rule. The performance index is approximated by

$$\hat{E} = \frac{1}{2} \mathbf{e}_q^T \mathbf{e}_q \quad (8.6)$$

where the total sum of squares is replaced by the squared errors for a single input/output pair. The approximate steepest (gradient) descent algorithm is then

$$\Delta w^k(i, j) = -\alpha \frac{\partial \hat{E}}{\partial w^k(i, j)} \quad (8.7)$$

$$\Delta b^k(i) = -\alpha \frac{\partial \hat{E}}{\partial b^k(i)} \quad (8.8)$$

where α is the learning rate. Define

$$\delta^k(i) \equiv \frac{\partial \hat{E}}{\partial n^k(i)} \quad (8.9)$$

as the sensitivity of the performance index to changes in the net input of unit i in layer k . Now it can be shown, using (1), (6), and (9), that

$$\frac{\partial \hat{E}}{\partial w^k(i, j)} = \frac{\partial \hat{E}}{\partial n^k(i)} \frac{\partial n^k(i)}{\partial w^k(i, j)} = \delta^k(i) a^{k-1}(j) \quad (8.10)$$

$$\frac{\partial \hat{E}}{\partial b^k(i)} = \frac{\partial \hat{E}}{\partial n^k(i)} \frac{\partial n^k(i)}{\partial b^k(i)} = \delta^k(i) \quad (8.11)$$

It can also be shown that the sensitivities satisfy the following recurrence relation

$$\delta^k = \dot{F}^k(\mathbf{n}^k) W^{k+1T} \delta^{k+1} \quad (8.12)$$

where

$$\dot{F}^k(\mathbf{n}^k) = \begin{bmatrix} \dot{f}^k(n^k(1)) & 0 & \dots & 0 \\ 0 & \dot{f}^k(n^k(2)) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \dot{f}^k(n^k(Sk)) \end{bmatrix} \quad (8.13)$$

and

$$\dot{f}^k(n) = \frac{df^k(n)}{dn} \quad (8.14)$$

The recurrence relation is initialized at the final layer

$$\delta^M = -F^M(\mathbf{n}^M)(\mathbf{t}_q - \mathbf{a}_q) \quad (8.15)$$

The overall learning algorithm now proceeds as follows:

1. Propagate the input forward using equations 8.3 and 8.4;
2. Propagate the sensitivities back using equations 8.15 and 8.12;
3. Update the weights and offsets using equations 8.7, 8.8, 8.10, and 8.11.

There are two different ways in which this gradient descent algorithm can be implemented: incremental mode and batch mode. In the batch mode which is applied here, all of the inputs are applied to the network before the weights are updated.

The training function has one learning parameter associated with it - the learning rate α shown in equations 8.7 and 8.8. With standard steepest descent as applied here, the learning rate is held constant throughout training. The performance of the algorithm is very sensitive to the proper setting of the learning rate. The larger the learning rate, the bigger the step. If the learning rate is made too large the algorithm may oscillate and become unstable. If the learning rate is set too small, the algorithm will take a long time to converge. It is not practical to determine the optimal setting for the learning rate before training, and, in fact, the optimal learning rate changes during the training process, as the algorithm moves across the performance surface.

8.2.2. BACKPROPAGATION APPLIED TO IT-NETS

Let t_{pk} be the value of the k th observed variable in the p th training sample and z_{pk} the corresponding IT-net approximation. Then the objective function to be minimized in network training is

$$E = \sum_p \sum_k (z_{pk} - t_{pk})^2 \quad (8.16)$$

The steepest descent direction for optimizing network inputs x_{pi} is given by

$$\Delta x_{pi} = -\frac{\partial E}{\partial x_{pi}} = \sum_k (t_{pk} - z_{pk}) \frac{\partial z_{pk}}{\partial x_{pi}} \quad (8.17)$$

Assuming that input and output nodes use the identity activation function, like in this study, while hidden nodes use a sigmoidal function ($f = \sigma$), the network output is given by

$$z_{pk} = \sum_j w_{kj} \sigma(b_j + \sum_i v_{ji} x_{pi}) \quad (8.18)$$

where $\sigma(\cdot)$ is a sigmoidal function, b_j is the bias of the j th hidden node, and V_{ji} and W_{kj} are network weights. Hence, the steepest descent direction for training network inputs is

$$\Delta x_{pi} = \sum_j v_{ji} \delta_{pj} \quad (8.19)$$

where δ_{pj} is the propagated error at the hidden layer and has been defined as

$$\delta_{pj} = \sigma'(b_j + \sum_i v_{ji} x_{pi}) (\sum_k w_{kj} (t_{pk} - z_{pk})) \quad (8.20)$$

Note that the steepest descent direction for training network weights between the input layer and the hidden layer is

$$\Delta v_{ji} = \sum_p x_{pi} \delta_{pj} \quad (8.21)$$

Therefore, the extra computation required for training the inputs is negligible compared with training the rest of the network. In the preceding derivation, it is assumed that only hidden nodes use sigmoidal functions and that input and output nodes are linear since this is the setup applied in this study. The same derivation can be carried out for networks with sigmoidal output and/or input nodes and Equation 8.19 still holds but with different δ_{pj} .

8.3. Levenberg-Marquardt

While backpropagation is a steepest descent algorithm, the Levenberg-Marquardt algorithm is an approximation to Newton's method. The Levenberg-Marquardt algorithm was designed to approach second-order training speed without having to compute the Hessian matrix. The Jacobian matrix can be computed through a standard backpropagation technique (Hagan and Menhaj, 1994) that is much less complex than computing the Hessian matrix. Suppose that we have a function $E(\mathbf{x})$ which we want to minimize with respect to the parameter vector \mathbf{x} , then Newton's method would be

$$\Delta \mathbf{x} = -[\nabla^2 E(\mathbf{x})]^{-1} \nabla E(\mathbf{x}) \quad (8.22)$$

where $\nabla^2 E(\mathbf{x})$ is the Hessian matrix and $\nabla E(\mathbf{x})$ is the gradient. If we assume that $E(\mathbf{x})$ is a sum of squares function

$$E(\mathbf{x}) = \sum_{i=1}^N e_i^2(\mathbf{x}) \quad (8.23)$$

then it can be shown that

$$\nabla E(\mathbf{x}) = J^T(\mathbf{x}) \mathbf{e}(\mathbf{x}) \quad (8.24)$$

$$\nabla^2 E(\mathbf{x}) = J^T(\mathbf{x}) J(\mathbf{x}) + S(\mathbf{x}) \quad (8.25)$$

where $J(\mathbf{x})$ is the Jacobian matrix, which contains first derivatives of the network errors with respect to the weights and biases, and \mathbf{e} is a vector of network errors.

$$J(\mathbf{x}) = \begin{bmatrix} \frac{\partial e_1(\mathbf{x})}{\partial x_1} & \frac{\partial e_1(\mathbf{x})}{\partial x_2} & \dots & \frac{\partial e_1(\mathbf{x})}{\partial x_n} \\ \frac{\partial e_2(\mathbf{x})}{\partial x_1} & \frac{\partial e_2(\mathbf{x})}{\partial x_2} & \dots & \frac{\partial e_2(\mathbf{x})}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_N(\mathbf{x})}{\partial x_1} & \frac{\partial e_N(\mathbf{x})}{\partial x_2} & \dots & \frac{\partial e_N(\mathbf{x})}{\partial x_n} \end{bmatrix} \quad (8.26)$$

and

$$S(\mathbf{x}) = \sum_{i=1}^N e_i(\mathbf{x}) \nabla^2 e_i(\mathbf{x}) \quad (8.27)$$

For the Gauss-Newton method it is assumed that $S(\mathbf{x}) \approx 0$, and the updated equation 8.22 becomes

$$\Delta \mathbf{x} = [J^T(\mathbf{x})J(\mathbf{x})]^{-1} J^T(\mathbf{x})\mathbf{e}(\mathbf{x}) \quad (8.28)$$

The Levenberg-Marquardt modification to the Gauss-Newton method is

$$\Delta \mathbf{x} = [J^T(\mathbf{x})J(\mathbf{x}) + \mu I]^{-1} J^T(\mathbf{x})\mathbf{e}(\mathbf{x}) \quad (8.29)$$

The parameter μ is multiplied by some factor (β) whenever a step would result in an increased $E(\mathbf{x})$. When a step reduces $E(\mathbf{x})$, μ is divided by some factor ζ . In this case a value of $\mu = 0.001$ was selected as an initial value, with $\zeta = 0.1$ and $\beta = 10$. Thus, it is decreased after each successful step (reduction in performance function) and is increased only when a tentative step would increase the performance function. In this way, the performance function will always be reduced at each iteration of the algorithm. Notice that when μ is large the algorithm becomes steepest descent (with step $1/\mu$), while for small μ the algorithm becomes Gauss-Newton. The Levenberg-Marquardt algorithm can be considered a trust-region modification to Gauss-Newton.

The key step in this algorithm is the computation of the Jacobian matrix. For the neural network mapping problem the terms in the Jacobian matrix can be computed by a simple modification to the backpropagation algorithm. The performance index for the mapping problem is given by equation 8.16. It is easy to see that this is equivalent in form to equation 8.23, where

$$x = [w^1(1,1)w^1(1,2) \dots w^1(S1,R)b^1(1) \dots b^1(S1)w^2(1,1) \dots b^M(SM)]^T \quad (8.30)$$

and $N = Q \times SM$

Standard backpropagation calculates terms like

$$\frac{\partial \hat{E}}{\partial w^k(i, j)} = \frac{\partial \sum_{m=1}^{SM} e_q^2(m)}{\partial w^k(i, j)} \quad (8.31)$$

For the elements of the Jacobian matrix that are needed for the Marquardt algorithm we need to calculate terms like

$$\frac{\partial e_q(m)}{\partial w^k(i, j)} \quad (8.32)$$

These terms can be calculated using the standard backpropagation algorithm with one modification at the final layer

$$\Delta^M = -\dot{F}^M(\mathbf{n}^M) \quad (8.33)$$

Note that each column of the matrix in equation 8.33 is a sensitivity vector which must be backpropagated through the network to produce one row of the Jacobian.

The Marquardt modification to the backpropagation algorithm thus proceeds as follows:

1. Present all inputs to the network and compute the corresponding network outputs (using equations 8.3 and 8.4), and errors $\mathbf{e}_q = \mathbf{t}_q - \mathbf{a}_q^M$. Compute the sum of squares of errors over all inputs ($E(\mathbf{x})$).
2. Compute the Jacobian matrix (using equations 8.33, 8.12, 8.10, 8.11, and 8.26).
3. Solve equation 8.29 to obtain $\Delta \mathbf{x}$.
4. Recompute the sum of squares of errors using $\mathbf{x} + \Delta \mathbf{x}$. If this new sum of squares is smaller than that computed in step 1, then reduce μ by β , let $\mathbf{x} = \mathbf{x} + \Delta \mathbf{x}$, and go back to step 1. If the sum of squares is not reduced, then increase μ by β and go back to step 3.
5. The algorithm is assumed to have converged when the norm of the gradient (equation 8.24) is less than some predetermined value, or when the sum of squares has been reduced to some error goal.

The training parameters as used in Matlab for the Levenberg-Marquardt training algorithm are:

- Maximum number of epochs to train (1000);
- Epochs between showing progress or updating display (25);
- Performance goal which is the sum-squared error goal (0.02);
- Minimum performance gradient (1e-6);
- Maximum validation failures (10);
- An initial value for μ (0.001);
- A value by which μ is multiplied whenever the performance function is reduced by a step (10);
- A value by which μ is multiplied whenever a step would increase the performance function (0.1);
- A maximum value of μ so that if μ becomes larger than this maximum value, the algorithm is stopped (1e10);
- Learning rate for input training (0.1).

The values in brackets are the default values.

8.4. Concept of Input Training

Instead of training a whole three-hidden-layer autoassociative network, only its demapping subnet can be trained. Training such a subnet is meaningful and can be done by extending the backpropagation algorithm, and in this case also combining it with the Levenberg-Marquardt training algorithm.

The difference between training a demapping subnet and training an ordinary feedforward network is that the inputs to the subnet are not given. Not only the internal network parameters but also the input values need to be changed to reproduce the given data as accurately as possible. When network inputs are adjusted, each output sample should be uniquely associated with one input vector. Figure 8.3 shows a 2-4-5 input training network with input adjustment used for reducing the dimensionality of a data set from five to two. Each input vector $(x_{p1} \ x_{p2})^T$ is adjusted to minimize only the error of its corresponding output vector $(z_{p1} \ z_{p2} \ \dots \ z_{p5})^T$ while internal network parameters are trained using all output samples.

After the demapping subnet and its inputs are properly trained, we obtain a reduced matrix and a demapping model in the form of a neural network. Thus all requirements for data dimensionality reduction can be fulfilled through training a single-hidden-layer network and its input simultaneously. The concept of input training (IT) gives an alternative to the autoassociative network architecture for reducing data dimensionality. It is this architecture that is referred to as an IT-net. Two characteristics are basic for an IT-net: the input layer has fewer nodes than any other layer, and inputs are adjusted according to corresponding outputs.

Note that the term input in the context of input training slightly differs from what is used for traditional neural networks, where inputs are always given. It is not unusual, however, to adjust inputs to a model while its parameters are being modified to minimize the output error. Examples of this model-fitting strategy include the polynomial PCA and factor analysis (FA). Input training is an application of the same strategy in neural networks. Training an IT-net with one input node and no hidden layer is equivalent to PCA.

IT-nets are basically feedforward networks. With one hidden layer of sigmoidal nodes, a feedforward network can approximate any nonlinear function to an arbitrary accuracy given sufficient hidden nodes (Cybenko, 1989). Let $\phi_k(\lambda_1, \dots, \lambda_f)$, $k = 1, \dots, n$, denote nonlinear mappings by an IT-net. When the output error for a given data vector, $(t_{p1}, \dots, t_{pn})^T$, is minimized at an input vector, $(x_{p1}, \dots, x_{pf})^T$, we have:

$$\left[\frac{\partial}{\partial \lambda_i} \sum_k (\phi_k - t_{pk})^2 \right]_{\lambda_i = x_{pi}} = 0, \quad i = 1, \dots, f \quad (8.34)$$

which can be rearranged into

$$\sum_k (z_{pk} - t_{pk}) \left[\frac{\partial \phi_k}{\partial \lambda_i} \right]_{\lambda_i = x_{pi}} = 0, \quad i = 1, \dots, f \quad (8.35)$$

where $z_{pk} = \phi_k(x_{p1}, \dots, x_{pf})$.

If the IT-net has exactly one input node, the functions $\phi_k(\lambda)$, $k = 1, \dots, n$, represents a smooth curve in n -dimensional space. Equation 8.35 indicates that the vector of output errors, $z_p - t_p$, is orthogonal to the tangent of the curve at $\lambda = x_p$ when the sum of the square errors is minimized through input training. Therefore, the result of training a single-input-node IT-net with a hidden layer of sigmoidal nodes is equivalent to a principal curve as defined by Hastie and Stuetzle (1989). Similarly, training a two-input-node IT-net will result in a principal surface, which is a vector of continuous functions

driven by two parameters and minimizes the orthogonal deviation of the data from the surface (Hastie and Stuetzle, 1989).

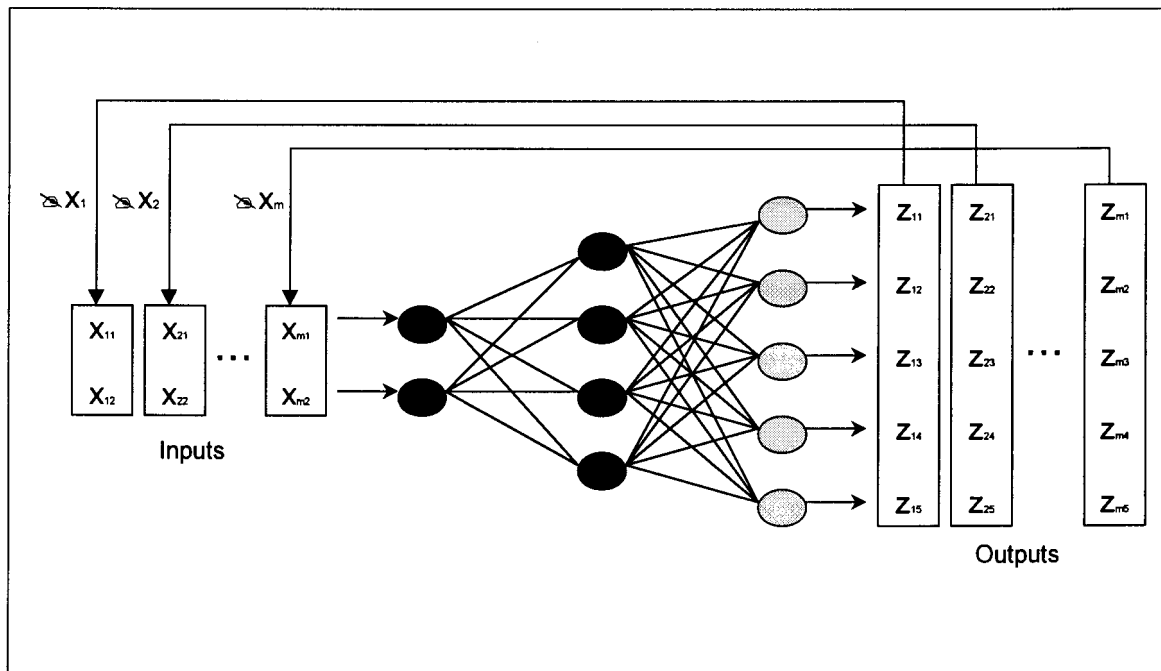


Figure 8.3. Concept of input training

8.5. Training IT-Nets

For IT-nets with hidden layers, a direct iterative procedure for network inputs is not available. The backpropagation and Levenberg-Marquardt training method discussed in Section 8.2 and 8.3 are applied to network inputs. Similar to network weights, network inputs are modified using errors backpropagated from the output layer. Thus, the steepest descent direction for minimizing the output errors through adjustment of network inputs is used.

8.6. Input Training

For input training we start with the Levenberg-Marquardt algorithm and proceed to the end. When step 5 from section 8.4 is completed, we jump to the general backpropagation algorithm with a view adjustments. We first calculate g_k , the gradient, and multiply it by α_k , the learning rate, according to equation 16. Since this is

equivalent to δ_{pj} , and the new weights have been calculated from the LM-algorithm, we can implement equations 8.17 and 8.18 to update the inputs.

When starting the training procedure from the IT-Net interface (Figure 9.1 discussed in Chapter 9), the training status will be displayed every *show* iterations of the algorithm. The other parameters determine when the training is stopped. The training will stop if the number of iterations exceeds the number of *epochs*, if the performance function drops below *goal*, if the magnitude of the gradient is less than the minimum gradient, or if the training time is longer than *time* seconds.

8.7. Testing and Using IT-Nets

A trained IT-net can be tested through cross validation. In this sense, testing and using an IT-net involve the same computing task. We will need to describe only testing. Because network inputs are unknown for testing samples, the appropriate way to test a trained network is to adjust network inputs while freezing all internal network parameters (weights and biases). That is, testing an IT-net still requires a searching procedure that optimizes each input pattern to yield a good approximation for its corresponding output sample.

Note that optimization of inputs for testing is much less time-consuming than training a whole IT-net. First, testing can be done for each individual sample and it involves much fewer searching variables than training the whole network. In addition, since the inputs of training data are available, we can apply a nearest neighbor algorithm to obtain good initial guesses for the inputs of testing samples. Finally, replacing the small fixed learning rate with a 1-D search significantly improves the speed of optimizing network inputs.

As an alternative to the preceding testing method, after training an IT-net we might proceed to train another network that maps the observed data to the reduced data. An autoassociative network could then be constructed by combining the mapping network and the IT-net. The major motivation for doing this is that using a trained autoassociative network would then be as simple as feedforward calculation. However, the result of this method of construction of an autoassociative network is often disappointing due to the following factors:

- (1) Training errors are introduced twice
- (2) Two network training rounds are independent of each other and there is no effective mechanism to ensure the quality of the whole autoassociative network.

In this study this alternative mapping method proved to be effective and gave acceptable results.

9.1. Introduction

In practice, both linear and non-linear correlations exist between process variables. The presence of linear correlations within the data impacts upon the non-linear PCA algorithm in terms of its ability to extract a parsimonious description of the underlying characteristics of the process. The presence of linear correlations between the variables results in the need for higher dimensionality to define the underlying non-linear structure.

Linear PCA is effectively a rotation. Since the rotation is carried out in linear space, the non-linear structures will be encapsulated within the principal component scores sub-space if the dimensionality of the sub-space is sufficiently large. Thus, the process of extracting linear and non-linear correlations from the data can be performed separately. In this chapter a non-linear PCA approach is proposed which combines the advantages of both linear PCA and the previous non-linear PCA algorithms. This is done by extending the principles of Input Training as discussed in Chapter 8 to nonlinear principal component analysis.

9.2. Nonlinear Principal Component Analysis

So how does NLPCA differ from LPCA? As we have seen, linear principal components analysis (LPCA) is a projection-based statistical tool traditionally used for dimensionality reduction. Consider an m -dimensional data set $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m]$ whose variance-covariance matrix has eigenvalue-eigenvector pairs $(\lambda_1, \mathbf{p}_1), (\lambda_2, \mathbf{p}_2), \dots, (\lambda_m, \mathbf{p}_m)$ where $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m \geq 0$. The linear principal component decomposition of \mathbf{X} can be represented as:

$$\mathbf{X} = \mathbf{TP}^T + \mathbf{E} = \sum_{i=1}^l \mathbf{t}_i \mathbf{p}_i + \mathbf{E} \quad (l < m)$$

where $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_l]$ is defined to be the matrix of principal component scores, $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_l]$ is the matrix of principal component loadings and \mathbf{E} is the residual matrix in the sense of minimum Euclidean norm. Non-linear PCA is an extension of linear PCA. Whilst PCA identifies linear correlations between process variables, non-linear PCA can extract both linear (second-order statistics) and non-linear (higher-order statistics) correlations. This generalisation is achieved by projecting the process

variables down onto curves or surfaces instead of lines or planes using the same objective function, i.e. minimising the mean-square error $E\{\|\mathbf{X} - \hat{\mathbf{X}}\|^2\}$. The data \mathbf{X} can be expressed in terms of k non-linear principal components, where $k \ll m$,

$$\mathbf{X} = F(\mathbf{T}) + \mathbf{E} \quad (9.1)$$

where $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_k]$ is the matrix of non-linear principal component scores, F is the non-linear function equivalent to the loadings in linear PCA and \mathbf{E} is the matrix of residuals.

The IT-net discussed in Chapter 8 can effectively represent non-linear systems which include both additive and multiplicative types of non-linearities through the simultaneous calculation of the latent variables. A potential disadvantage of this approach is that the complexity of training the IT-net increases exponentially with the dimensionality of the training dataset (Bakshi, 1998).

Two steps form the basis of the algorithm. In the first step, linear PCA is applied to the original observations as in Chapter 7, resulting in a new set of uncorrelated ordinates. By retaining sufficient data variability, the underlying non-linear structure is not compromised and only those linear principal components associated with noise are discarded. By reducing the dimensionality of the data, the non-linear structure becomes more apparent. In the second step the IT-net is used to extract the latent non-linear structure in the transformed dataset. To overcome the problem of local minima, the training of the IT-net is repeated several times with different initial conditions for each network architecture to ensure that the global minimum is found. The proposed non-linear PCA representation can be defined as follows:

$$\mathbf{X} = F(\mathbf{T}) \cdot \mathbf{P}^T + \mathbf{E} \quad (9.2)$$

where $\mathbf{T} (k < l < m)$ is the matrix of non-linear principal component scores which are identified from the input layer of the IT-net at the second stage and $\mathbf{P} (m \times l)$ is the matrix of linear principal loadings calculated from the first stage of the algorithm. $F(\cdot)$ represents the input-training network function and \mathbf{E} is the matrix of model residuals. Equation 9.2 gives the non-linear principal scores \mathbf{T} . However, for a new observation, to calculate the corresponding non-linear principal component score requires the implementation of a time consuming non-linear optimisation algorithm and is thus not appropriate for on-line application. An alternative and more straightforward approach is to develop a model between the process observations \mathbf{X} and the non-linear principal scores \mathbf{T} using a feed-forward neural network. Finally, the function relating the non-linear principal component scores to the process observations is defined as

$$\mathbf{T} = G(\mathbf{X} \cdot \mathbf{P}) \quad (9.3)$$

The function $G(\cdot)$ is the feed-forward neural network model with the linear PCA transformed data set $\mathbf{X} \cdot \mathbf{P}$ as the input layer and the non-linear principal scores \mathbf{T} as the output layer. Once models based on Equations 9.2 and 9.3 are built, the task of developing an on-line monitoring and fault detection scheme is straightforward requiring minimal computational effort.

9.3. Application

9.3.1. SOFTWARE SETUP

The NLPCA setup interface can be accessed by using the Next button from the LPCA interface or can be accessed directly from the Main interface. In Figure 9.1 to Figure 9.4 the NLMSPCA model is created. The use of the interfaces are straightforward.

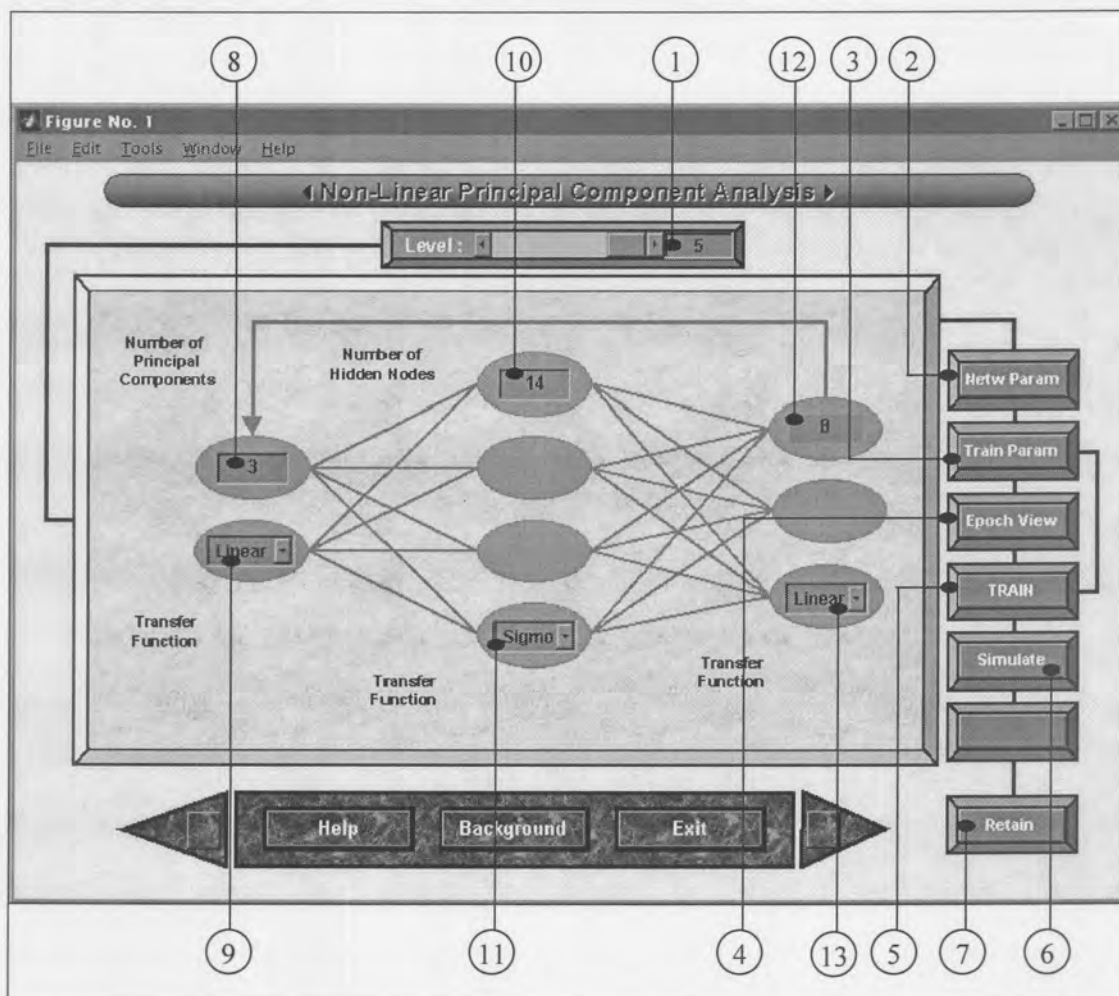


Figure 9.1. IT-Net parameter setup interface

Figure 9.1 Tags:

1. Dataset used to construct the NLMSPCA model. In this case it will either be dataset 1 or dataset 2.
2. Display the Network Parameter window (Figure 9.1). This allows the user to set up the input training neural network structure.
3. Display the Training Parameters window (Figure 9.2). This allows the user to set up the training parameters for the input training neural network.
4. Display the Epoch Viewer. This allows the user to view the Mean Squared Error (MSE) graphically as the training of the IT-Net progresses in order to get an idea of how fast and how well the training is progressing.
5. Train the IT-Net. After setting up Figure 9.1 and Figure 9.2 and after selecting Figure 9.3 this button is used to start the training and the progress will be displayed via Figure 9.3. Depending on the network structure and the error goal this can take a long time to complete. Training of the IT-Net will stop as soon as one of the stopping criteria is reached or when terminated by the user.
6. Simulate the network. This will display Figure 9.4.
7. Retain the network. If the user is satisfied with the trained network the network structure and parameters can be saved to the database.
8. Number of nonlinear principal components. By default the same number of nonlinear as linear principal components are chosen. However, the user has the option to change the number of nonlinear principal components to retain. Recall that the linear principal component scores are used as initial input to the IT-Net.
9. Input node/level transfer function. By default a linear transfer function is used and should remain so.
10. Number of hidden nodes. This is actually the only network structure parameter, except for perhaps the number of nonlinear principal components, that will be changed. There is no set rule for the optimum number of hidden nodes and therefore the user will have to play around with this parameter together with the training parameters until acceptable results are obtained. The more hidden nodes, the longer the training will take.
11. Transfer function of the hidden node. By default the sigmoidal transfer function is used and it should not be necessary to use any other transfer function. Other options include the linear and tangent transfer functions.

12. Number of original variables. This will be automatically displayed according to the number of variables in the database. Recall that the process variables are used as output for the IT-Net.

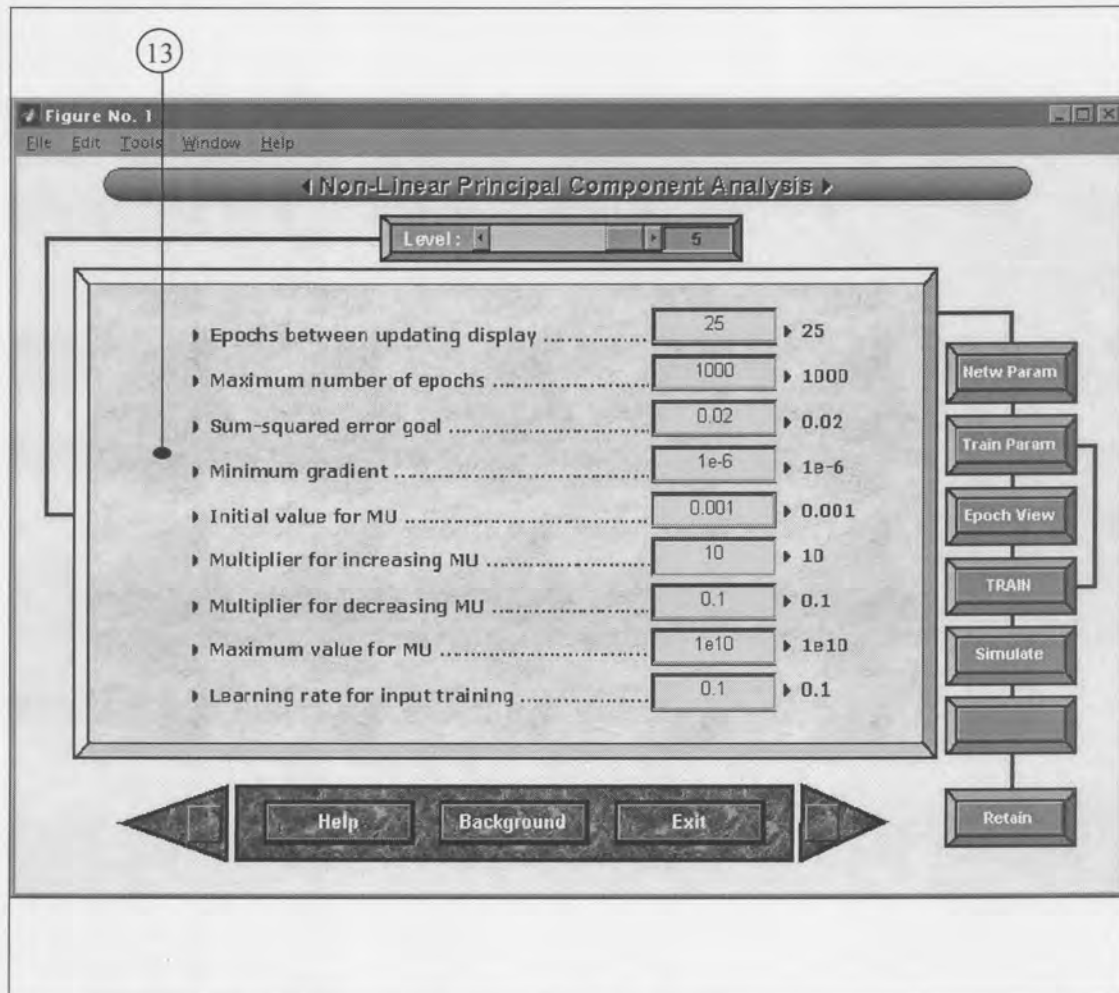


Figure 9.2. IT-Net training parameter interface

Figure 9.2 displays default values for the training parameters next to the edit boxes. These values have proven to work in most cases. If the error goal is not met after training the first parameter that can be altered is to increase the maximum number of epochs that will also increase the training time. After the training has stopped, a message will be displayed in the Matlab workspace indicating the stopping criteria used to stop the training. If, for example, training was terminated because the maximum value for mu was exceeded and the user is not yet satisfied with the performance, this will be an indication that the maximum value of mu needs to be increased. At this stage no other value except 0.1 for the learning rate will work.

Figure 9.2 Tags:

13. Training parameters display window.

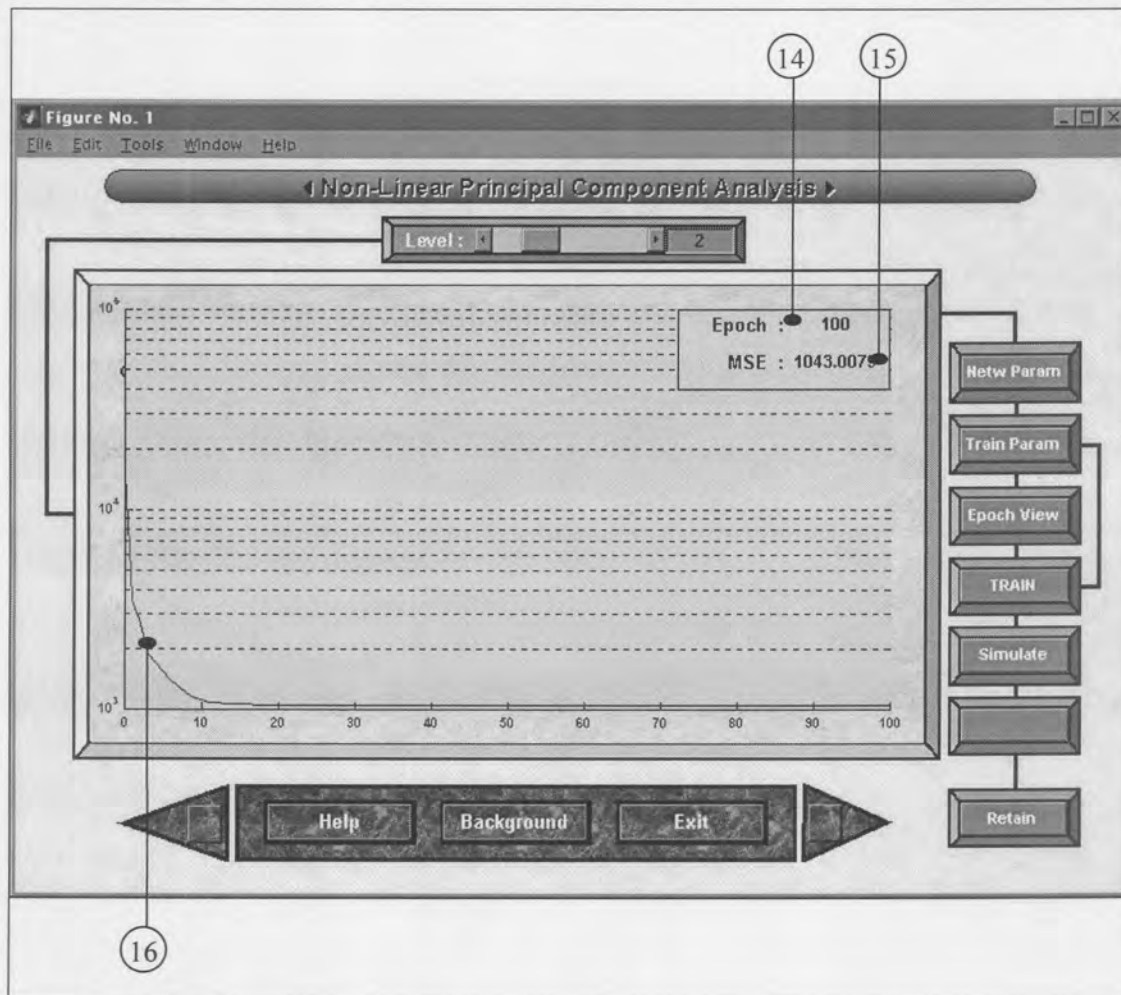


Figure 9.3. Epoch view interface

Figure 9.3 Tags:

14. Number of epochs completed so far.

15. The mean squared error of the network so far. This is also displayed via the graph. This window is updated on intervals specified in the Training Parameters window (Figure 9.3, Epochs between updating display).

16. MSE plot.

After the training has stopped the interface in Figure 9.4 can be used to simulate the network. The newly generated nonlinear principal components are used as input and the original process variables are used as output. This allows the user to visually inspect the performance of the network.

Figure 9.4 Tags:

17. Data name. By default the original data representing normal operation in the database is used. The user can however test the network performance by using

new unseen data. The name of the variable containing this data can be entered here. The variable must reside in the Matlab workspace with each column representing a separate process variable.

18. Process Variable number. Each process variable can be viewed separately.
19. Original versus simulated variable. The original process variable and the same process variable generated from the IT-Net through simulation are displayed to give a visual comparison between the original and trained data. This normally gives an indication of the level of network performance, especially in the case of new (validation) data.
20. Error plot between the original and simulated process variable.

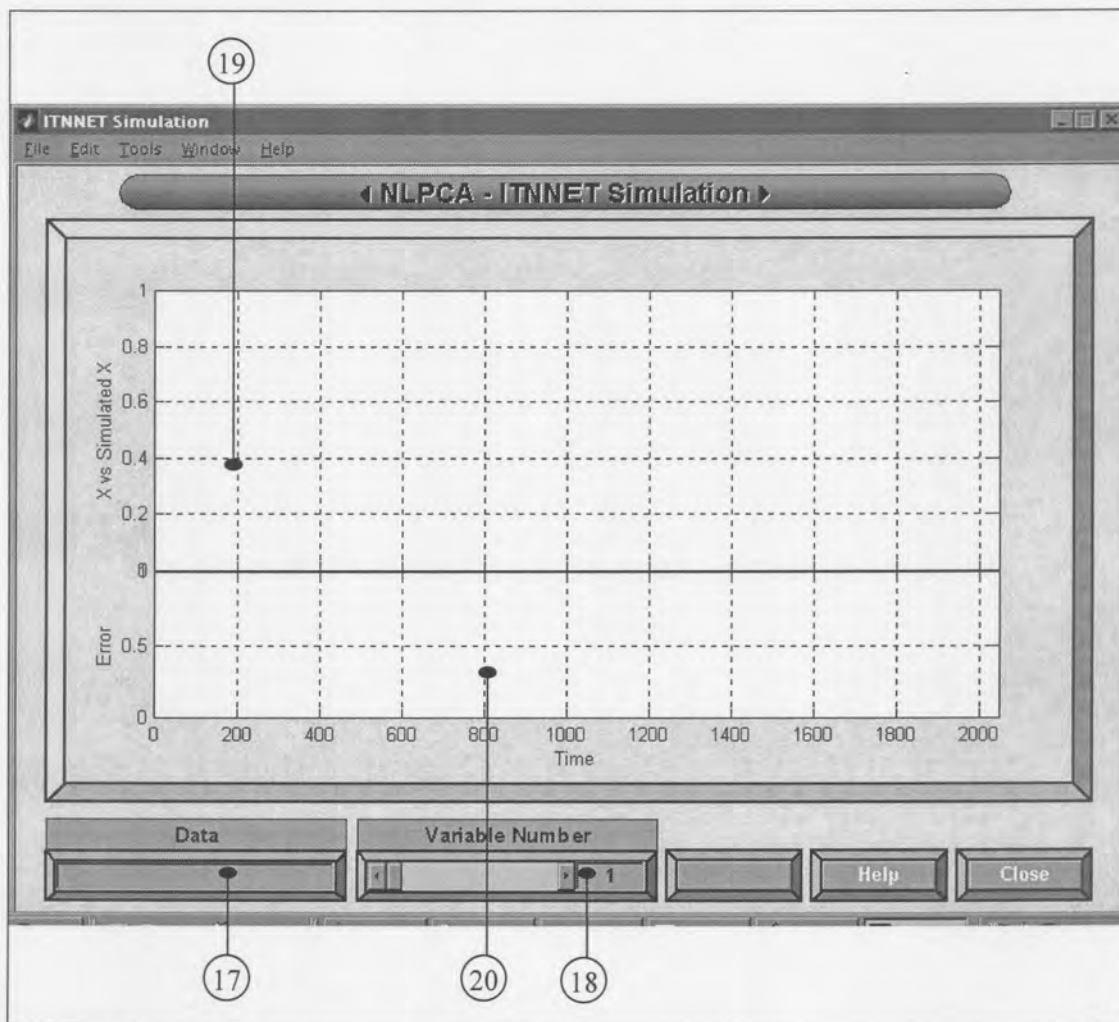


Figure 9.4. IT-Net simulation interface

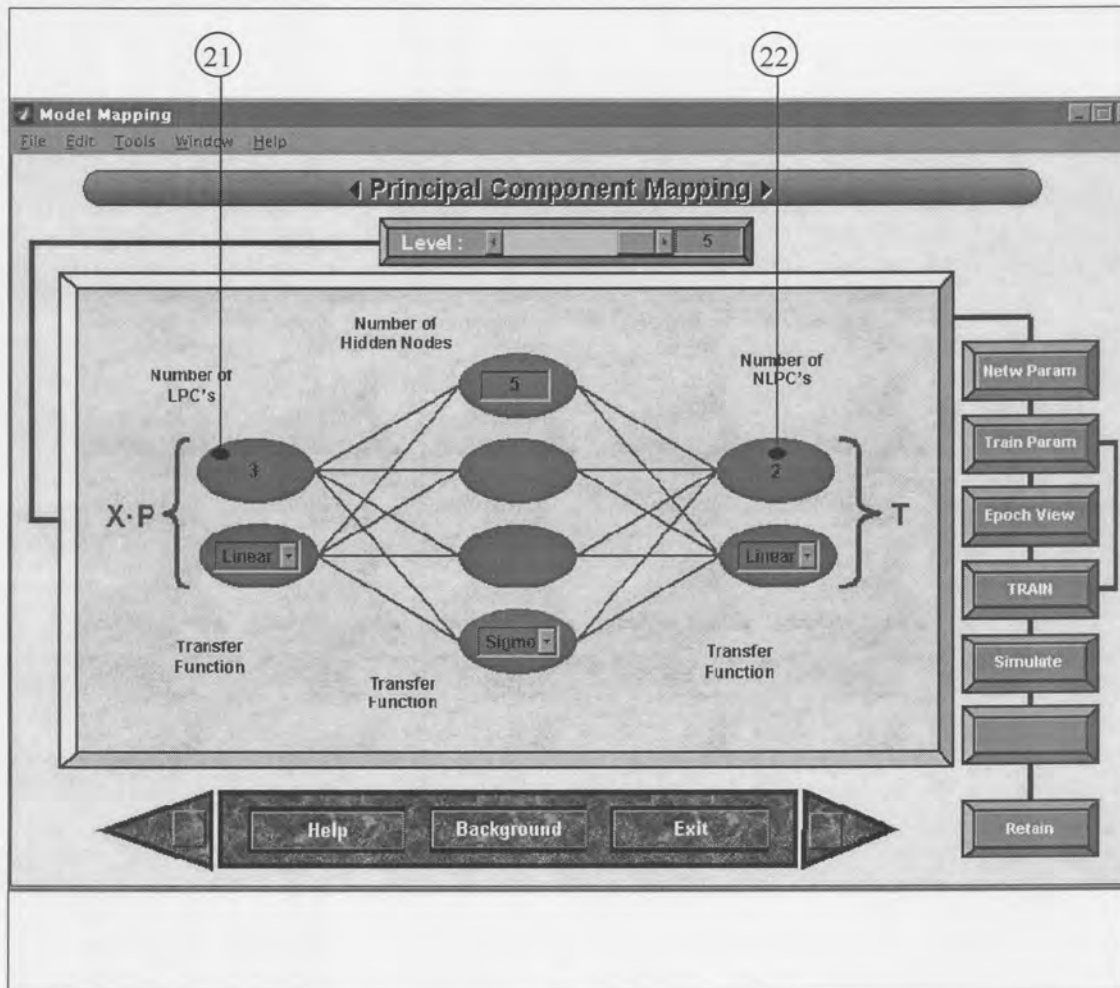


Figure 9.5. Neural network mapping interface

After the IT-Net has been trained the interface in Figure 9.5 can be used to generate a mapping model that generates a model between the linear and nonlinear principal component scores that can be connected to the IT-Net model. This interface is displayed by using the next button from the NLPCA interface (Figure 9.1) or can be accessed directly from the Main interface. Its setup and use are identical to Figure 9.1 to Figure 9.4 except for its inputs and outputs and for the fact that this is just a normal feedforward neural network that is trained using the Levenberg Marquardt training algorithm.

Figure 9.5 Tags:

21. Number of linear principal components. This is automatically displayed according to the number of linear principal components recorded in the database.
22. Number of nonlinear principal components. This is automatically displayed according to the number of nonlinear principal components recorded in the database.

9.3.2. EXPERIMENTAL

Non-linear PCA was applied to dataset 1 and dataset 2 using the IT-net methodology to extract the nonlinear correlations between the process variables and to build the nonlinear multiscale PCA representation.

Here, as in the case of linear principal component analysis, three principal components were retained in the model for dataset 1 capturing 97.72% of the data variability and again four for dataset 2 capturing 98.53% of the data variability making this a 3-4 principal component-model. The discarded principal components were attributed to process noise. Different network initialisation and neural network structures were used to train the neural networks to address the local minima problem. Finally, an IT-net with a 4-13-40 structure and a 40-18-4 feedforward neural network were built to model the projection of the data onto the nonlinear sub-space and its inverse, respectively for dataset 2 and a 3-12-8 structure and 8-14-3 feedforward neural network for dataset 1. The training parameters are summarised in Table 9.1. The shadowed blocks indicate the criteria on which the training was terminated for each case.

Table 9.1. Training parameters for IT-Net and mapping model

Training parameter	IT-Net		Mapping	
	Dataset 1	Dataset 2	Dataset 1	Dataset 2
Epochs between updating display	25	25	25	25
Maximum number of Epochs	2000	4000	1000	1000
Sum squared error goal	0.02	0.02	0.01	0.01
<i>Sum squared error achieved</i>	<i>0.029</i>	<i>0.032</i>	<i>0.01</i>	<i>0.01</i>
Minimum gradient	1e-6	1e-6	1e-6	1e-6
Initial value for MU	0.001	0.001	0.001	0.001
Multiplier for increasing MU	10	10	10	10
Multiplier for decreasing MU	0.1	0.1	0.1	0.1
Maximum value of MU	1e10	1e10	1e10	1e10
Learning rate for Input Training	0.1	0.1	-	-

Using the combined backpropagation and Levenberg-Marquardt training algorithm resulted in an average improved convergence rate of 350%. The results showed that the final nonlinear PCA representation captures 89.5% of the variability in dataset 2 and 93.1% in dataset 1. For the validation data it is 89.9% and 94.4% respectively. Action and warning limits for the bivariate score and SPE plots were then derived.



10.1. Introduction

To be effective, a plant-wide control monitoring and performance assessment system should have the following properties:

- (i) automated background operation, including scheduled remote collection of control loop data and data integrity checks,
- (ii) theoretically sound, efficient, and automated computational procedures,
- (iii) decision support (for example, problem reporting by exception),
- (iv) technical support, and
- (v) a suitable user interface.

Together, these properties form the basis of a comprehensive control performance monitoring and assessment system, as shown in Figure 10.1.

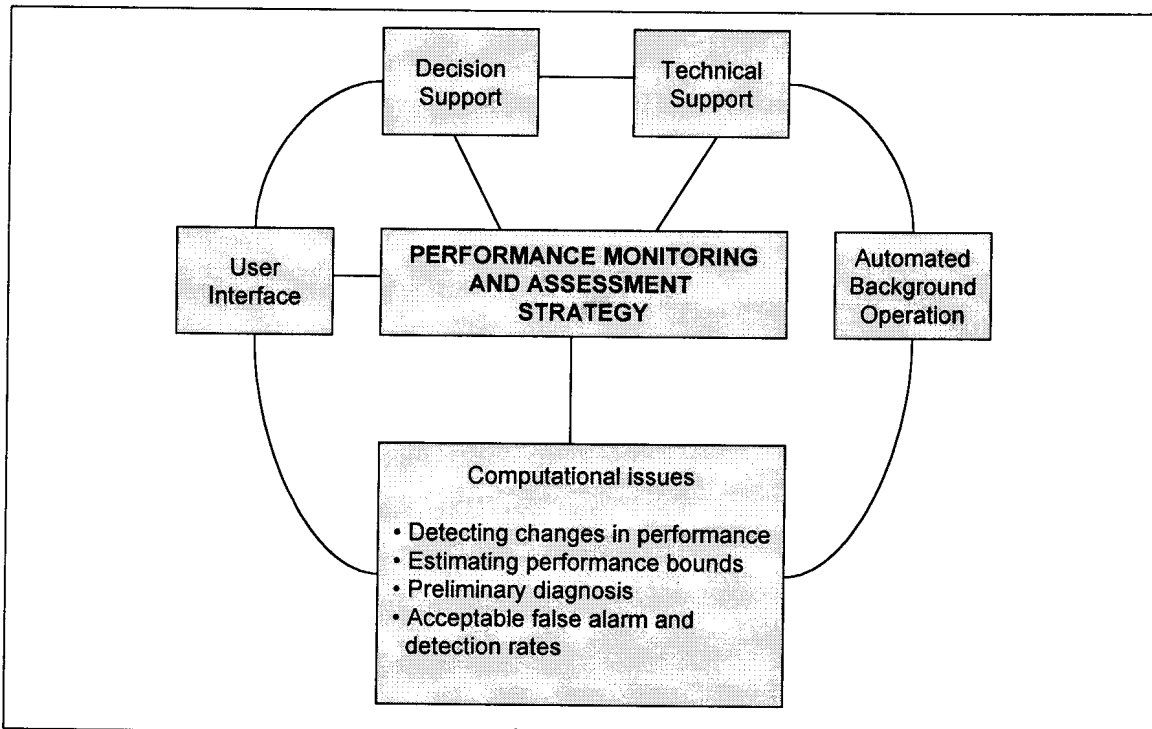


Figure 10.1. Important components of an industrial performance monitoring assessment, and diagnosis strategy.

The main aim here is to concentrate on the computational issues part of the industrial performance monitoring assessment and diagnosis strategy. Complete separate studies can be conducted on the other issues on their own and are therefore outside the scope of this study.

10.2. Interpretation

During the LPCA and NLPCA processes principal scores and loadings are calculated. A subset of the first few scores, $A < N$, provides information in a lower dimensional space, the *score space*, of the behaviour of the process during the period in which the measurements were made. This set of scores and the PCA loadings can be used to determine if the present process operation has changed its behavior relative to the data that were used to define the scores and loadings (Piovoso et al., 1992a).

There are several ways of interpreting the PCA results. Typical monitoring control charts include:

- the Q-statistic, a measure of the model mismatch;
- the Hotelling T^2 -statistic, a measure of the fit of new observations to the model space;
- variance plots, a measure of the samples' variability;
- univariate and bivariate principal component score plots, a qualitative representation of the process performance, relative to the calibration model in the model space defined by the calibration model;
- and a time-series plot of the squared prediction error (SPE).

These have been widely used to obtain early warning of the occurrence of nonconforming operation. Once a NLPCA representation has been built for process performance monitoring and fault detection, action and warning limits require to be calculated.

Bivariate score plots and the squared prediction error (SPE) were used in the approach adopted by Dong and McAvoy (1996) for defining the action and warning limits for their non-linear performance monitoring scheme. However, adopting this approach requires that the non-linear scores and residuals follow a multivariate normal distribution.

10.3. Action and warning limits

For multivariate statistical process monitoring by NLMSPCA, the region of normal operation is determined at each scale from data representing normal operation. For new data, an abnormal situation is indicated when the current coefficient violates the detection limits. The actual state of the process is confirmed by checking whether the signal reconstructed from the selected coefficients violates the detection limits of the PCA model for the significant scales. This approach is equivalent to adaptively filtering each value of the scores and residuals by a filter of dyadic length that is best suited for separating the deterministic change from the normal process variation. The detection limits for the scores and residuals also adapt to the nature of the signal.

These detection limits consist of action and warning limits. Warning limits are usually 95% confidence bounds and serve as a warning that the process is approaching an abnormal condition. Action limits are usually 99% confidence bounds indicating that an abnormal operation is occurring and that action needs to be taken.

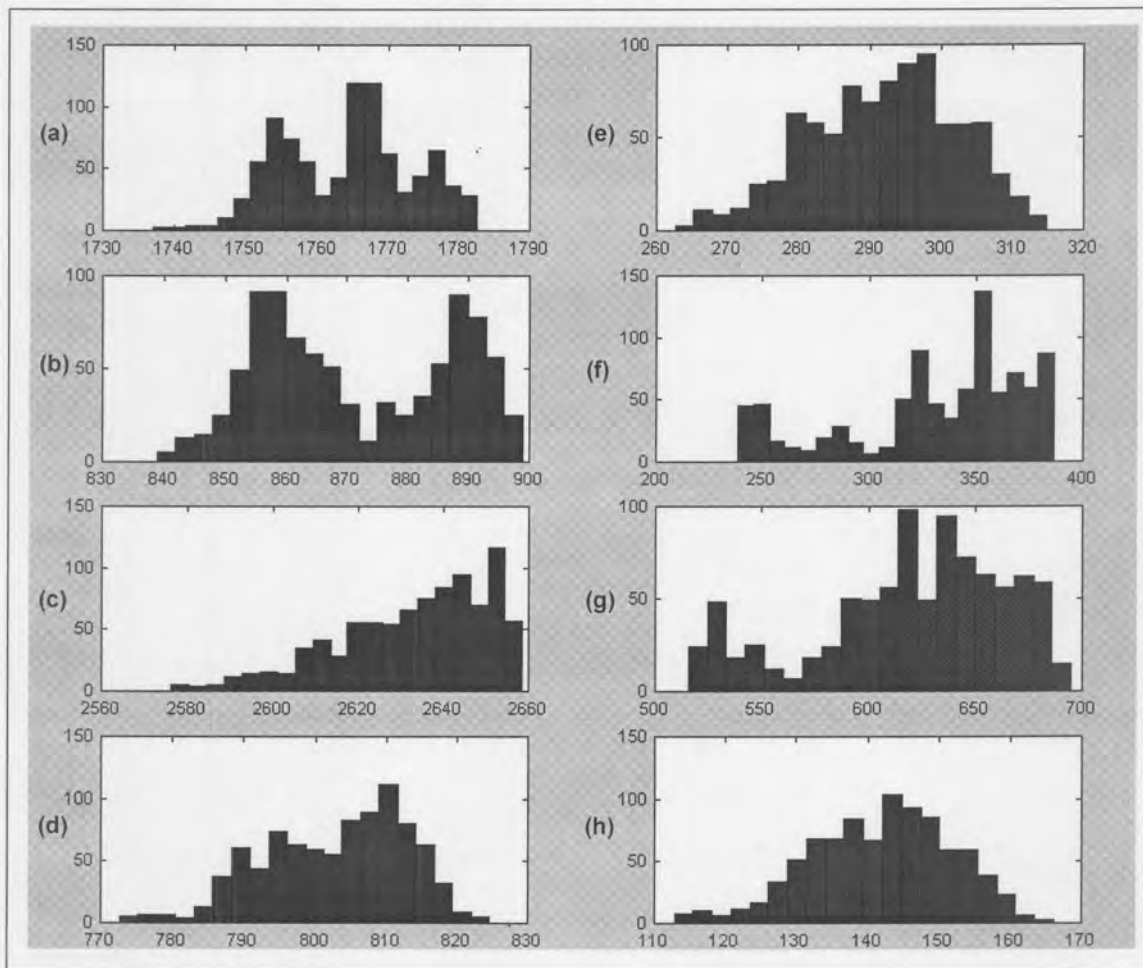


Figure 10.2. Histogram plots of the eight variables ((a)-(f)) used for training.

However, for non-linear PCA the assumption that the non-linear scores and residuals follow a multivariate normal distribution cannot be guaranteed. Although it is possible to define the region of normal operation without any underlying assumption concerning the probability distribution of the measurements, an assumption is still required to apply hypothesis-based statistical tests to identify when the process is moving outside the action or warning limits. The assumption of a normal distribution is incorrect as illustrated in Figure 10.2, which gives the histogram plots of the eight variables representing normal operation used in the application. Except for the last variable they do not closely resemble a normal distribution. Therefore, using this assumption will introduce a significant error. An alternative approach that effectively deals with this problem is introduced in Section 10.4.

10.4. Non-parametric bounds

An alternative approach to defining the action and warning limits is based upon non-parametric density estimation. Non-parametric bounds for process performance monitoring have previously been developed (Martin and Morris, 1996), using kernel estimation. Density estimation is the construction of an estimate of the density function from the observed data. A multivariate product kernel estimator can be constructed based upon the m -dimensional random samples $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ from a density f (Scott, 1992):

$$f(\mathbf{x}) = \frac{1}{nh^m} \sum_{i=1}^n K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right) \quad (10.1)$$

where h is the window width, also called the smoothing parameter or bandwidth, and K is the kernel function which satisfies the conditions

$$K(\mathbf{x}) \geq 0, \quad \text{and} \quad \int_{\mathbb{R}^m} K(\mathbf{x}) d\mathbf{x} = 1 \quad (10.2)$$

The shape of the density estimate is determined by the choice of the smoothing parameter h , and to a lesser extent by the choice of the kernel (Scott, 1992; Bowman, 1984). An automatic procedure for determining the optimal window width was used, the minimisation of the mean integrated squared error cross validation (Bowman, 1984). When using the density estimation-based approach to define the action and warning limits for the monitoring charts, the density function of the non-linear principal component scores and squared prediction error are calculated for the nominal (reference) data. Depending upon the confidence level required, 95% for the warning limits and 99% for the action limits, the contour or value can be calculated to define the control limits for the non-linear principal component scores plot and the SPE. Action and warning limits based upon kernel density estimation are theoretically more appropriate in the development of a non-linear PCA

monitoring scheme. However, if the underlying distribution is normal, similar results to those obtained from the conventional approaches are obtained.

10.5. Detection limit adjustment for on-line monitoring

For on-line monitoring, the MSPCA algorithm is applied to measurements in a moving window of dyadic length. The use of a moving window makes the on-line wavelet decomposition algorithm equivalent to wavelet decomposition without downsampling, causing a signal of length n to result in a total of $n(L+1)$ coefficients, where L is the depth of the wavelet decomposition. This increase in the number of coefficients requires on-line monitoring by MSPCA to increase the detection limits at each scale to maintain the desired confidence limit for the reconstructed signal. For example, for normally distributed uncorrelated measurements in a window length of 128, approximately one sample will lie outside the 99% confidence limits. The off-line wavelets transform will also result in 128 uncorrelated coefficients, and approximately one coefficient will violate the 99% limits. In contrast, the on-line wavelet transform of these data will result in 128 coefficients at each scale, and approximately one coefficient will violate the 99% detection limits at each scale. Thus, if the signal is decomposed to four detail signals and one scaled signal, that is, for $L = 4$ as in this case, application of the 99% confidence limit at each scale will result in an effective confidence of only 95% for the reconstructed signal, since the coefficients violating the detection limits at each scale need not be at the same location. Consequently, the detection limits at each scale for on-line monitoring by MSPCA need to be adjusted to account for the overcompleteness of the on-line wavelet decomposition by the following equation:

$$C_L = 100 - \frac{1}{L+1}(100 - C) \quad (10.3)$$

where C is the desired overall confidence limit, C_L is the adjusted confidence limit at each scale at present, and L is the number of scales to which the signal is decomposed, resulting in L detail signals and one scaled signal.

Another effect of the on-line wavelet decomposition approach is that the wavelet coefficients retain more of the autocorrelation in the signal due to the use of overlapping windows for the decomposition. Fortunately, the performance of monitoring by NLMSPCA is not adversely affected by the autocorrelated coefficients in adjacent windows, since the confidence limits at each scale are increased by equation 10.3, and even relatively small deterministic features are captured by large wavelet coefficients.

10.6. Residual analysis

Another interesting property of PCA is the fact that the equation

$$\mathbf{z} = \mathbf{U}'[\mathbf{x} - \bar{\mathbf{x}}] \quad (10.4a)$$

may be inverted so that the original variables may be stated as a function of the principal components, viz.,

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{U}\mathbf{z} \quad (10.4b)$$

because \mathbf{U} is orthonormal and hence $\mathbf{U}^{-1} = \mathbf{U}'$. This means that, given the z-scores, the values of the original variables may be uniquely determined. However, \mathbf{x} will be determined exactly only if all the pc's are used. If $k < p$ pc's are used, only an estimate $\hat{\mathbf{x}}$ of \mathbf{x} will be produced, viz.,

$$\hat{\mathbf{x}} = \bar{\mathbf{x}} + \mathbf{U}\mathbf{z} \quad (10.5)$$

where \mathbf{U} is now $p \times k$ and \mathbf{z} is $k \times 1$. Equation 10.4b can be rewritten as

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{U}\mathbf{z} + (\mathbf{x} - \hat{\mathbf{x}}) \quad (10.6)$$

a type of expression similar to those often found in other linear models. In this case, the first term on the right-hand side of the equation represents the contribution of the multivariate mean, the second term represents the contribution due to the pc's, and the final term represents the amount that is unexplained by the pc model – the residual.

Gnanadesikan and Kettenring (1972) divided multivariate analysis into

1. The analysis of internal structure.
2. The analysis of superimposed or extraneous structure.

There are outliers associated with each of these and it is important to keep their identities distinct. (Hawkins refers to them as Type A and Type B outliers.)

The Type A outlier refers to a general outlier from the distribution form one wishes to assume. Usually this assumption will be multivariate normal and these outliers will be detected by large values of T^2 and/or large absolute values of the z-scores. The important thing about this type of outlier is that it would be an outlier whether or not PCA has been employed and hence could be picked up by conventional multivariate techniques without using PCA. However, the use of PCA might well enhance the chance of detecting it as well as diagnosing what the problem might be.

Here we will be concerned with the Type B outlier, the third term in equation 10.6, which is an indication that a particular observation vector cannot be adequately characterized by the subset of pc's one chose to use. This result can occur either because too few pc's were retained to produce a good model or because the observation is, truly, an outlier from the model. It is also possible in repetitive operations, such as quality control, that the underlying covariance structure and its associated vector space may change with time. This would lead to general lack-of-fit by the originally defined pc's.

10.7. Biplots

When most of the variance of the variables is summarized by only two principal components, then we can express the results as a biplot. Although biplots are originally meant two-dimensional plots, they may be used for any number of dimensions. For two-dimensional plots it means that for the singular value decomposition we are summarizing a lot of the information in only two dimensions. Two-dimensional plots are very popular because they are easy to work with but should always include some statement with regard to the proportion of the total variability explained by the first two characteristic roots. Unless this quantity is sufficiently large, the interpretation of the plot is suspect.

Although biplots will be used, they will only be used indirectly. Another technique will be introduced for viewing the information contained in a biplot.

Since the data contained in samples when the process was not operating normally, applying PCA as a simple outlier detector revealed which data could be classified as such. Score biplots of the first few principal components can be used to visually detect these outliers (Piovoso et al., 1992).

10.8. Hotelling's T^2 statistic: An overall measure of variability

Hotelling's T^2 -statistic measures unusual variability within the calibration model space. That is, if the calibration model data represent process operation at one operating condition, and the process has shifted to a different one, then the T^2 -statistic will show that data at this operating condition cannot be classified with the calibration data. The T^2 -statistic is proportional to the sum of the squares of the scores on each of the principal components (Piovoso et al., 1992).

The T^2 statistic can be applied (Johnson and Wichin, 1992) to the principal component scores to calculate the control limits. It is based upon the assumption that the limits of the

control charts are calculated assuming that the original data \mathbf{X} follows a multivariate normal distribution. Under these assumptions, the principal component scores and residuals obtained from linear PCA will also exhibit normality since PCA is a linear transformation and a linear combination of a normal distribution is itself normally distributed.

The T^2 -quantity can be calculated as follows:

$$T^2 = \mathbf{y}'\mathbf{y} \quad (10.7)$$

which is a quantity indicating the overall conformance of an individual observation vector to its mean or an established standard. This quantity, due to Hotelling (1931), is a multivariate generalization of the Student t -test and does give a single answer to the question: "Is the process in control?"

The original form of T^2 is

$$T^2 = [\mathbf{x} - \bar{\mathbf{x}}]'\mathbf{S}^{-1}[\mathbf{x} - \bar{\mathbf{x}}] \quad (10.8)$$

which does not use PCA and is a statistic often used in multivariate quality control. Substituting $\mathbf{S}^{-1} = \mathbf{W}\mathbf{W}'$ and $y_i = \mathbf{w}'_i[\mathbf{x} - \bar{\mathbf{x}}]$ in equation 10.8 results in

$$\begin{aligned} T^2 &= [\mathbf{x} - \bar{\mathbf{x}}]'\mathbf{S}^{-1}[\mathbf{x} - \bar{\mathbf{x}}] \\ &= [\mathbf{x} - \bar{\mathbf{x}}]'\mathbf{W}\mathbf{W}'[\mathbf{x} - \bar{\mathbf{x}}] = \mathbf{y}'\mathbf{y} \end{aligned} \quad (10.9)$$

so equations 10.7 and 10.8 are equivalent. The important thing about T^2 is that it not only fulfills Condition 1 for a proper multivariate quality control procedure as listed in Chapter 7, Section 7.9.2, but Conditions 2 and 3 as well. The only advantage of equations 10.7 over 10.8 is that if \mathbf{W} has to be obtained, the computations are considerably easier as there is no matrix to invert. In fact, $\mathbf{y}'\mathbf{y}$ is merely the sum of squares of the principal components scaled in this manner ($T^2 = y_1^2 + y_2^2$ for the two-variable case) and demonstrates another advantage in using \mathbf{W} -vectors. If one uses \mathbf{U} -vectors, the computations become, essentially, a weighted sum of squares:

$$T^2 = \mathbf{z}'\mathbf{L}^{-1}\mathbf{z} \quad (10.10)$$

and the use of \mathbf{V} -vectors would produce a similar expression.

Few books include tables for the distribution of T^2 because it is directly related to the F -distribution by the relationship

$$T^2_{p,n,\alpha} = \frac{p(n-1)}{n-p} F_{p,n-p,\alpha} \quad (10.11)$$

In this example, $p = 8$, $n = 900$, $F_{8,892,0.05} = 3,8056$, so

$$T_{8,900,0.05}^2 = 8,187$$

An observation vector that produces a value of T^2 greater than 8,187 will be out of control on the chart.

However, the traditional approach to calculating action and warning limits for multivariate process performance monitoring based on Hotelling's T^2 , 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 as discussed earlier. This problem was addressed by calculating the control limits using the non-parametric technique of kernel density estimation in Section 10.4. This approach has the advantage that no a priori assumption of normality is required.

An alternative method of plotting T^2 is to represent it in histogram form, each value of T^2 being subdivided into squares of the y -scores. This is sometimes referred to as a stacked bar-graph, and indicates the nature of the cause of any out-of-control situations. However, the ordinate scale would have to be arithmetic rather than logarithmic.

Process monitoring can also be referred to as a form of multivariate quality control. The procedure or guidelines for monitoring a multivariate process using PCA is as follows:

1. For each observation vector, obtain the y -scores of the principal components and from these, compute T^2 . If this is in control, continue processing.
2. If T^2 is out of control, examine the y -scores. As the pc's are uncorrelated, it would be hoped that they would provide some insight into the nature of the out-of-control condition and may lead to the examination of particular original observations.

The important thing is that T^2 is examined first and the other information is examined only if T^2 is out of control. This will take care of the first three conditions listed in Section 7.9.2 and, hopefully, the second step will handle the fourth condition as well. Even if T^2 remains in control, the pc data may still be useful in detecting trends that will ultimately lead to an out-of-control condition.

10.9. The Q-statistic

The residual term of Equation 10.6 can be tested by means of the sum of squares of the residuals:

$$Q = (\mathbf{x} - \hat{\mathbf{x}})'(\mathbf{x} - \hat{\mathbf{x}}) \quad (10.12)$$

This represents the sum of squares of the distance of $\mathbf{x} - \hat{\mathbf{x}}$ from the k -dimensional space that the PCA model defines.

To obtain the upper limit for Q , let:

$$\theta_1 = \sum_{i=k+1}^p l_i$$

$$\theta_2 = \sum_{i=k+1}^p l_i^2$$

$$\theta_3 = \sum_{i=k+1}^p l_i^3$$

and

$$h_0 = 1 - \frac{2\theta_1\theta_3}{3\theta_2^2}$$

Then the quantity

$$c = \theta_1 \frac{\left[\left(\frac{Q}{\theta_1} \right)^h - \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} - 1 \right]}{\sqrt{2\theta_2 h_0^2}} \quad (10.13)$$

is approximately normally distributed with zero mean and unit variance (Jackson and Mudholkar, 1979). Conversely, the critical value for Q is

$$Q_\alpha = \theta_1 \left[\frac{c_\alpha \sqrt{2\theta_2 h_0^2}}{\theta_1} + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} + 1 \right]^{1/h_0} \quad (10.14)$$

where c_α is the normal deviate cutting of an area of α under the upper tail of the distribution if h_0 is positive and under the lower tail if h_0 is negative. This distribution

holds whether or not all of the significant components are used or even if some nonsignificant ones are employed.

In Section 2.6, it was suggested that the last two characteristic roots in the example were not significantly different from each other and hence the last two pc's were deleted. If only the first two pc's were retained, what would be the limit for Q ? The last two roots were 29.33 and 16.41. From these, $\theta_1 = 45.74$, $\theta_2 = 1129.54$, $\theta_3 = 29650.12$, and from these $h_0 = 0.291$. Letting $\alpha = 0.05$, the limit for Q , using equation 10.14 is

$$Q_{0.05} = 45.74 \left[\frac{(1.645)\sqrt{(2)(1129.54)(0.291)^2}}{45.74} + \frac{(1129.54)(0.291)(-0.709)}{(45.74)^2} + 1 \right]^{1/0.291}$$

$$= 140.45$$

Values of Q higher than this are an indication that a data vector cannot be adequately represented by a two-component model.

10.10. Contribution plots

When a new observation moves outside the control limits, it is assumed that an unusual process event or equipment malfunction has occurred and operator personnel need a tool to identify which variables, or combination of variables, are responsible for, or indicative of, changes in the process. One approach is through the implementation of a process variable contribution plot (Miller et al., 1993). Consequently, for the identification of variables indicative of non-conforming operation, differential contribution plots based upon model residuals and non-linear principal component scores are used.

By comparing the contribution plot of a sample taken from the calibration set with one that is outside the confidence limits, differences in the expected variables' magnitude may provide an indication of which variables have exceeded their expected limits, and a possible compensation to correct the problem.

Contribution plots describe the change in the magnitude of the variables for the new observation relative to the average value calculated from the nominal linear PCA model. It decomposes the scores into their summation operands and graphs them versus the contributing variable. The summation operands are the products of the loadings of variable j and the corresponding value of variable j . A large product associated with a particular variable implies a correspondingly large contribution (Piovoso et al., 1992)

Using a similar argument, a contribution plot can be derived and applied in a non-linear situation. The contribution of the process variables to the SPE can be calculated in a similar way to that for linear PCA. However, since the mapping function between the process variables and their non-linear principal scores is non-linear, the relationship between the variables and the non-linear principal scores is not as straightforward as in the linear case where the scores can be decomposed as a weighted sum of the process measurements. An alternative approach is based upon the assumption that the partial derivative of a function with respect to a specific dimension can indicate the relative influence of the corresponding variable on that function. If the first-order partial derivatives of a multivariate function are known for a specific variable space coordinate, then these derivatives can be used to compare the relative influence of the individual variables on the function at a particular location in variable space. Thus a differential contribution plot which describes the difference between the contribution of the process variables to its non-linear scores can be defined by comparing the influence of the first-order partial derivatives of the non-linear scores to each process variable for a specific sample or time point. The differential contribution plot that indicates the contribution of the process variables at a specific time point (\mathbf{x}_0) to a non-linear score t_i , can then be examined by calculating the individual components of the vector product

$$\mathbf{x} \Big|_{\mathbf{x}=\mathbf{x}_0} \cdot \frac{\partial \mathbf{t}}{\partial \mathbf{x}} \Big|_{\mathbf{t}=t_i} \quad (10.15)$$

where $\partial \mathbf{t} / \partial \mathbf{x}$ is the first-order partial derivative function between \mathbf{t} and \mathbf{x} . The relationship between the non-linear principal scores \mathbf{t} to the process variables \mathbf{x} is given in Equation 9.3. This approach is also suitable for linear PCA since linearity can be viewed as a special case of non-linearity. In the linear case, the first-order partial derivatives of \mathbf{t} relative to \mathbf{x} become constant which in practice are the principal component loadings, \mathbf{P} . Thus Equation 10.15 can be simplified to $\mathbf{x} \Big|_{\mathbf{x}=\mathbf{x}_0} \cdot \mathbf{p}_i$ which is the same expression as that for the contribution plot to the scores proposed by Miller et al. (1993).

10.11. Bivariate summary plots

Figure 10.3 illustrates the traditional biplots with detection limits for the normal linear case (a) and based on the kernel density estimation (b). The contours represent normal operation detection limits. If an observation moves outside these detection limits it will indicate that an abnormal condition has occurred. With a new observation we are actually just interested in how far from abnormal the condition is. This can be calculated using the methodology illustrated in Figure 10.4.

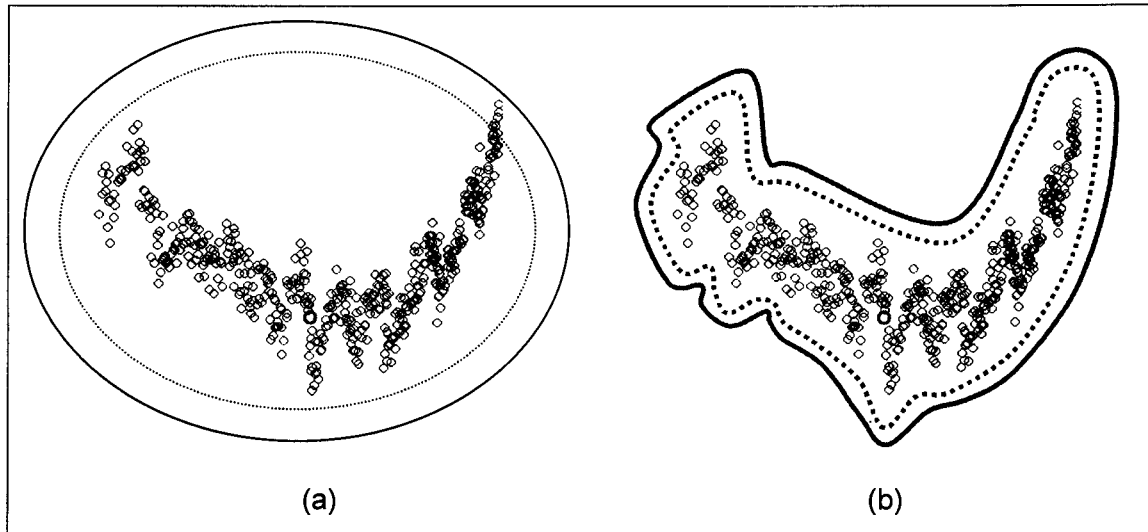


Figure 10.3. Traditional bivariate plot

We are interested in the shortest tangent line to the warning limit which can be extended to find the shortest tangent line to the action limit. In Figure 10.4(a) there exist two tangent lines with one (d_2) being the shortest line to any position on the warning limit contour. Figure 10.4(b) contains more than two such tangent lines. Thus, for a new observation we just need to calculate all the possible tangent lines from the observation point to the warning limit contour and select the shortest as an indication of how far the process is from a nonconforming condition. This can be calculated for different biplot combinations and summarized as illustrated in Figure 10.5 where the first bar is based on Figure 10.4(b). The warning limit line forms the baseline. d_m is a non-tangent line and thus will not be considered.

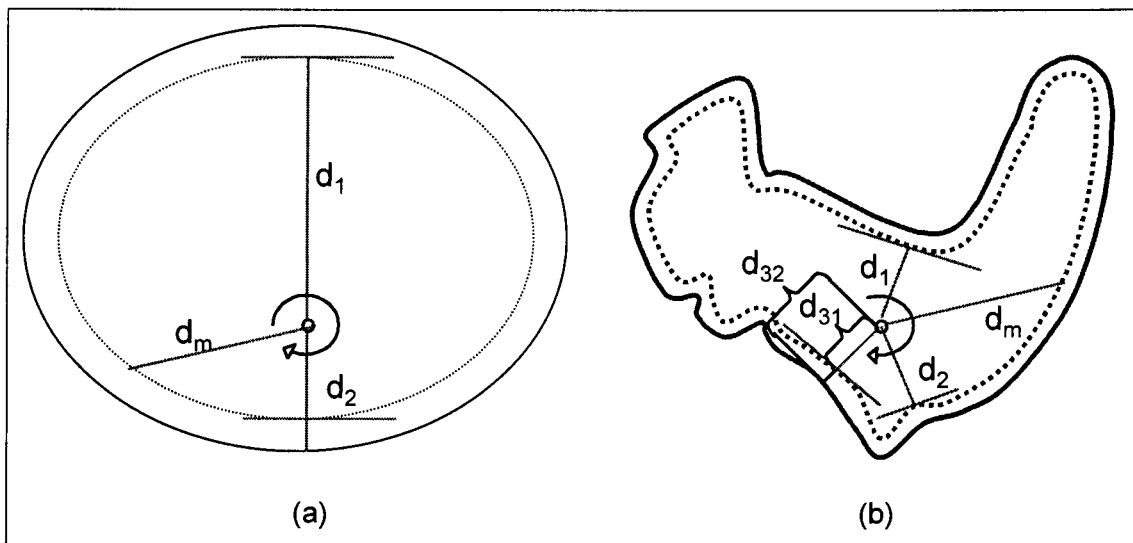


Figure 10.4. Bivariate summary plot calculation

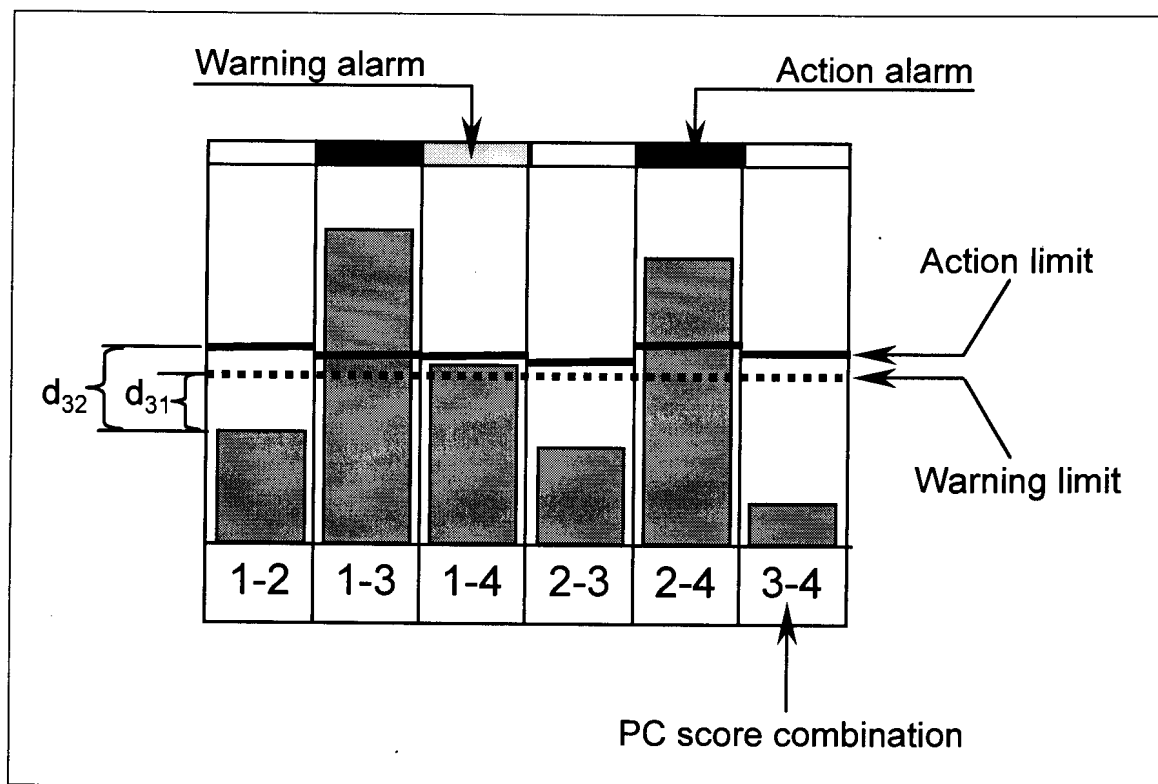


Figure 10.5. Bivariate summary plot for 6 biplots at one time instance

10.12. Application

10.12.1. SOFTWARE SETUP

Figure 10.6 is used to set up the bivariate contour plots. Its sole purpose is to choose the confidence limits for the action and warning limits. It shows the linear and nonlinear limits for comparison. The different combinations of scores can be viewed. The summary bivariate plots are automatically generated from the bivariate plots and thus do not need to be set up separately.

Figure 10.6 Tags:

1. Dataset number slider
2. Dataset number display. In this case one will first set up the bivariate plots for dataset one and then for dataset two.
3. Y-axes principal component number slider.
4. Y-axes principal component number display. For this application one will have a choice between three principal components for dataset one and four for dataset two.

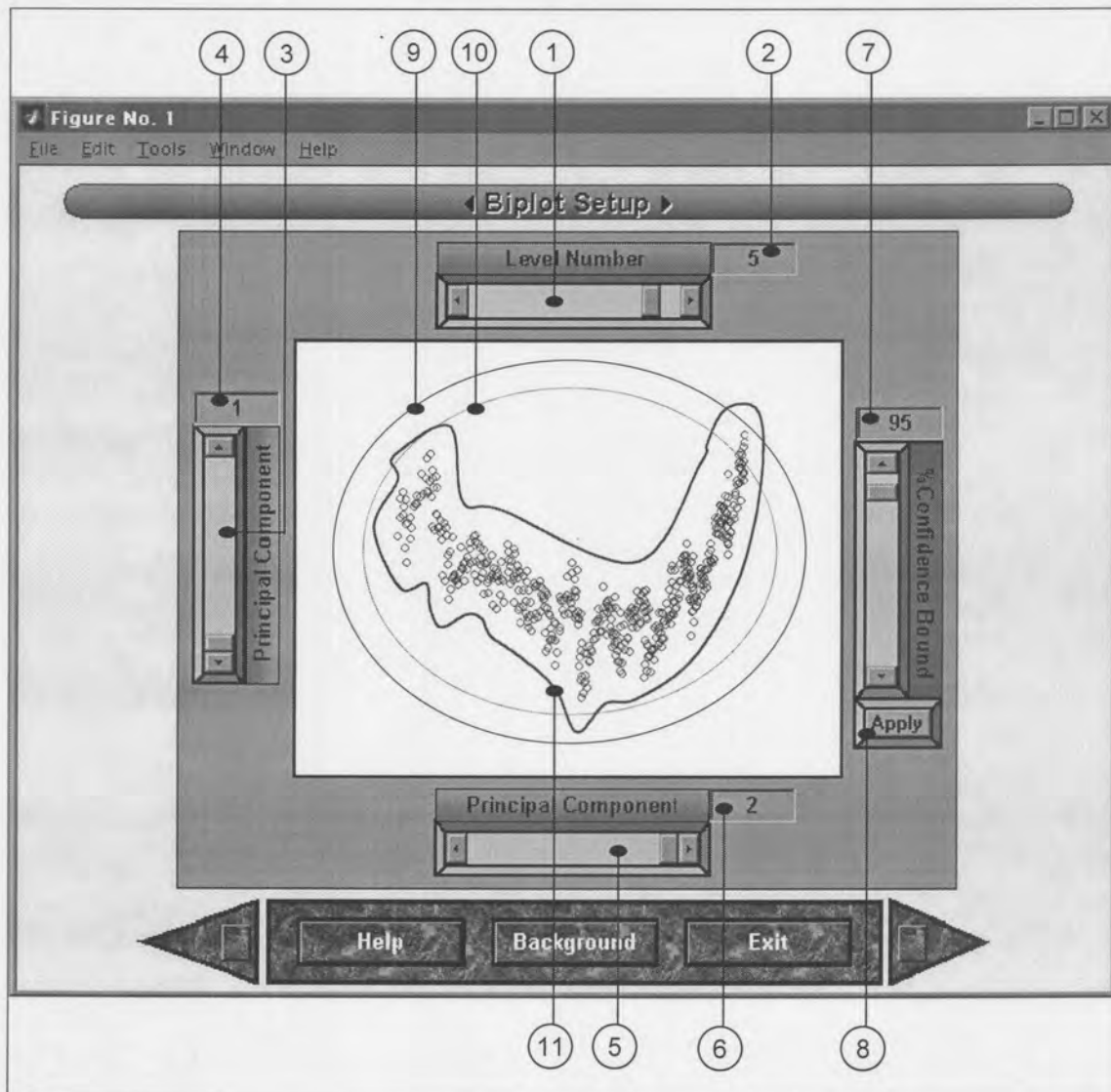


Figure 10.6. Bivariate plot setup interface

5. X-axes principal component number slider.
6. X-axes principal component number display. For this application one will have a choice between three principal components for dataset one and four for dataset two.
7. Confidence limit for the bivariate principal component score plot. As one changes the confidence limit the contour plot on the interface will change accordingly.
8. Application button to save the parameters to the database. The first one accepted will be the warning limit and the second one will be the action limit. After the second limit has been applied one can advance to the next pair of principal component scores for a new bivariate plot.
9. Action limit for linear case where normality is assumed.

10. Warning limit for linear case where normality is assumed.
11. Action limit for nonlinear case when using density estimation.

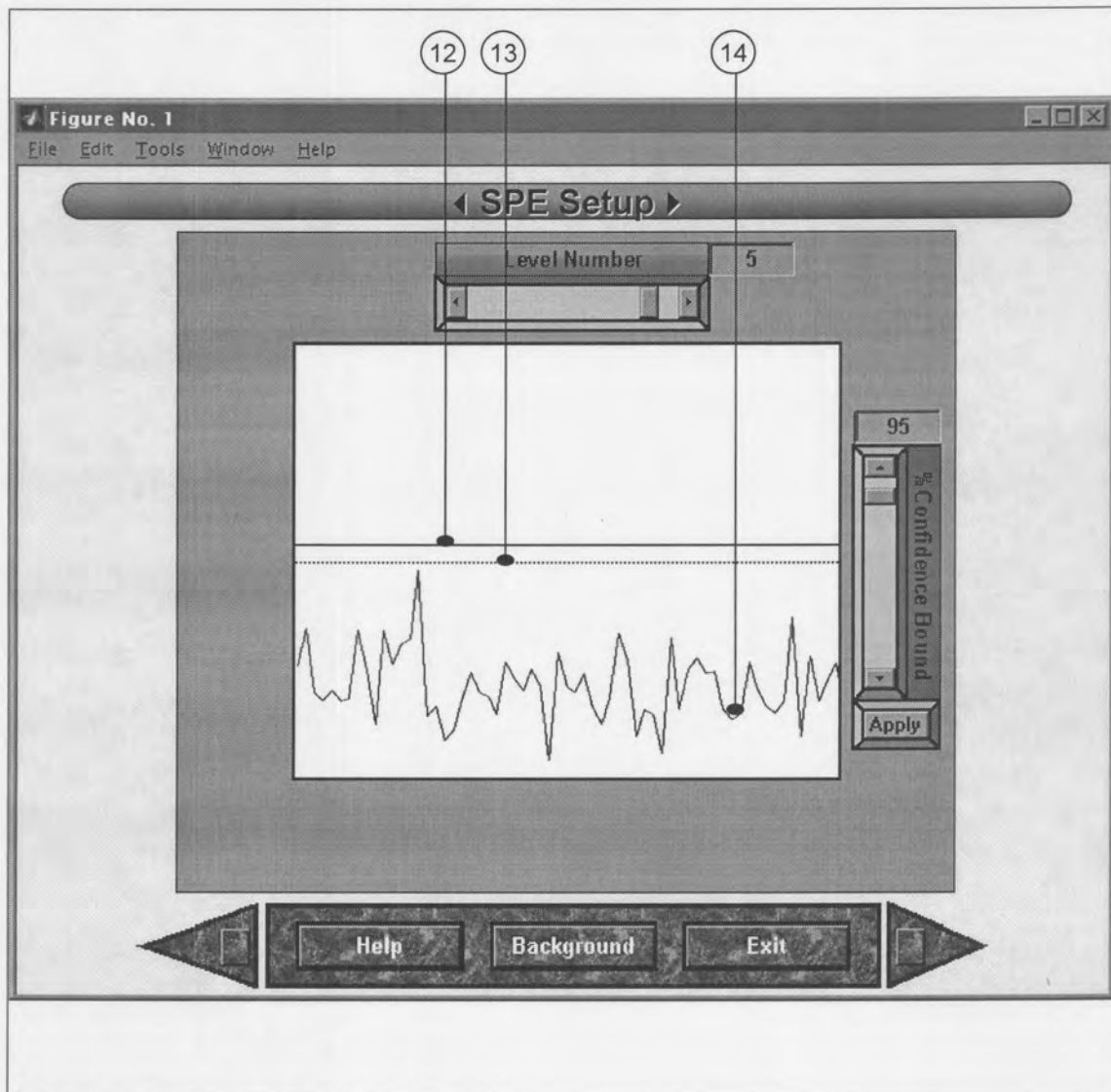


Figure 10.7. SPE Setup interface

Figure 10.7 is used to set up the action and warning limits for the SPE plot and works in a similar way to Figure 10.6.

Figure 10.7 Tags:

12. SPE action limit.
13. SPE warning limit.
14. SPE plot

Up to this point all previous software was just used to set up the NLMCPA model and is not used again except when changes to the model need to be done. All the relevant information is stored in the database. Figure 10.7 is the actual process monitoring interface which is used to monitor new process data. This interface can be accessed by using the *next* button in Figure 10.6 or can be accessed directly via the Main interface.

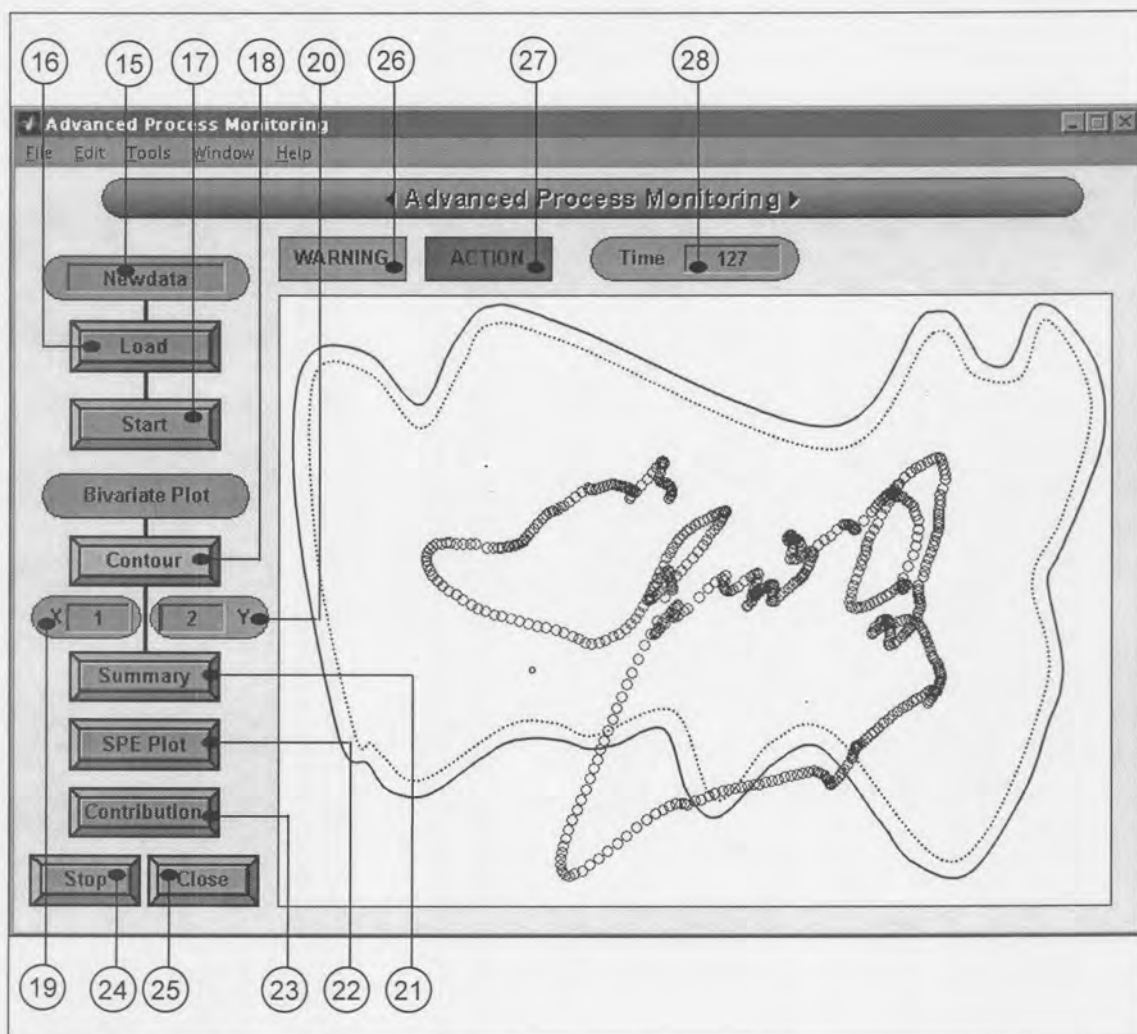


Figure 10.8. NLMSPCA Monitor interface

This interface shown in Figure 10.8 allows the user to choose between the following plots:

- Bivariate contour plot
- Bivariate summary plot
- SPE plot
- Contribution plot

Only one of these plots can be viewed at a time. One would normally use the bivariate summary plot and only view the SPE and contribution plots when an abnormal operation is detected. While viewing the bivariate summary plot, the SPE is also calculated. In an event of an abnormal operation first being detected by the SPE the SPE plot will automatically replace the bivariate summary or contour plot.

Figure 10.8 Tags:

15. Name of the variable containing the new data for investigation purposes. The NLMSPCA model will be applied to this data in order to detect the existence of abnormal behavior in the data. This variable must reside in the matlab workspace and contain the data of each variable in a separate column.
16. Load the data into the NLMSPCA model.
17. Start the NLMSPCA monitoring process using the parameters selected during the setup/training of the NLMSPCA model with normal data.
18. Bivariate contour plot. Only one bivariate contour plot can be plotted at a time. By default, principal component one is plotted versus principal component two. Other combinations can be selected using 5 and 6. If dataset 1 contains four principal components and dataset contains six, principal component one to four will refer to dataset 1 and five to ten to dataset 2.
19. Select principal component number for the x-axes.
20. Select principal component number for the y-axes.
21. Summary bivariate plot. All possible combinations for dataset 1 and dataset 2 are plotted.
22. View the SPE-plot.
23. View the contribution plot.
24. Stop the monitoring process.
25. Close the current window (exit the monitor interface).
26. Warning alarm. This alarm is shown as soon as the warning limits of the bivariate or SPE plots are violated. This alarm will remain for three time intervals before being cleared automatically.
27. Action alarm. This alarm is shown as soon as the action limits of the bivariate or SPE plots are violated. This alarm will remain for three time intervals before being cleared automatically.

28. The current time-interval.

10.12.2. EXPERIMENTAL DATA

First a calibration model was developed based upon 900 one-minute samples taken of the eight monitored variables according to the preceding chapters. To test the new methodology, data representing abnormal operation had to be collected. Figure 10.9 shows typical variable traces taken of 830 process data points at one minute intervals and gives a plot of the nonconforming or abnormal operation data used in the assessment and validation of the methodology.

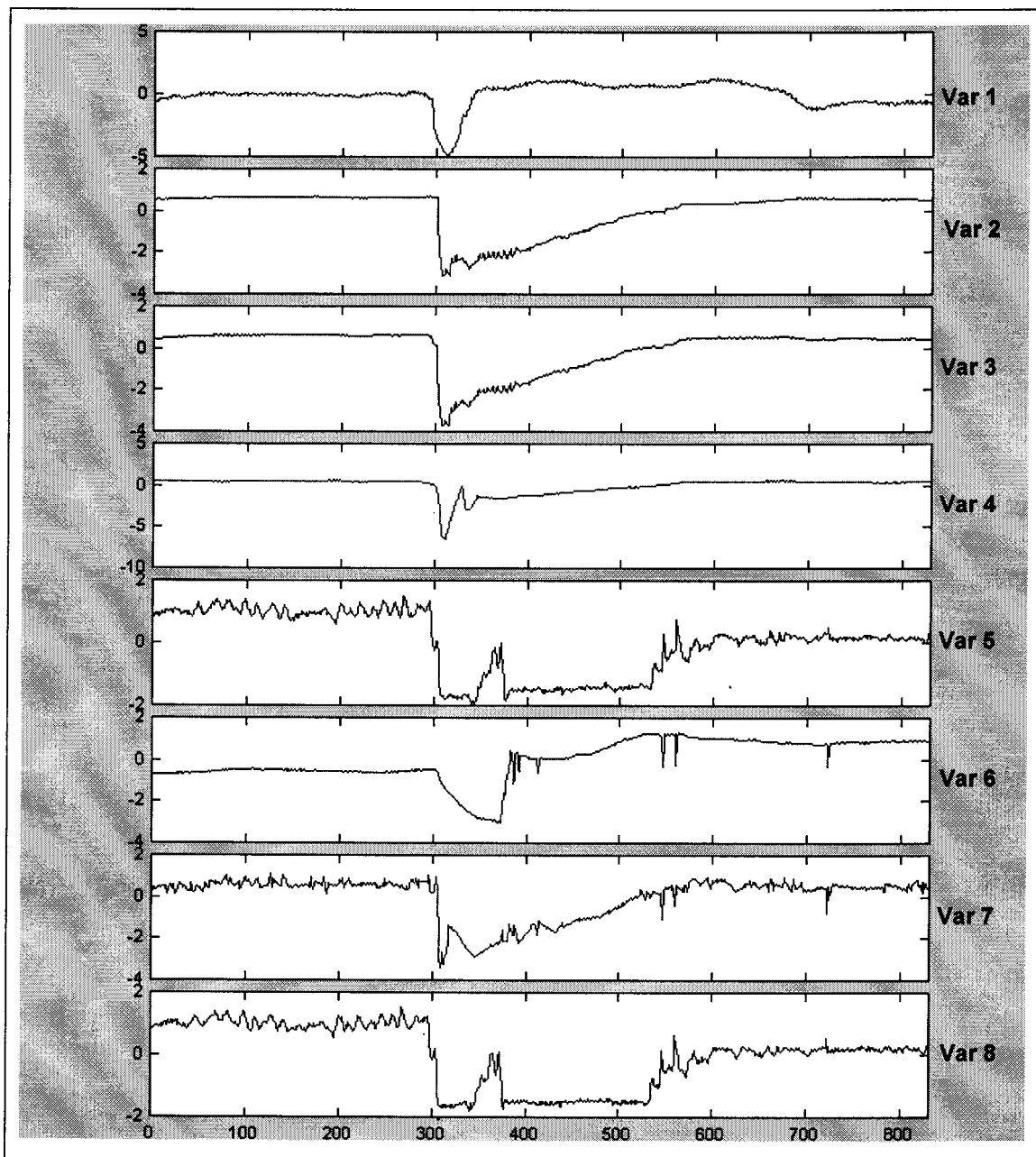


Figure 10.9. Data representing abnormal operation

10.12.3. EVALUATION OF COMPARED MODELS

The aim is not just to show that the algorithm works, but also to satisfy the 3rd stated objective in terms of the critical assessment and validation of the methodology so that the comparative advantages that each of the elements of the NLMSPCA approach offers, becomes clear. This is the main objective of this section. It sets a standard to which the NLMSPCA methodology can be compared. In the end this should enable one to answer the question: How much better is the NLMSPCA approach?

The two methodologies that follow were assessed using the set of unseen data in Figure 10.9 through both the SPE and principal component scores plot. In both cases the action and warning limits were calculated using kernel density estimation.

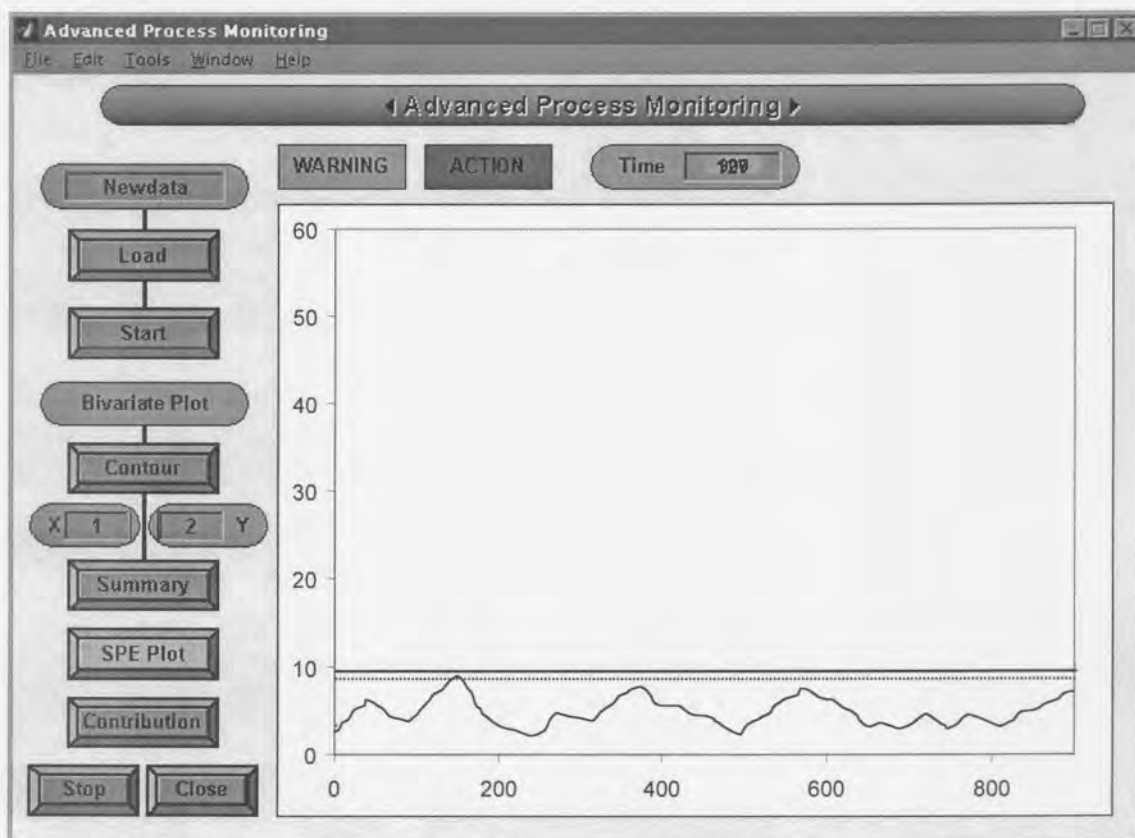


Figure 10.10. SPE plots for the test data based on the 4-3 LMSPCA model with 95% and 99% non-parametric limits

In Figure 10.10 and Figure 10.11 a similar methodology to NLMSPCA was applied except that, instead of using NLPCA, a classical linear PCA was used. The whole process of using neural networks was thus omitted. This will be referred to as the LMSPCA methodology. Using linear principal scores results in different action and warning limits as compared to nonlinear principal scores. Furthermore, using LPCA resulted in six and five principal components to be retained for dataset 1 and dataset 2 respectively to describe the same degree of variability, compared to four and three in the case of NLPCA. As can be seen from the results, this methodology was unable to

effectively detect the point of nonconforming operation. This is to be expected, since the data used was highly nonlinear.

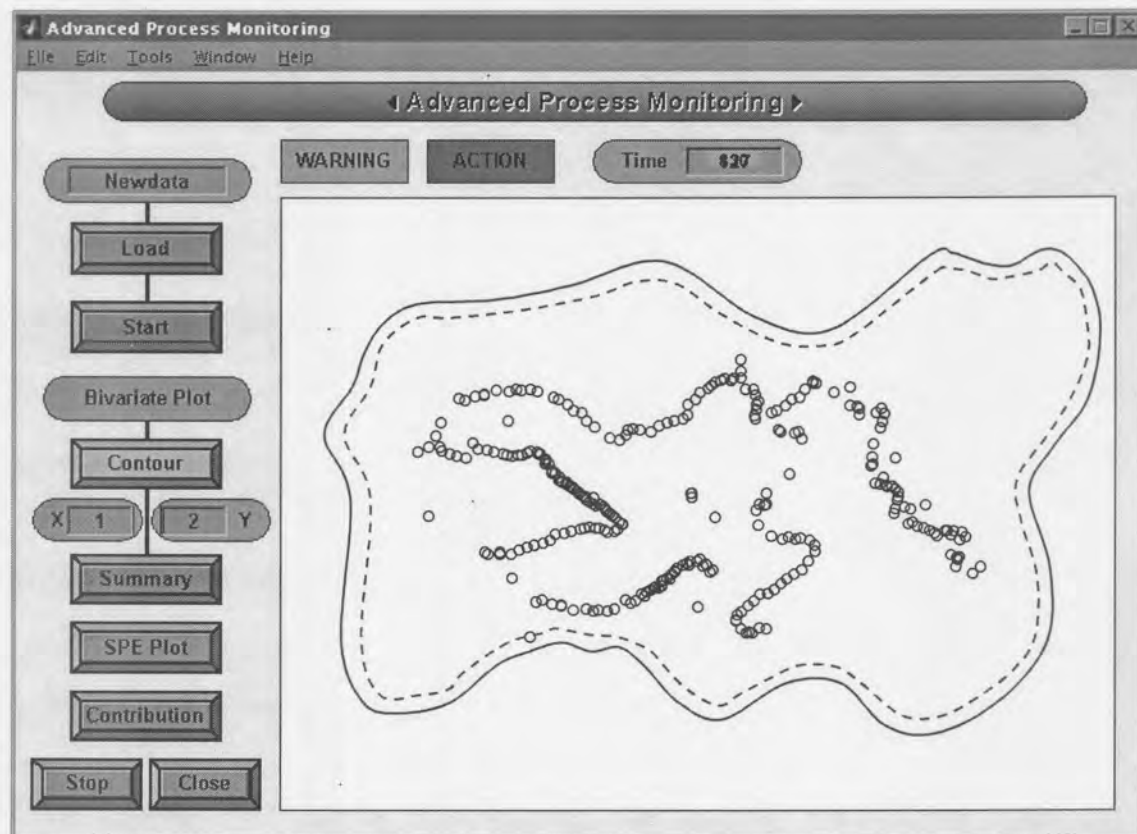


Figure 10.11. Scores plots for the non-conforming test data based on the 4-3 LMSPCA-model.

In the next exercise, again the same methodology was applied to the data in Figure 10.9 except that, instead of using a multiscale methodology, a singlescale methodology was used. This will be referred to as the NLPCA methodology. The processes of multiresolution analysis and wavelet thresholding were thus omitted. This also resulted in only one dataset to be used instead of two. Figure 10.12 and Figure 10.13 gives the results after applying this methodology. As can be seen, it was able to detect the point of nonconforming operation three time intervals earlier than the current alarm system. However, it continued giving false alarms after the process returned to normal operation.

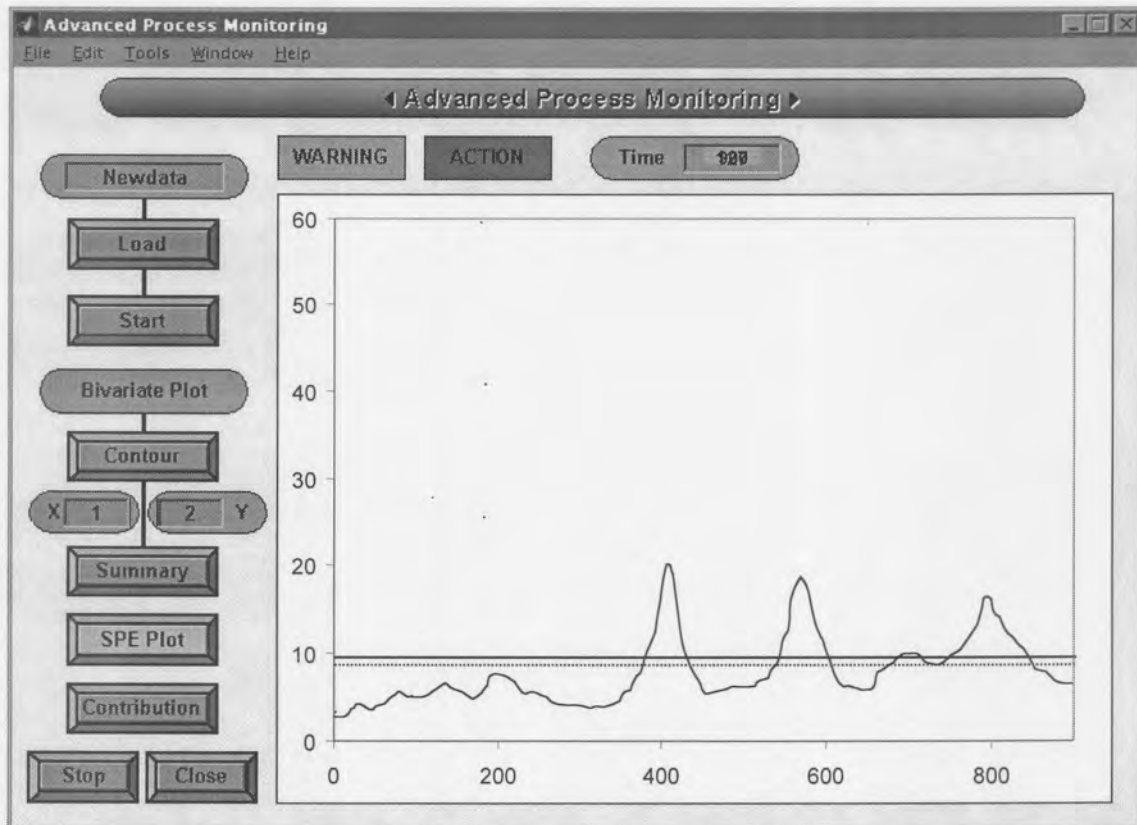


Figure 10.12. SPE plots for the test data based on the 4-3 NLPCA model with 95% and 99% non-parametric limits

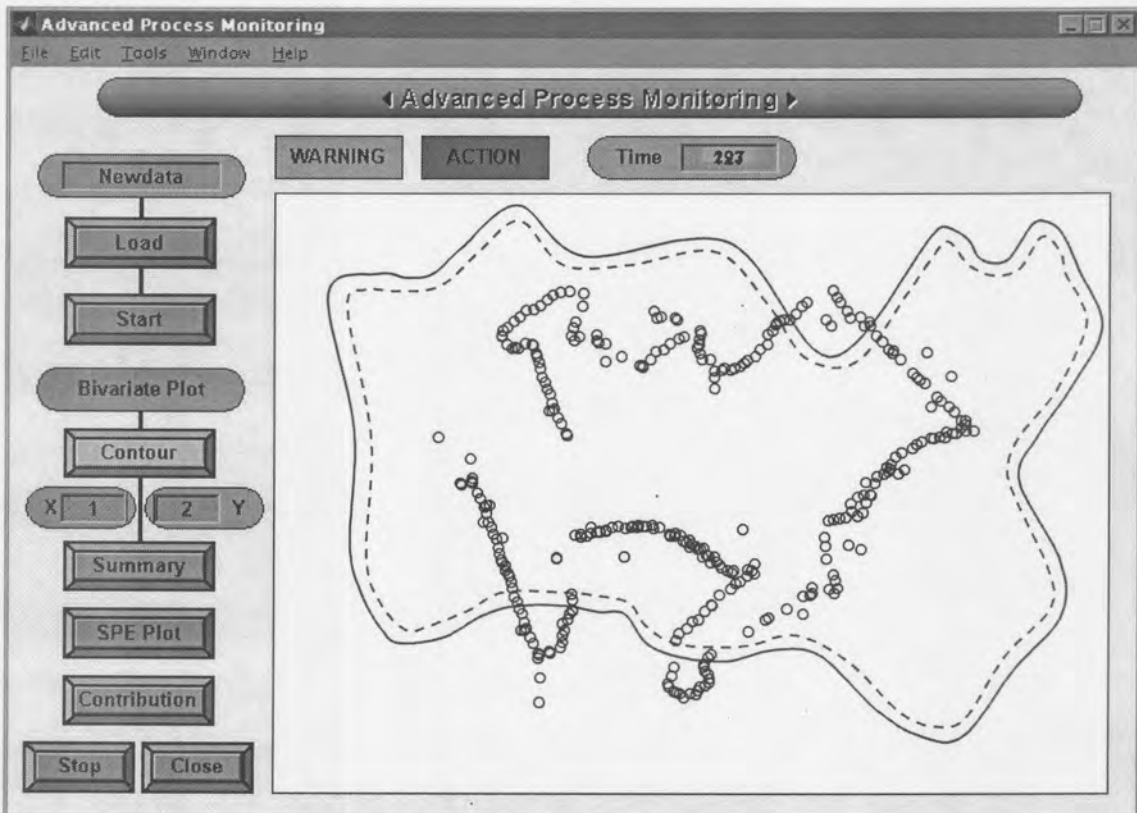


Figure 10.13. Scores plots for the non-conforming test data based on the 4-3 NLPCA-model.

10.12.4. EXPERIMENTAL

In the industrial application indications of failure are difficult to identify, due to the large number of monitored variables, large interactions and nonlinearities as illustrated through Figure 10.10 and Figure 10.11. Current alarm limits as well as the number and type of alarms also have a tendency to conceal the development of abnormal situations. In some cases, for the process under investigation, what was thought to be a failure mode turned out to be a false alarm with no evidence of failure as was partly illustrated in Figure 10.12 and Figure 10.13. It is under these circumstances that one can appreciate a process monitoring scheme, like the one developed here, that is able to overcome these problems and limitations.

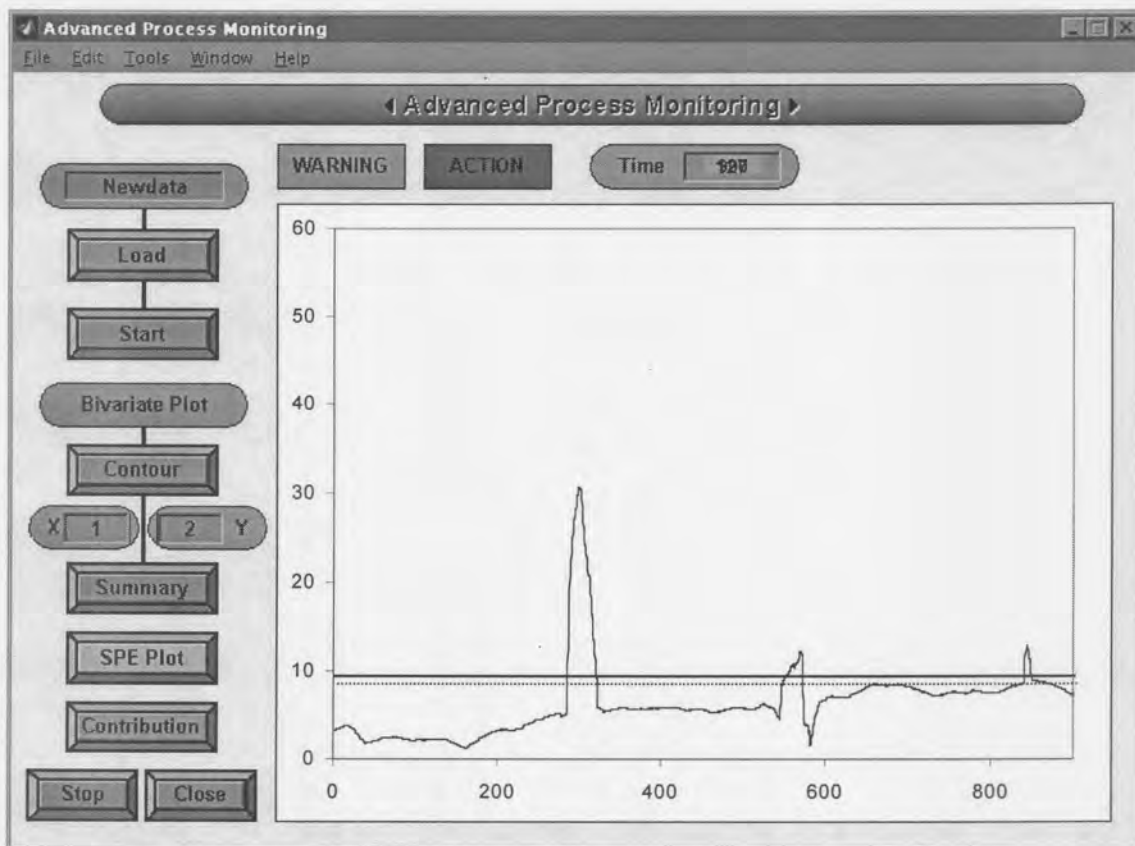


Figure 10.14. SPE plots for the test data based on the 4-3 NLMSPCA model with 95% and 99% non-parametric limits

The nonlinear multiscale PCA scheme introduced in Chapter 8 and 9 was assessed using the set of unseen data in Figure 10.9 through both the SPE and nonlinear principal component scores plot. Figure 10.14 illustrates the results for the SPE for the nonconforming test data set. Also shown are the action and warning limits calculated using kernel density estimation. In this application, the non-parametric control limits are wider than the corresponding limits calculated based upon the assumption of normality. Figure 10.15 shows the nonlinear bivariate scores plot of principal component one

versus two of dataset 1 of the test data with non-parametric control limits, indicating 39 points violating the action limits with the first indication of nonconformance at sample 289.

Figure 10.12 shows the summary plot of the bivariate scores plots for dataset 1 and dataset 2 of the testdata. It can be seen that the process disturbance could be identified from the SPE and bivariate scores plots. The advanced monitoring system was able to detect the process disturbance seven time intervals earlier than the current alarm system. The number of false alarms are also reduced. This illustrates its superiority over the methodologies in Section 10.12.3 and can thus be assumed to be a better methodology.

After a process deviation is identified, the next step is to investigate the cause. In Figure 10.17, a differential contribution plot for non-linear principal component two of dataset 1 and a residual contribution plot were calculated for sample 289, respectively. From both figures, process variable two has the largest contribution, therefore reflecting the possible cause of the process deviation, which gave a positive indication in this case.

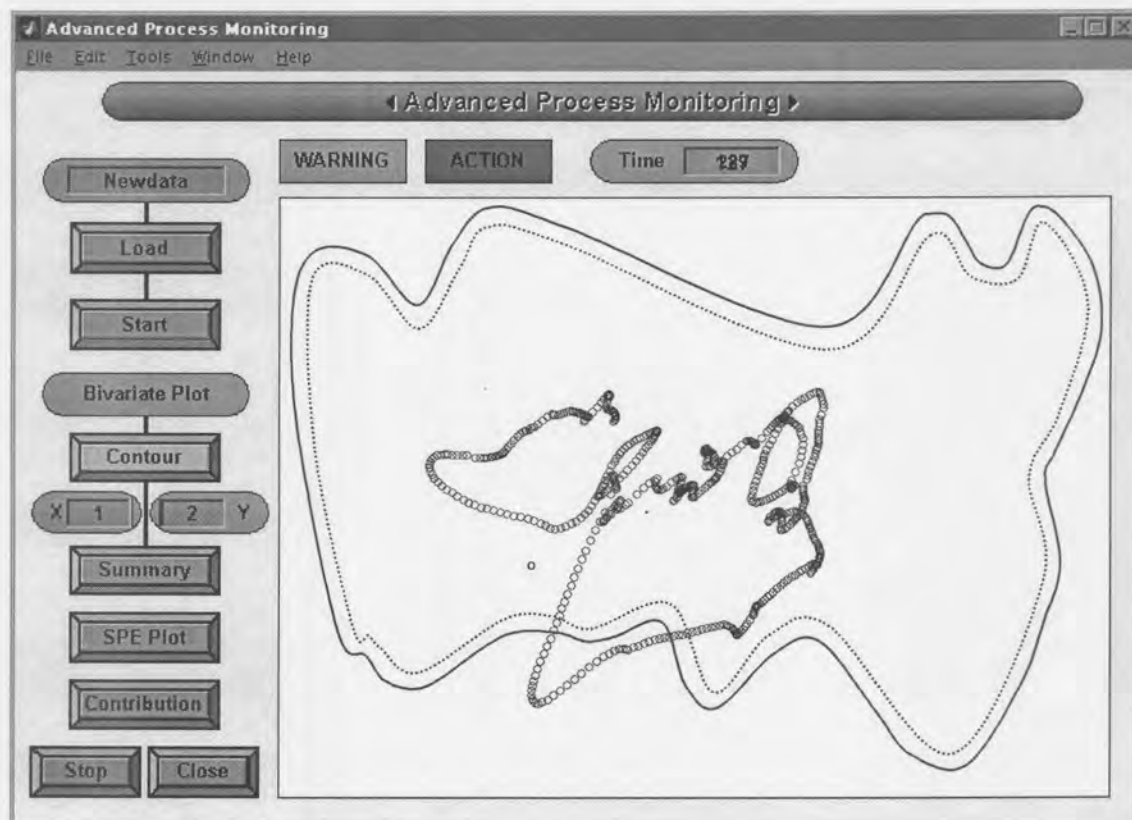


Figure 10.15. Scores plots for the non-conforming test data based on the 4-3 NLMSPCA-model.

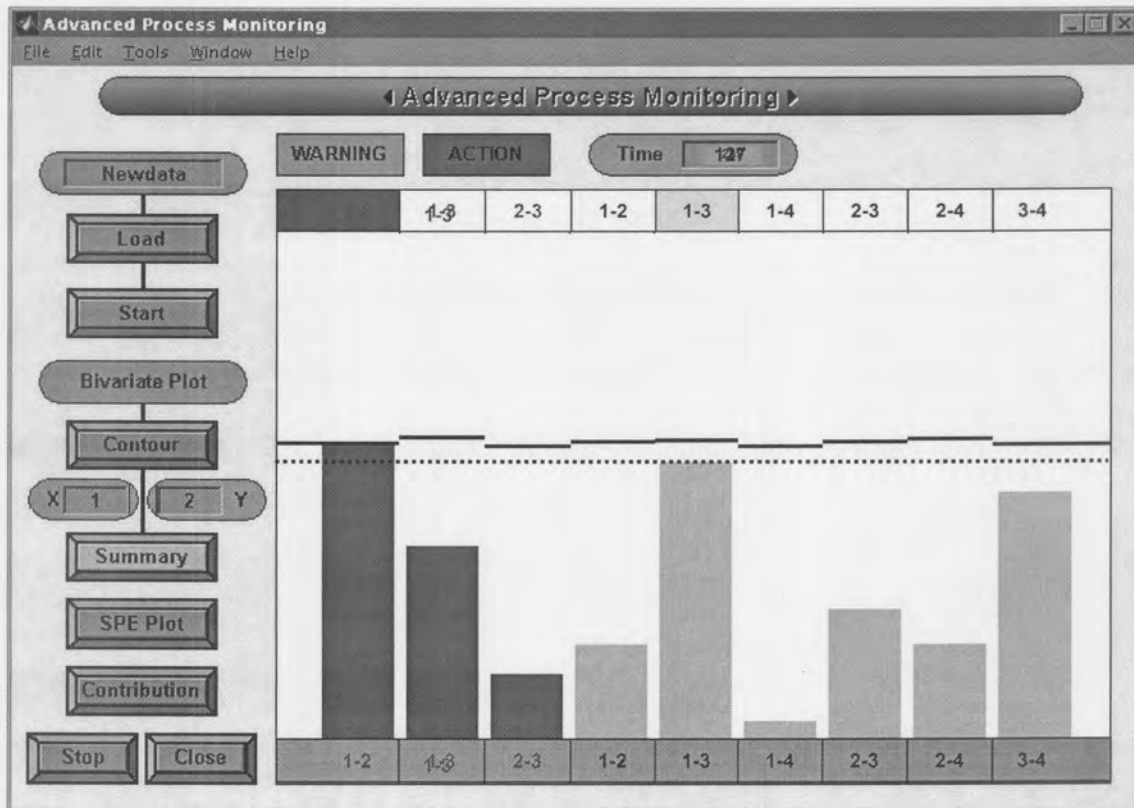


Figure 10.16. Summary plot of the bivariate scores plots based on the 4-3 NLMSPCA-model

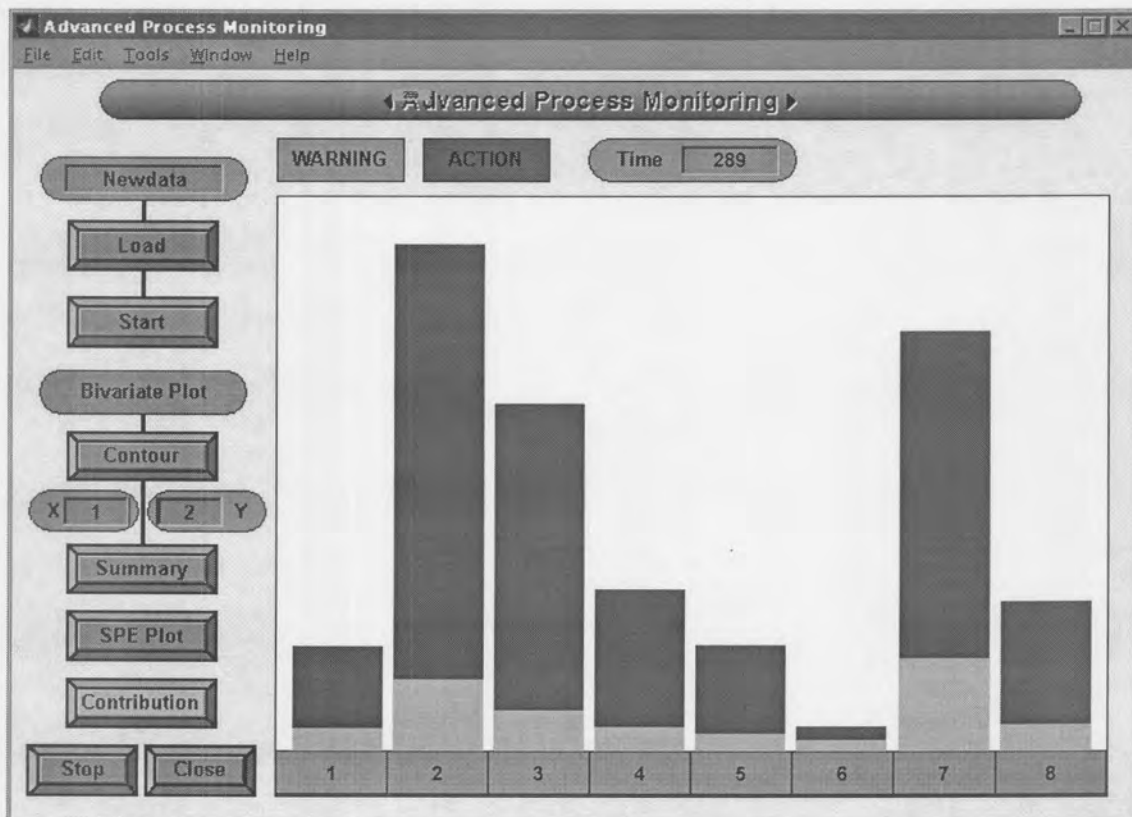


Figure 10.17. Differential () and residual () contribution plots to investigate the cause of process deviation in the non-conforming data

From the differential contribution plot, it is interesting to observe that not only is variable two flagged up, but a large number of other variables appear to contribute to the out-of-control signal. This scenario has been discussed by a number of researchers including Dunia et al. (1996) and Tong and Crowe (1995) and has been identified to be due to the contribution analysis to the SPE being based upon reconstruction. As a consequence, the effect of the changes in the original set of non-conforming variables can propagate to other variable estimates, increasing the chance of erroneous identification. In this respect, interrogation of the contribution plot for the non-linear principal component score is more reliable for the diagnosis step than the linear case. From this industrial application, it can be seen that the nonlinear multiscale PCA model achieves good fault detection results which is also better than using classical LPCA or singlescale analysis.

11.1. Summary

In this study an online multiscale nonlinear PCA approach was derived for process monitoring and fault detection and its performance validated on unseen test data from a nonlinear industrial process. The advantage of this method is that both linear and nonlinear correlations can be extracted from the process data to obtain a more parsimonious description of the original data. The data was first decomposed into different levels of detail and approximations through multilevel wavelet decomposition. Heavy high-frequency noise and sharp data spikes in the industrial data sets were then eliminated through wavelet thresholding. The thresholded level coefficient vectors that contained important information were reconstructed to form reconstructed details and approximations containing the most important information of the data at different levels. Thus, in applying the discrete wavelet transform the underlying process trend was preserved in the approximation and detail coefficients. This was then used to develop a linear and nonlinear principal component model. All the scales were also combined for deriving a combined principal component model. Using wavelet coefficients in the derivation of the nonlinear PCA model significantly reduces the computational burden without impacting upon the predictive ability of the process representation. Moreover, the possibility of the input-training network to overfit the data is greatly reduced and the generalisation properties of the network enhanced. Fortunately, the last MSPCA steps of selecting the scales that indicate significant events, reconstructing the signal to the multilevel time domain, and computing the scores and residuals for both the thresholded and non-thresholded reconstructed signals, improve the speed of detecting abnormal operation and eliminate false alarms after a process returns to normal operation. Using the multilevel methodology also greatly enhances the ability of the monitoring system to detect different types of abnormal conditions. Data-driven, nonlinear control limits and modified contribution plots were derived to facilitate the comprehensive and robust monitoring and fault detection.

The results of the application of the conjunction of the multilevel wavelet decomposition, wavelet thresholding technique and nonlinear PCA algorithm to an industrial process demonstrates the advanced performance for fault detection and isolation. According to the results it should be possible to determine the development of an abnormal situation in the steam distribution system early enough in order to reduce the consequences of the abnormal event. Here the methodology was only applied to one specific case. The accuracy and reliability of the methodology needs to be validated on more scenarios.

Keeping the process in mind it should be clear at this stage that it is not yet possible to apply this methodology in real time since the factory currently lacks the infrastructure. At this stage it is not possible to access or monitor all the variables throughout the factory from a single point. It is only possible to access a specific unit's variables from that unit's control room. However, this infrastructure will be implemented over the next two years. The only way to currently gain access to all the variables from a single point is through the

History Module which is a central database. This database only saves data at a minimum sampling rate of one minute and access to the database is not very reliable.

Apart from this industrial application, it can be seen that the NLMSPCA model achieves good fault detection results. Moreover, the non-parametric control limits are statistically more valid for non-linear, on-line, process performance monitoring.

The use of the summarised plots allows enhanced global visualisation capabilities and interpretation and reduces the space taken up by conventional multivariate statistical plots. It can be concluded that the advanced monitoring system architecture is qualified for further development.

11.2. Further development

The next step would involve the development of an application to create plans to recover from malfunctions and threatened goals. It should be capable of replanning in real time, and use knowledge of the process represented in blackboards to carry out planning without the need for human planners to exhaustively explore every possible scenario. This function should assess the success of the plan and be capable of closed loop control of the process if so authorized.

11.3. Practical implications

In process monitoring and fault detection the major issue becomes that of practical implications.

Common to all approaches described as intelligent fault detection, is that they derive or synthesize higher order statements about the plant from lower order information, e.g. process measurements, event information, alarms. They must all be seen as add-ons which complement an existing good quality basic process alarm system. They will produce results if the basic information system is sound. None of the approaches will cure fundamental faults in the basic alarm system, and should not be considered as doing so.

The major problem with all the computerized fault detection techniques is that, even with sufficient implementation tools, they require considerable engineering analysis of plant behavior. Some of this can be done 'on paper' from the plant design information, but generally considerable post-commissioning tuning is also required. Applying these techniques also demands some 'failure mode analysis' to be performed to ensure missing or incorrect input data not causing false conclusions. Questions also remain with artificial intelligence and expert systems techniques

about demonstrating that a procedure that is developed on a limited range of plant transients will be effective in unexpected situation.

The development and application of this technique would require specialist knowledge. Unfortunately the reports of large scale practical applications on working plants are few and far between. Priority should be given to applying other more basic methods to eliminate the simple problems, and only then to invest in the more advanced methods. So why develop more advanced technology? By the time the more basic problems have been addressed, advanced methods like the one developed here should be ready for application since the experimental stage, and especially the period up to general acceptance and reliability, for new and advanced technology is much longer.

APPENDIX A

This table lists the variables contained the database *data_base.mat* together with a description of each variable.

Matlab Variable	Variable Description
i	Wavelet level number
j	Variable number
x	Dataset number
bipl_index_lev_i	Biplot level index
demap_b1_lev_i	First layer bias values for the demapping neural network
demap_b2_lev_i	Second layer bias values for the demapping neural network
demap_f1_lev_i	First layer transfer function for the demapping neural network
demap_f2_lev_i	Second layer transfer function for the demapping neural network
demap_w1_lev_i	First layer weights for the demapping neural network
demap_w2_lev_i	Second layer weights for the demapping neural network
ddeg_var_j	Daubechies degree applied to each variable during wavelet analysis
filt_var_j	Reconstructed, thresholded signal
FSr_level_i	Reduced linear principal component scores
lpca_l_lev_i	Eigenvalues of covariance matrix
lpca_lr_lev_i	Reduced eigenvalues of covariance matrix
level_thwc_i	Thresholded wavelet coefficients
lpca_scr_lev_i	Linear principal component scores
lpca_sumry_lev_i	1 st column: Proportion of total variability accounted for by each pc 2 nd column: Cumulative variability
lpca_u_lev_i	Linear principal loading (eigenvectors of the covariance matrix)
lpca_ur_lev_i	Reduced principal loadings
map_b1_lev_i	First layer bias values for the mapping neural network
map_b2_lev_i	Second layer bias values for the mapping neural network
map_f1_lev_i	First layer transfer function for the mapping neural network
map_f2_lev_i	Second layer transfer function for the mapping neural network
map_w1_lev_i	First layer weights for the mapping neural network
map_w2_lev_i	Second layer weights for the mapping neural network
mra_var_j	Multilevel reconstructed signal based on wavelet coefficients
mwsizе_var_j	Maximum window size for wavelet analysis
newdata	New data on which to test the final NLMSPCA model
newdatan	Normalized new data
newdatas	Standardized new data
nlpca_b1_lev_i	First layer bias values for the input training neural network
nlpca_b2_lev_i	Second layer bias values for the input training neural network
nlpca_f1_lev_i	First layer transfer function for the input training neural network
nlpca_f2_lev_i	Second layer transfer function for the input training neural network
nlpca_scr_lev_i	Nonlinear principal components generated from the input training neural network
nlpca_w1_lev_i	First layer weights for the input training neural network
nlpca_w2_lev_i	Second layer weights for the input training neural network
num_var	Total number of variables
Num_pcs_x	Number of principal components to retain
orig_var_j	Original variables before normalization or standardization

APPENDIX A

testdata	Validation data
testdatan	Normalized validation data
testdatas	Standardized validation data
th_level_i	Thresholded wavelet coefficients for each level
thmra_var_j	Multilevel reconstructed signal based on thresholded wavelet coefficients
thtype_var_j	Threshold type applied to each variable
traindata	Training data used to train the NLMSPCA model
traindatan	Normalized training data
traindatas	Standardized training data
Ur_level_i	Reduced linear principal component loadings
wlevels	Total number of wavelet decomposition levels
wlevs_var_1	Number of wavelet decomposition levels applied to each variable



APPENDIX B

This appendix gives a short description of how the interfaces were created if someone is interested in creating similar interfaces.

The following procedure creates the basic interface template on which to add buttons, text boxes, etc.:

1. Create the background using Microsoft PowerPoint.
2. After creating the background, run the background as a PowerPoint presentation, filling the whole screen.
3. Capture the screen by pressing the *Alt* and *Print Screen* keys on the keyboard simultaneously.
4. Past the image in a picture editor, in this case Microsoft Picture Editor.
5. Cut out the part of the image that is needed and past it as a new image.
6. Save the image as a bitmap file.
7. From the Matlab prompt change to the directory containing your bitmap image.

```
>> cd c:\asm\interfaces\images
```

8. Type the following command at the Matlab prompt

```
>> [a, b] = imread('my_image.bmp');
```

```
>> image(a);
```




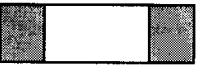
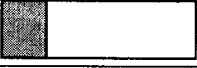

```
>> guide
```

This will display the image as a new figure and activate Guide Control Panel which is the graphical user interface editor. The figure can be sized and scaled to preference and is ready to be used as a background. Buttons, edit boxes, etc. can be added on top of the image.

9. The best way to create such an interface is to open an existing interface from the Matlab command window and to type

```
>> guide
```


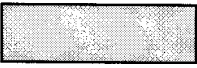

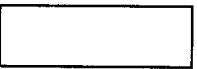
which will launch the Guide Control Panel. Use the Property Editor and Callback Editor to view the different properties and their values.

Colour Code	Description
	Background (bitmap)
	Push button
	Edit box
	Slider
	Popup menu
	Axis

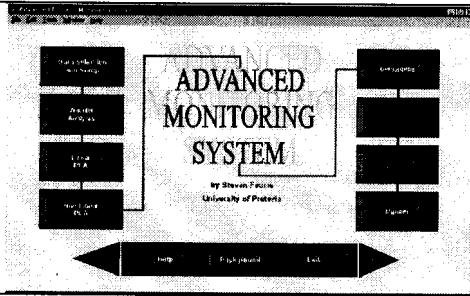

APPENDIX B

A few examples are provided to illustrate the most important parameters and their values.



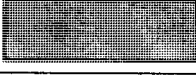
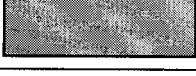
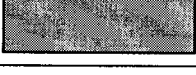
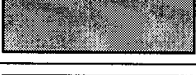

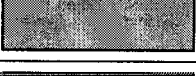


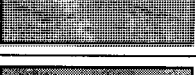

Database_fig

		Callback	Other	Function
FileName : 'C:\ASM\monitor\Interfaces\setup\database_fig.m' Tag : dbfig Name : Database Setup			NumberTitle: 'Off' Resize: 'Off'	
	psuccess.bmp			Interface background
Axes			'Ytick','[]', 'Xtick','[]', 'Layer','Bottom'	
	Db_push_01	dbsetup	'String','OK'	Accept and advance to next interface
	Db_edit_01		'String','data_base'	Specify name of database

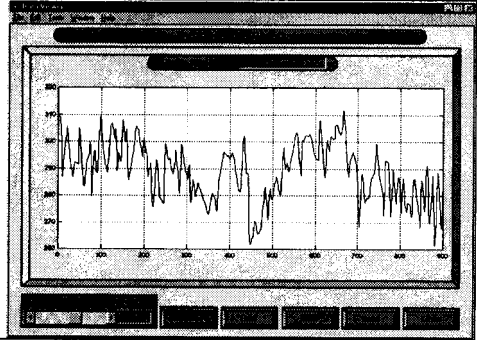

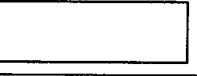
Int_main

		Callback	Other	Function
FileName : 'C:\ASM\monitor\Interfaces\int_main.m' Tag : int_01 Name : Advanced Monitoring System				
	Int_main.bmp			Interface background
Axes			'Ytick','[]', 'Xtick','[]', 'Layer','Bottom'	
Figure			'Name','Advanced Process Monitoring System' 'NumberTitle','Off', 'Resize','Off'	


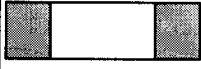


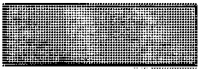


APPENDIX B

	int_01_start_01	int_data	CData : a [a,b]=imread('button 01.bmp');	Data selection and Setup
	int_01_start_02	Int_wave	CData : a [a,b]=imread('button 02.bmp');	Wavelet Analysis
	int_01_start_03	Int_lpca_main	CData : a [a,b]=imread('button 03.bmp');	LPCA
	int_01_start_04	Fig_05_main	CData : a [a,b]=imread('button 04.bmp');	NLPCA
	int_01_start_05		CData : a [a,b]=imread('button 05.bmp');	Demapping
	int_01_start_06		CData : a [a,b]=imread(bck_m arb.bmp');	
	int_01_start_07		CData : a [a,b]=imread(bck_m arb.bmp');	
	int_01_start_08		CData : a [a,b]=imread('button 08.bmp');	Monitoring
	Int_01_help_01		CData : a [a,b]=imread(bck_m arb.bmp');	
	int_01_bckg_01	Bckgr_01_01	CData : a [a,b]=imread(bck_m arb.bmp');	Background
	int_01_exit_01	Int_exit	CData : a [a,b]=imread(bck_m arb.bmp');	Exit
	int_01_next_01	int_data	CData : bck_marb.bmp	Next

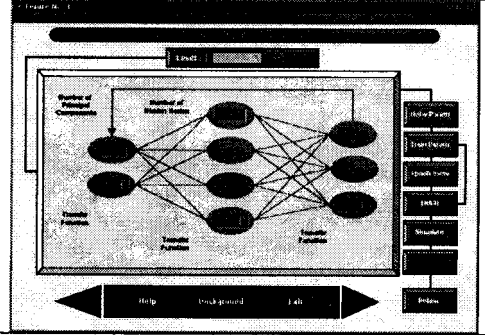
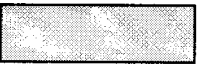
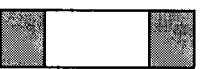




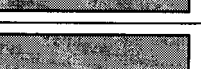
View_01a

	Callback	Other	Function
Name : Data Viewer Filename : c:\asm\monitor\programs\data\view_01a.m Tag : viewer_01a			
	View_01a.bmp		Background image
	view_01a_edit_02		Matlab variable containing data

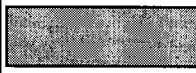


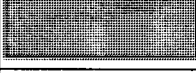

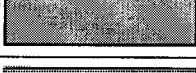



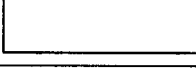

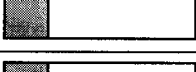
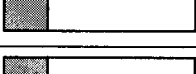

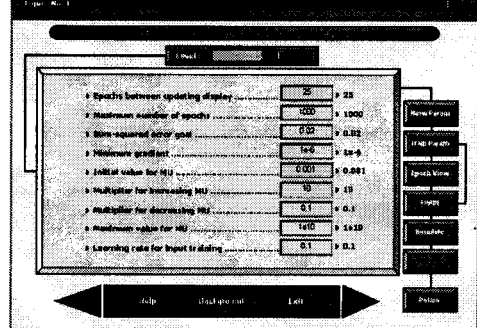
APPENDIX B

	view_01a_plot_01		Layer : top	
	view_01a_slid_01	plot_01a	Max Min SliderStep Value	
	view_01a_edit_01	int_main	String	
	view_01a_grid_01	grid_on_off	String	Grid on/off
	view_01a_hold_01	hold_on_off	String	Hold On/off
	view_01a_hold_01			Help
	view_01a_close_01			Close

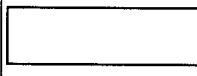
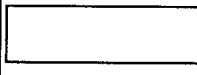
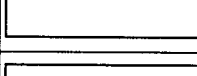
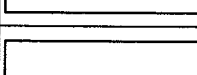
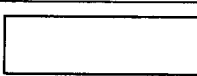
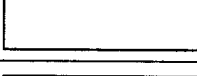




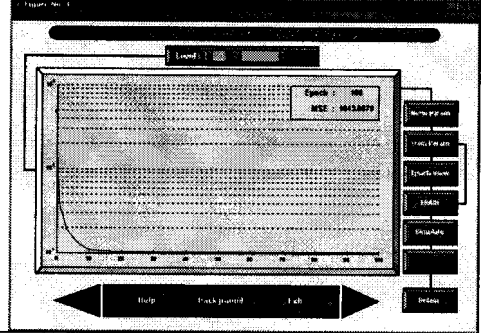





Fig_05_main

		Callback	Other	Function
MAIN				
				Interface background
	fig_05m_pop_01			Specify wavelet level to analyze
	fig_05m_edit_01			Display slider value
	fig_05m_push_01	Fig_05_push('net w')		Network parameters Display layer 1
	fig_05m_push_02	Fig_05_push('train')		Training parameters
	fig_05m_push_03	Fig_05_push('epoch')		Epoch view
	fig_05m_push_04			Train network

APPENDIX B

	fig_05m_push_05			Simulate network and plot comparison and error
	fig_05m_push_06			
	fig_05m_push_07			Retain network parameters if results are satisfactory
				Previous step (lpca)
				Help
				Background
				Exit
				Next step (demapping)
LAYER 1				
	Fig_05m_edit_03			Number of PC's
	Fig_05m_edit_02			Number of hidden nodes
	Fig_05m_text_01			Number of variables
	Fig_05m_pop_01			Input layer transfer function
	Fig_05m_pop_02			Hidden layer transfer function
	Fig_05m_pop_03			Output layer transfer function
				
LAYER 2	Fig_05_a			

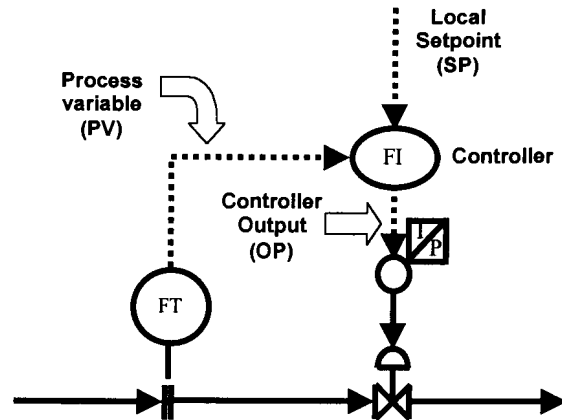
APPENDIX B

	Fig_05a_edit_01		'Visible','Off'	Epochs between updating display
	Fig_05a_edit_02		'Visible','Off'	Maximum number of epochs to train
	Fig_05a_edit_03		'Visible','Off'	Sum-squared error goal
	Fig_05a_edit_04		'Visible','Off'	Minimum gradient
	Fig_05a_edit_05		'Visible','Off'	Initial value for mu
	Fig_05a_edit_06		'Visible','Off'	Multiplier of increasing mu
	Fig_05a_edit_07		'Visible','Off'	Multiplier of decreasing mu
	Fig_05a_edit_08		'Visible','Off'	Maximum value for mu
	Fig_05a_edit_09		'Visible','Off'	Learning rate for input training
	Fig_05a_image_01		'Visible','Off'	
				
LAYER 3	Fig_05_b		'Visible','Off'	
	Fig_05b_line_01		'Visible','Off'	
	Fig_05b_text_04		'Visible','Off'	Epoch string
	Fig_05b_text_02		'Visible','Off'	Display epoch number
	Fig_05b_text_03		'Visible','Off'	Error string
	Fig_05b_text_01		'Visible','Off'	Display SSE

APPENDIX C

The purpose of the taglist is as follows:

- It gives a summary of all the variables (total number and type) that had to be considered when selecting the most important variables for modeling purposes.
- It gives an indication of the signal source which can be one of the following:
 - OP = controller output signal
 - SP = set point
 - PV = process variable
 - CR, CK, CS = calculated



- The tag number gives an indication of the type of signal, i.e. 36P1001D.OP is a pressure signal.
- The tag number also gives the process unit where the signal is generated, 36P1001D.OP indicates that the signal is generated in unit 36. If all the tags are viewed it can be gathered that variables from many process units had to be considered which made it even more difficult since each unit is operated independently.
- The shaded cells contain the variables that were used in the modeling and indicate how they were calculated.

TAGLIST			
NO	TAG	DESCRIPTION	INFO
1	63P1001D.OP	43-8BAR LETDOWN VLV 1	
2	63P1001E.OP	43-8BAR LETDOWN VLV 2	
3	63P1001F.OP	43-8BAR LETDOWN VLV 3	
4	63P1001G.OP	43-8BAR LETDOWN VLV 4	
5	63P1005A.OP	43-4BAR LETDOWN VLV 1	
6	50FIC140.PV	43 BAR STEAM FLOW	Gasification exp
7	50FIC142.PV	43 BAR STEAM FLOW FASE 2	Gasification exp
8	10F0164.PV	HP STEAM HEADER FLOW	
9	10F2564.PV	HP STEAM HEADER FLOW	
10	20F1127.PV	SATURATD HPSTM EXPORT	
11	20F1239.PV	HP SAT STEAM EXPORT	
12	16F1036.PV	ES115 REB STM FL	Phenosolvan exp
13	23F2077.PV	H.P.STM TO REB	Benfield/Cold Sep exp
14	32F1034.PV	HPSU STM TRAIN 1	Cat Cond/LPG Recovery exp
15	32F2034.PV	HPSU STM TRAIN 2	Cat Cond/LPG Recovery exp
16	29F1006.PV	HP STEAM TO ES-101B	Light Oil Fractionation exp

APPENDIX C

17	29F1005.PV	HP STEAM TO ES-101A	Light Oil Fractionation exp
18	37F1023.PV	FIC MEK DEH TWR ST	CWU exp
19	37F1022.PV	FIC ACETONE RECYC	CWU exp
20	15F1028.PV	HPSA STM	Naphta exp
21	35F1108.PV	HPSA STM TO UNIT 35	Distillate HTU exp
22	27F1077.PV	HP STM ES-109	Isomerisation exp
23	14F1003.PV	HPSA STM TO REB	Tar exp
24	14F2003.PV	HPSA STM TO REB	Tar exp
25	14F3003.PV	HPSA STM TO REB	Tar exp
26	14F4003.PV	HPSA STM TO REB	Tar exp
27	71F0126.PV	2500 kPa steam to VL104	Acid Recovery exp
28	71F0202.PV	2500 kPa steam to VL201	Acid Recovery exp
29	71F0209.PV	2500 kPa steam to VL202	Acid Recovery exp
30	71F0218.PV	2500 kPa steam to 71VL203	Acid Recovery exp
31	20F084.CS	REACTION WATER	
32	20F1111.CS	T.T REACT H2O	
33	20F1001A.PV	PURE GAS FEED TRAIN 1	
34	20F2001A.PV	PURE GAS FEED TRAIN 2	
35	20F3001.PV	PURE GAS FEED TRAIN 3	
36	20F4001.PV	PURE GAS FEED TRAIN 4	
37	20F5001.PV	PURE GAS FEED TRAIN 5	
38	20F6001A.PV	PURE GAS FEED TRAIN 6	
39	20F7001A.PV	PURE GAS FEED TRAIN 7	
40	20F8001.PV	PURE GAS FEED TRAIN 8	
41	20F1002A.PV	EXT RECYCLE FEED TRN 1	
42	20F2002.PV	H2 RICH REF.GAS TRAIN 2	
43	20F3002.PV	H2 RICH REF.GAS TR 3	
44	20F4002A.PV	EXT RECYCLE FEED TRN 4	
45	20F5002.PV	H2 RICH REF GAS TRAIN 5	
46	20F6002A.PV	EXT RECYCLE FEED TRN 6	
47	20F7002A.PV	EXT RECYCLE FEED TRN 7	
48	20F8002.PV	H2 RICH REF GAS TRAIN 8	
49	20F0102.PV	REF GAS FLOW TO PLANT	
50	20F1080.PV	TAILGAS PRODUCT TRAIN 1	
51	20F2080.PV	TAILGAS PRODUCT TRAIN 2	
52	20F3080.PV	TAILGAS PRODUCT TRAIN 3	
53	20F4080.PV	TAILGAS PRODUCT TRAIN 4	
54	20F5080.PV	TAILGAS PRODUCT TRAIN 5	
55	20F6080.PV	TAILGAS PRODUCT TRAIN 6	
56	20F7080.PV	TAILGAS PRODUCT TRAIN 7	
57	20F8080.PV	TAILGAS PRODUCT TRAIN 8	
58	12F1003.PV	PURE GAS FLOW TR1	
59	12F2003.PV	PURE GAS FLOW TR2	
60	12F4003.PV	PURE GAS FLOW TR4	
61	12F5003.PV	PURE GAS FLOW TR5	
62	F0F1013A.PV	S3 TO S2 PURE GAS	
63	F0F1013B.PV	S2 TO S3 PURE GAS	

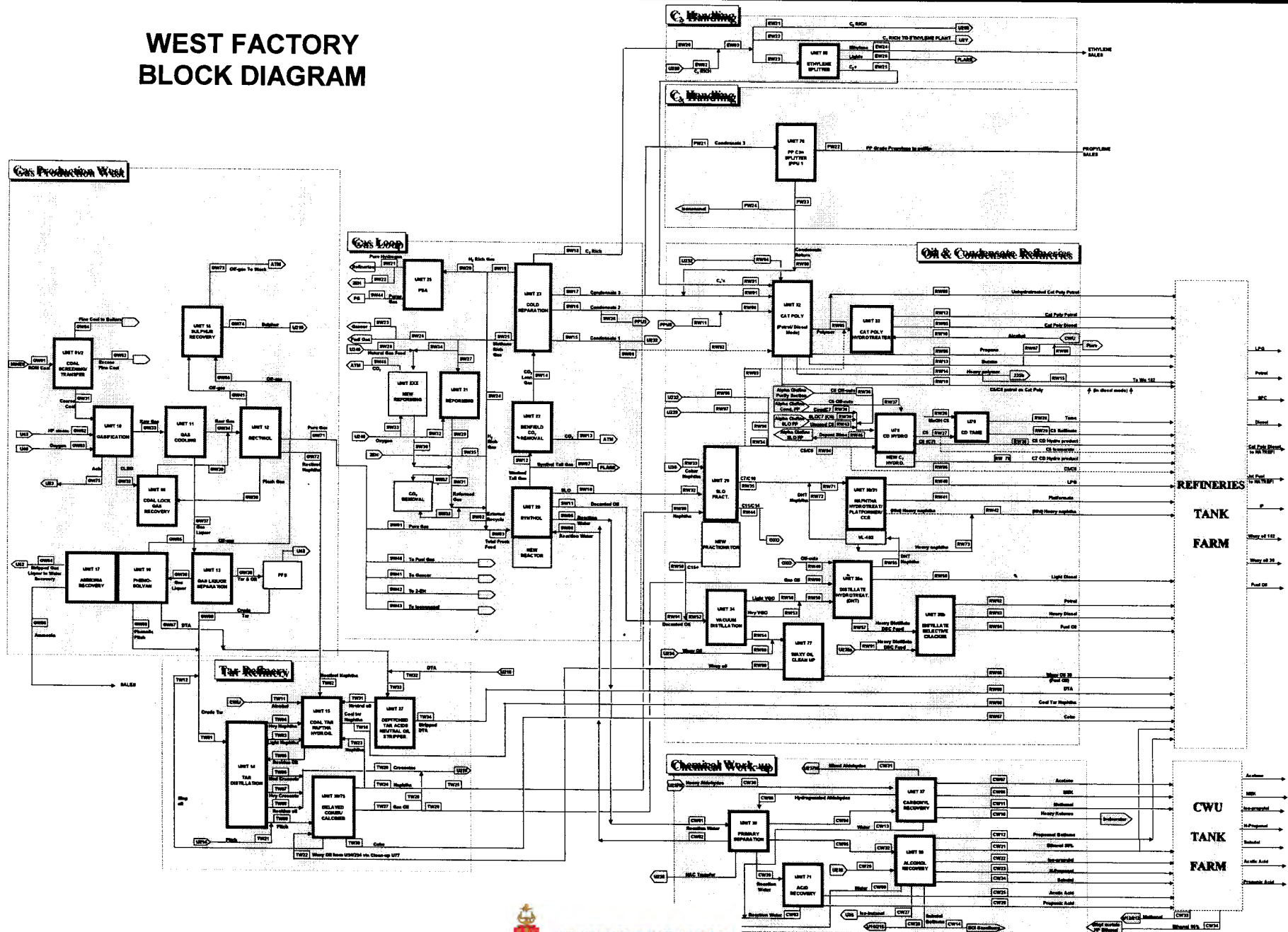


APPENDIX C

64	20F1092.PV	TAIL GAS WASH TWR OVHD A	
65	20F1096.PV	TAIL GAS WASH TWR OVHD B	
66	10K0146.CS	TOT LPTYD.VERGSRS	Total gassifiers on line
67	21K0168.CR	TOTAAL	Total reformers on line
68	20K1001.CR	use for 20K1001A.CR	
69	20K2001.CR	use for 20K1001A.CR	
70	20K3001.CR	RUN STATUS TR3	
71	20K4001.CR	RUN STATUS TR4	
72	20K5001.CR	RUN STATUS TR5	
73	20K6001.CR	RUN STATUS TR6	
74	20K7001.CR	RUN STATUS TR7	
75	20K8001.CR	RS901 on Line	
76	20F0101.CK	TOT SUIWERGAS TO SYNTHOL	
77	20F0102.CK	TOTAL REF GAS +H2	
CALCULATED			
78		Total Rectisol Feed	58+59+60+61+62-63
79		Total PG feed	33+34+35+36+37+38+39+40
80		Total RG Feed	41+42+43+44+45+46+47+48
81		Total FF	79+80
82		Total Tailgas	64+65
83		CFB's on Line	70+72
84		SAS (10.7m) on Line	(68+69+73+74)/[2]
85		SAS (8.0m) on Line	71+75
86		43-8BAR LETDN FLOW VLV1	1*[2.3379]
87		43-8BAR LETDN FLOW VLV2	2*[2.3379]
88		43-8BAR LETDN FLOW VLV3	3*[1.96695]
89		43-8BAR LETDN FLOW VLV4	4*[1.96695]
90		43-4BAR LETDN FLOW	5*[1.96248]
91		Total Steam Consumers	
92		Total Steam Letdn	86+87+88+89+90
93		Total Steam Export (Calc)	91+92
94		Total Steam Export (Meas)	10+11

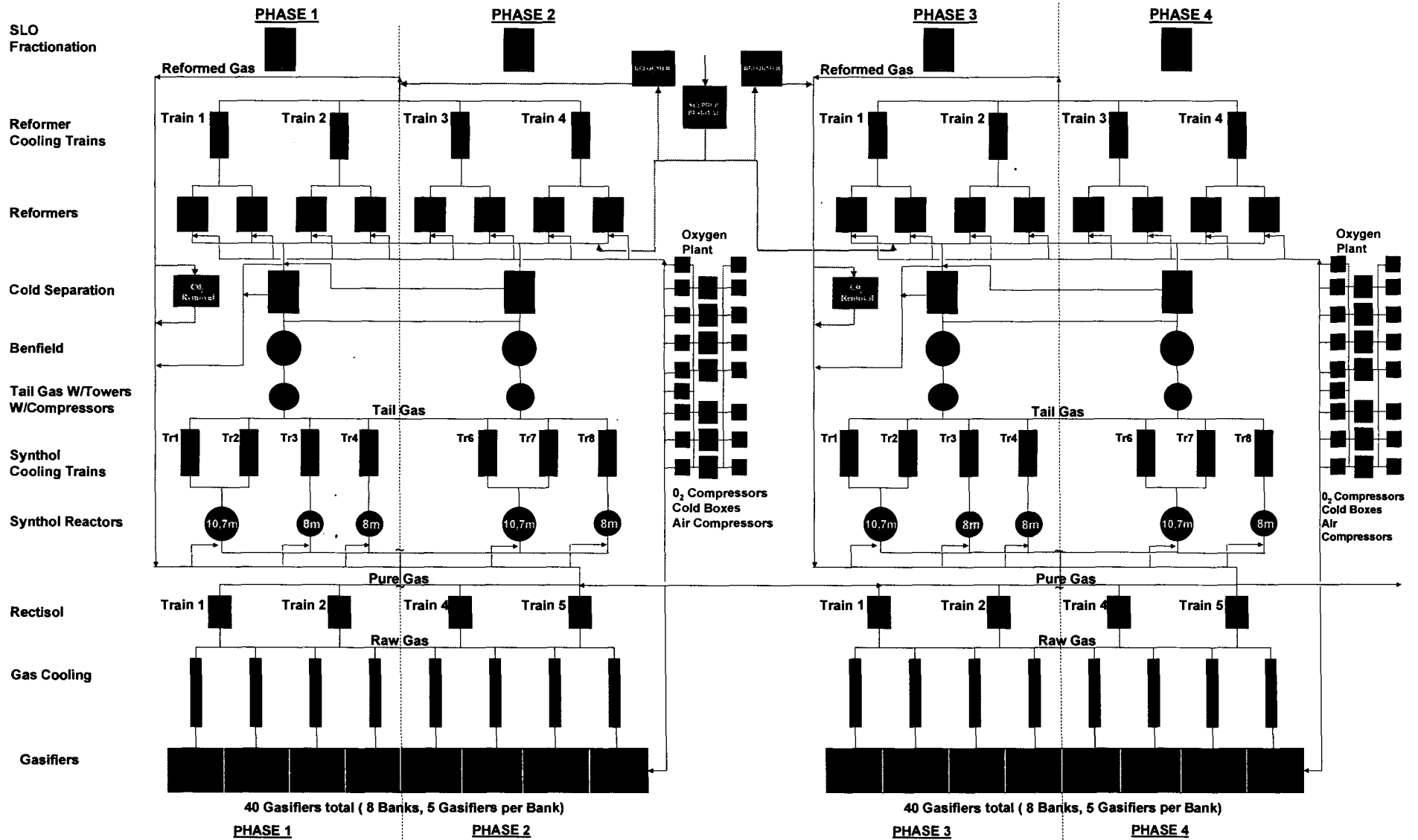


WEST FACTORY BLOCK DIAGRAM



WEST FACTORY PHASES AND TRAINS

EAST FACTORY PHASES AND TRAINS



REFERENCES

- Abbas, H. M., and M. M. Fahmy, "Neural Model for Karhunen Loeve Transformation with Application to Adaptive Image Compression," *Proc. Inst. Elec. Eng., I. Commun. Speech Vision*, **2**, 135 (1993).
- Abnormal Situation Management Consortium Website: www.iac.honeywell.com/Pub/AbSitMang/.
- Ackley, D. H., G. E. Hinton, and T. J. Sejnowski, "A Learning Algorithm for Boltzmann Machines," *Cognitive Sci.*, **9**, 147 (1985).
- Anderson, N.R. and P. Vamsikrishna, "Best Practices for Information Presentation to Operators," *Proceedings of the AIChE 1996 Process Plant Safety Symposium*, Houston, Texas, 224 (1996).
- Bakshi, B. R. and G. Stephanopoulos, G., "Representation of Process Trends – 3: Multiscale Extraction of Trends from Process Data," *Computers and Chemical Engineering*, **18**, 267 (1994).
- Bakshi, B. R. and G. Stephanopoulos, "Reasoning in Time: Modelling, Analysis and Pattern Recognition of Temporal Process Trends," In G. Stephanopoulos, & Han, *Intelligent Systems in Process Engineering – Paradigms from design to Operations*, 487 (1996).
- Bakshi, B. R., "Multiscale PCA with Application to Multivariate Statistical Process Monitoring," *AIChE J.*, **44**(7), 1596 (1998).
- Berman, Z., and J. S. Baras; "Properties of the Multi-Scale Maxima and Zerocrossing Representations," *IEEE Transactions on Signal Processing*, **41**, 3216 (1993).
- Bowman, A. W., " An Alternative Method of Cross-Validation for the Smoothing of Density Estimates," *Biometika*, **71**, 353 (1984).
- Bullemer, P. and I. Nimmo, "Tackle Abnormal Situation Management with Better Training," *Chemical Engineering Progress*, **94**(1), 43 (1998).
- Chen, B. H., X. Z. Wang, S. H. Yang and C. McGreavy, "Application of Wavelets and Neural Networks to Diagnostic System Development, 1, Feature Extraction," *Comput. Chem. Eng.*, **23**, 899 (1999).
- Cheung, J. T. Y. and G. Stephanopoulos, "Representation of Process Trends – 1: a Formal Representation Framework," *Comput. Chem. Eng.*, **14**, 495 (1990).
- Chui, C. K., *An Introduction to Wavelets*, Academic Press: New York, 1-18 (1992).
-

References

- Cochran, E., and P. Bullemer, "Abnormal Situation Management: Not By New Technology Alone," *Proceedings of the AIChE 1996 Process Plant Safety Symposium*, Houston, Texas, 218 (1996).
- Cohen, A., I. Daubechies, and P. Vial, "Wavelets on the Interval and Fast Wavelet Transform," *Appl. Comput. Harmonic Anal.*, **1**, 54 (1993).
- Cottrel, G. W., P. Munro, and D. Zipser, "Learning Internal Representations from Gray-Scale Images: An Example of Extensional Programming," *Proc. Conf. Cognitive Sci. Soc.*, 461 (1987).
- Cvetkovic, Z., and M. Vetterli, "Discrete-Time Wavelet Extrema Representation: Design and Consistent Reconstruction," *IEEE Transactions on Signal Processing*, **43**, 681 (1995).
- Cybenko, G., "Approximation by Superpositions of a Sigmoidal Function," *Math. Control, Signals, Syst.*, **2**, 303 (1989).
- Dai, X., B. Joseph, and R. L. Motard, "Introduction to Wavelet Transformation and Time Frequency Analysis," In R. L. Motard, and B. Joseph, *Wavelet Applications in Chemical Engineering*, 1-32, Dordrecht: Kluwer (1994).
- Daiguji M.; O. Kudo, and T. Wada, "Application of Wavelet Analysis to Fault Detection in Oil Refinery," *Comput. Chem. Eng.*, **21**, Suppl., S1117 (1997).
- Daubechies, I., "Orthonormal Bases of Compactly Supported Wavelets," *Comm. Pure Appl. Math.*, **XLI**, 909 (1988).
- Daubechies, I., *Ten Lectures on Wavelets*. Philadelphia, PA: Society of Industrial and Applied mathematics (1992).
- Dijkerman, R. W., and R. R. Majmdar, "Wavelet Representations of Stochastic Processes and Multiresolution Stochastic Models," *IEEE Trans. Signal Process.*, **42**, 1640 (1994).
- Dong, D., and T. J. McAvoy, "Nonlinear Principal Component Analysis-Based on Principal Curves and Neural Networks," *Comput. Chem. Eng.*, **20**(1), 65 (1996).
- Donoho, D. L., "De-Noising by Soft-Thresholding," *IEEE Trans. Inform. Theory*, **41**(3), 613 (1995).
- Donoho, D. L., and I. M. Johnstone, "Ideal Spatial Adaptation by Wavelet Shrinkage," *Biometrika*, **81**(3), 425 (1994).
- Donoho, D. L., and I. M. Johnstone, "Adapting to Unknown Smoothness via Wavelet Shrinkage," *J. Amer. Stat. Assoc.*, **90**(432), 1200 (1995).



References

- Donoho, D. L., I. M. Johnstone, and G. Kerkyacharian, "Density-Estimation by Wavelet Thresholding," *Annals of Statistics*, **24**(2), 508 (1996).
- Donoho, D. L., I. M. Johnstone, G. Kerkyacharian, and D. Picard, "Wavelet Shrinkage: Asymptopia?" *J. R. Stat. Soc. B*, **57**, 301 (1995).
- Dunia, R., S. J. Quin, T. F. Edgar, and T. J. McAvoy, "Identification of Faulty Sensors Using Principal Component Analysis," *AIChE J.*, **42**(10), 2797 (1996).
- Dunia, R., S. J. Quin, "Subspace Approach to Multidimensional Fault Identification and Reconstruction," *AIChE J.*, **44**(8), 1813 (1998).
- Eastment, H. T. and W. J. Krzanowski, "Cross-Validatory Choice of the Number of Components from a Principal Component Analysis," *Technometrics*, **24**, 73 (1982).
- Embrey, D. E., "Approaches to Aiding and Training Operators' Diagnosis in Abnormal Situations," *Chemistry and Industry*, **7**, 454 (1986).
- Hagan, M. T. and M. B. Menhaj, "Training feedforward Networks with the Marquardt Algorithm," *IEE Transactions on Neural Networks*, **5**(6), 989, (1994).
- Hall, P., I. McKay, and B. A. Turlach, "Performance of Wavelet Methods for Functions with Many Discontinuities," *Annals of Statistics*, **24**(6), 2462 (1996).
- Harrold, D., "How to Avoid Abnormal Situations," *Control Engineering International*, **5**(3), 23 (1998).
- Harrold, D., "No single Method can Satisfy all the Complexities of ASM", *Control Engineering*, **5**(9), 75 (1998).
- Hastie, T. J., and W. Stuetzle, "Principal Curves," *J. Amer. Stat. Assoc.*, **84**, 505 (1989).
- Heinonen, P., and Y. Neuvo, "FIR-Median Hybrid Filters," *IEEE Trans. on Acoustics, Speech, and Signal Process.*, **ASSP-35**, 832 (1987).
- Hertz, J., A. Krogh, and R. G. Palmer, *Introduction to the theory of Neural Computation*, Addison-Wesley, Redwood City, CA, p. 198 (1991).
- Jackson, J. E., "Principal Components and Factor Analysis: I. Principal Components," *J. Qual. Technology*, **12**(4), 201 (1980).
- Jackson, J. E. and G. S. Mudholkar, "Control Procedures for Residuals Associated with Principal Components Analysis," *Technometrics*, **21**, 341 (1979).
- Janusz, M. E. and V. Venkatasubramanian, "Automatic Generation of Qualitative Descriptions of Process Trends for Fault Detection and Diagnosis," *Engineering Applications of Artificial Intelligence*, **4**, 329 (1991).

References

- Jia, F., E. B. Martin, and A. J. Morris, "Non-linear Principal Components Analysis for Process Fault Detection," *Comput. Chem. Eng.*, **22**(SS), S851 (1998).
- Johnson, R. A., and D. W. Wichin; *Applied Multivariate Statistical Analysis*, Englewood Cliffs, NJ: Prentice-Hall (1992).
- Kosanovich, K. A., and M. J. Piovoso, "PCA of Wavelet Transformed Process Data for Monitoring," *Intell. Data Anal.*, <http://www.elsevier.com/locate/ida>, **1**, 2 (1997).
- Kramer, M.A., "Nonlinear Principal Component Analysis Using Autoassociative Neural Networks," *AIChE J.*, **37**(2), 233 (1991).
- Kramer, M. A., "Autoassociative Neural Networks," *Comput. Chem. Eng.*, **16**, 313 (1992).
- Kramer, M. A. and R. S. H. Mah, "Model-Based Monitoring," *Proc. Int. Conf. On Foundations of Computer Aided Process Operations*, D. Rippin, J. Hale, J. Davis, eds. CACHE, Austin, TX (1994).
- Kresta, J., J. F. MacGregor, and T. E. Marlin, "Multivariate Statistical Monitoring of Process Operating Performance," *Can. J. Chem. Eng.*, **69**, 35 (1991).
- Krzanowski, W. J., "Cross-validation in Principal Component Analysis," *Biometrics*, **43**, 515 (1987).
- Ku, W., R. H. Storer, and C. Georgakis, "Disturbance Detection and Isolation by Dynamic Principal Component Analysis," *Chem. Intell. Lab. Syst.*, **30**, 179 (1995).
- Lorenzo, D., *A Manager's Guide to Reducing Human Errors Improving Human Performance in the Chemical Industry* (1991).
- MacGregor, J. F., C. Jaeckle, C. Kiparissides, and M. Koutoudi, "Process Monitoring and Diagnosis by Multiblock PLS Methods," *AIChE J.*, **40**, 827 (1994).
- Malinowski, E. R., *Factor Analysis in Chemistry*, Wiley, New York (1991).
- Mallat, S. G., "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation," *IEE Trans. Pattern Anal. Mach. Intell.*, **PMAI-11**, 674 (1989).
- Mallat, S. G., "Zero-Crossings of a Wavelet Transform," *IEEE Trans. Inform. Theory*, **IT-37**(4), 1019 (1991).
- Mallat, S., and W. L. Hwang,, "Singularity Detection and Processing with Wavelets," *IEEE Transactions on information theory*, **38**, 617 (1992).
- Mallat, S., and S. Zhong, "Characterisation of Signals from Multi-Scale Edges," *IEEE Transaction Pattern Analysis and Machine Intelligence*, **14**(7), 710 (1992).

References

- Martin, E. B. and A. J. Morris, "Non-Parametric Confidence Bounds for Process Performance Monitoring Charts," *Journal of Process Control*, **6**(6), 349 (1996).
- Miller, P., R. E. Swanson, and C. F. Heckler, "Contribution Plots: The Missing Link in Multivariate Quality Control," *37th Annual Fall Conference*, ASQC, Rochester, NY (1993).
- Motard, R. L. and B. Joseph, *Wavelet Applications in Chemical Engineering*, Kluwer Academic Publishers: Boston, 1-26 (1994).
- Musliner, D. J., and K. D. Krebsbach, "Applying a Procedural and Reactive Approach to Abnormal Situations in Refinery Control," *Chemical Technology*, **5**(9), 3 (1998).
- Nason, G. P., "Wavelet Shrinkage Using Cross-Validation," *J. R. Stat. Soc. B*, **58**(2), 463 (1996).
- Nason, G. P. and B. W. Silverman, "The Discrete Wavelet Transform in S," *J. Comput. Graph. Statist.*, **3**, 163 (1994).
- Nason, G. P. and B. W. Silverman, "The Stationary Wavelet Transform and Some Statistical Applications," *Lect. Notes Statist.*, **103**, 281 (1995).
- Nimmo, I., "Abnormal Situation Management," *Process & Control Engineering*, **49**(5), 8 (1996).
- Nimmo, I., "Abnormal Situation Management: Giving Your Control System Ability to Cope," *The Journal for Industrial Automation and Control*, **32**(9), 23 (1998a).
- Nimmo, I., "Adequately Address Abnormal Operations," *Chemical Engineering Progress*, **91**(9), 36 (1995).
- Nimmo, I., "Industry Initiative Addresses Abnormal Events," *Hydrocarbon Processing*, October, 71 (1998b).
- Nomikos, P., and J. F. MacGregor, "Monitoring Batch Processes Using Multiway Principal Component Analysis," *AIChE J.*, **40**(8), 1361 (1994).
- Nounou, M. N., and B. R. Bakshi, "On-Line Multiscale Rectification of Random and Gross Errors without Process Models," Technical Report, Dept. of Chemical Engineering, Ohio State Univ. (1998).
- Ogden, R. T., *Essential Wavelets for Statistical Applications and Data Analysis*, Boston: Birkhauser (1997).
- Piovosio, M. J., K. A. Kosanovich, and R. K. Pearson, "Monitoring Process Performance in Real-Time," *Proc. of the Amer. Contr. Conf.*, Chicago, IL, **3**, 2359 (1992).



References

- Ramesh, T.S., and B.V. Kral, "Plant Monitor: An On-line Advisory System for Monitoring Polyethylene Plants," *Intelligent Systems for Process Engineering (IPSE) Conference*, Snowmass, Colorado, (1995).
- Ramesh, T., B. Kral, and J. Freeman, "A Generic Real-Time Monitor for Detecting Abnormal Events in Continuous Processes," *Proceedings of the AIChE 1996 Process Plant Safety Symposium*, Houston, Texas, 209 (1996).
- Rioul, O. and M. Vetterli, "Wavelets and Signal Processing," *IEEE Sig. Proc. Mag.*, **8**, 14 (1991).
- Rothenberg, D., and I. Nimmo, "The Concept of Abnormal Situation Management and Mechanical Reliability," *Proceedings of the AIChE 1996 Process Plant Safety Symposium*, Houston, Texas, 193 (1996).
- Scott, D. W., *Multivariate Density Estimation: Theory, Practice and Visualisation*, New York: Wiley (1992).
- Shao, R., F. Jia, E. B. Martin and A. J. Morris, "Wavelets and Non-Linear Principal Components Analysis for Process Monitoring," *Control Engineering Practice*, **7**, 865 (1999).
- Shimizu, H., K. Uchiyama, and S. Shioya, "On-Line Fault Diagnosis for Optimal Rice Alpha-Amylase Production Process of a Temperature-Sensitive Mutant of *Saccharomyces Cerevisiae* by an Autoassociative Neural Network," *Journal of Fermentation and Bioengineering*, **83**(5), 435 (1997).
- Sticles, R. P. and G. A. Melhem, "How much Safety is Enough?" *Hydrocarbon Processing*, October, 50 (1998).
- Strang, G. and R. Nguyen, *Wavelets and Filter Banks*, John Wiley & Sons: New York (1995).
- Tan, S., and M. L. Mavrovouniotis, "Reducing Data Dimensionality Through Optimizing Neural Network Inputs," *AIChE J.*, **41**, 1471 (1995).
- Tong, H., and C. M. Crowe, "Detection of Gross Errors in Data Reconciliation by Principal Component Analysis," *AIChE J.*, **41**, 1712 (1995).
- Wang, X. Z., B. H. Chen, S. H. Yang and C. McGreavy, "Application of Wavelets and Neural Networks to Diagnostic System Development, 2, an Integrated Framework and its Application," *Comput. Chem. Eng.*, **23**, 945 (1999).
- Whitely, J. R., J. F. Davis, "Knowledge-Based Interpretation of Sensor Patterns." *Computers and Chemical Engineering*, **16**, 329 (1992).



References

- William, B. C., "Doing Time: Putting Qualitative Reasoning on Firmer Ground," *National Conference on Artificial Intelligence*, Philadelphia, (1986).
- Wise, B. M., N. L. Ricker, D. F. Veltkamp, and B. R. Kowalski, "A Theoretical Basis for the Use of Principal Component Models for Monitoring Multivariate Processes," *Process Control and Quality*, **1**, 41 (1990).
- Wise, B. M. and Gallagher, N. B., "Process Chemometrics Approach to Process Monitoring and Fault Detection," *J. Process Control*, **6**, 329 (1996).
- Wold, S., "Cross-Validatory Estimation of the Number of Principal Components in Factor and Principal Component Analysis," *Technometrics*, **20**, 397 (1978).
- Wold, S., K. Esbensen, and P. Geladi, "Principal Component Analysis," *Chem. Intell. Lab. Syst.*, **2**, 37 (1987).
- Wornell, G. W., "A Karhunen-Loeve-Like Expansion for $1/f$ Processes via Wavelets," *IEEE Trans. Inform. Theory*, **IT-36**(4), 859 (1990).
- Wold, S., N. Kettaneh, and K. Tjessem, "Hierarchical Multiblock PLS and PC Models for Easier Model Interpretation and as an Alternative to Variable Selection," *J. Chemometrics*, **10**, 463 (1996).
- Wold, S., "Exponentially Weighted Moving Principal Component Analysis and Projection to Latent Structures," *Chem. Intell. Lab. Syst.*, **23**, 149 (1994).