

Subspace System Identification for Multivariate Statistical Process Control

*A thesis submitted to the University of Manchester for the degree of
Engineering Doctorate in the Faculty of Science and Engineering*

2003

Richard Treasure

Department of Engineering

ProQuest Number: 10757306

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



ProQuest 10757306

Published by ProQuest LLC (2018). Copyright of the Dissertation is held by the Author.

All rights reserved.

This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346



Th 26396



Table of Contents

Abstract	7
Declaration and Copyrights	8
Acknowledgement	9
Operators and Notational Conventions	10
Abbreviations and Acronyms	17

PART ONE

CHAPTER 1	18
------------------	-----------

PROJECT BACKGROUND	18
---------------------------	-----------

1.1	Introduction	18
1.2	Control Technology Centre Ltd	19
1.2.1	MonitorMV	20
1.3	The Engineering Doctorate	22
1.3.1	Project Background	22
1.4	Subspace System Identification	23
1.4.1	Subspace methods for monitorMV	25
1.4.2	Benefits to the company	26
1.5	Statement of aims	27
1.6	A brief survey the thesis	28

CHAPTER 2	30
------------------	-----------

SYSTEM IDENTIFICATION	30
------------------------------	-----------

2.1	Introduction	30
2.2	The System	31
2.3	System Identification	32

2.3.1	The system	32
2.3.2	The model	32
2.3.3	The identification procedure	35
2.3.4	The Experimental Condition	37
2.4	Model Structures	38
2.4.1	ARX Model Structure	39
2.4.2	ARX Identification Procedure	40
2.4.3	FIR Model Structure	41
2.4.4	State Space Model Structure	41
2.4.4.1	Identifying a state space model using least squares	43
2.5	Conclusion	44
CHAPTER 3		46
SUBSPACE SYSTEM IDENTIFICATION		46
3.1	Introduction	46
3.2	Literature Review	47
3.3	Realisation theory	53
3.3.1	Realisation theory using impulse excitation	53
3.3.2	Realisation theory using white noise excitation	56
3.4	Subspace Methods	58
3.4.1	CVA Approach	59
3.4.2	The MOESP Algorithm	62
3.4.3	N4SID Approach	62
3.5	Conclusion	74
CHAPTER 4		75
IDENTIFICATION OF A SIMPLE SIMULATED SYSTEM		75
4.1	Introduction	75
4.2	Simulation of a mass-spring-damper system	78
4.2.1	The identification procedure.	79

4.2.2	Choosing the model order	80
4.2.2.1	Cross-validation	81
4.2.2.2	Akaike Information Criterion (AIC)	82
4.2.2.3	Using the rank of the projection matrix	83
4.2.3	Residuals analysis	84
4.2.4	Pure time delay	85
4.2.5	User choices for the subspace algorithms	86
4.3	Results	86
4.3.1	Measures to determine the optimum model structure	87
4.3.2	The prediction accuracy of state space and ARX models	89
4.3.3	Using the singular values to determine model order	91
4.3.4	The effect of noise	92
4.3.5	The effect of the number of block rows on model accuracy	95
4.3.6	Procedures for dealing with time delays	96
4.3.7	MSPE as a random variable	97
4.4	Conclusion	98
4.5	Figures and Tables	103
CHAPTER 5		122
IDENTIFICATION OF INDUSTRIAL PLANT		122
5.1	Introduction	122
5.2	The FCCU Simulation	124
5.2.1	FCCU Experimental Design	125
5.2.2	FCCU Model Identification	127
5.2.3	Optimising the FCCU model structures	127
5.2.4	The effect of the number of block rows on prediction accuracy	128
5.2.5	The use of the eigenvalue plot to determine the system order.	130
5.2.6	State space model order reduction	130
5.2.7	A comparison between state space and ARX model structures	131
5.2.8	Relative speed of the algorithms	131
5.3	Conclusion	132
5.4	Figures and Tables	135

PART TWO

CHAPTER 6	150
A SUBSPACE METHOD FOR MONITORMV	150
6.1 Introduction	150
6.1.1 State space models for process monitoring	153
6.2 A subspace method for process monitoring	154
6.2.1 Calculating the state sequences	155
6.2.1.1 Online updating of the state sequence	156
6.3 The Subspace method	158
6.3.1 Multivariate Statistics for the Subspace method	161
6.3.2 Calculation of Hotelling's T^2 Statistic	162
6.3.3 Calculation of the Q Statistics	164
6.3.4 Determining the number of principal components	165
6.3.5 Determination of the Subspace Model Structure	166
6.3.6 Subspace Condition Monitoring Procedure	167
6.4 Conclusion	169
CHAPTER 7	170
DYNAMIC MODELS FOR MSPC	170
7.1 Introduction	170
7.2 Multivariate Statistical Process Control	171
7.2.1 Subspace Method	171
7.2.2 Principal Components Analysis	172
7.2.2.1 PCA Univariate Statistics	174
7.2.2.2 PCA contribution calculations	174
7.2.3 Partial Least Squares	176
7.2.4 Dynamic PCA and Dynamic PLS	178
7.3 CSTR Simulation	179
7.4 Setting the Model Structure	181
7.4.1 The model structure for the Subspace Method	181

7.4.2	The model structures for dynamic PCA and dynamic PLS	182
7.5	Results	183
7.5.1	Fault A	183
7.5.2	Fault B	184
7.5.3	Fault C	184
7.5.4	Comparison between the subspace method and PCA	186
7.5.4.1	The effect of scaling the states	187
7.6	Conclusion	188
7.7	Figures and Tables	190
CHAPTER 8		204
THE SUBSPACE METHOD AND DPCA		204
8.1	Introduction	204
8.2	Comparison of model structures	207
8.3	Simulation Studies	210
8.3.1	Simulation Study 1: A Deterministic 1 st order system	211
8.3.2	Simulation Study 2: Auto-correlated process data	215
8.3.3	Simulation Study 3: A Deterministic 2 nd order system	218
8.4	Conclusion	222
8.5	Figures	224
CHAPTER 9		234
CONCLUSIONS AND FURTHER WORK		234
9.1	Conclusions	234
9.2	Further Work	238
9.3	Publications	241
APPENDIX		242
REFERENCES		244

Abstract

In this work, a novel technique for process condition monitoring of continuous industrial processes is developed using subspace system identification techniques. The efficacy of the novel Subspace Method is assessed by benchmarking it against the current process condition monitoring methods used in the process monitoring software package monitorMV. Several case studies are provided and guidelines regarding the model structure are described for the subspace method.

The novel subspace method provides an improved dynamic modelling capability for the software product monitorMV. Dynamic models are used for process condition monitoring where there is a need accommodate dynamic transients or where autocorrelated data has an adverse effect on the statistical analysis. The current dynamic approach in MonitorMV involves building a data matrix from time-shifted process data. The disadvantage of this approach is the amount of data that is involved, which involves a number of process variables that are included in the calculations. In contrast, the subspace method condenses the process data into time-shifted state sequences. This leads to a simpler analysis procedure using fewer process variables.

The first part of the thesis deals with subspace system identification and the modelling of linear processes. A review of time series modelling and system identification methodology is provided, including a description of subspace system identification. Although well received in the academic community, subspace methods are largely untried and unused in industry, however they have been claimed to represent a significant advance in linear system identification techniques. In contrast to traditional approaches, subspace methods identify state sequences directly from the process measurements. One barrier to the wider acceptance of the methodology is the rather complicated theory that lies behind the algorithms. The mechanics of the subspace algorithms are described, then simulation studies are provided to demonstrate various aspects, and the effects of user choices on model accuracy. Finally industrial simulations are used to benchmark the models against industry standard linear modelling methods.

Declaration and Copyrights

Declaration

No portion of this work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyrights

- (1) Copyright in text of this thesis rests with the Author. Copies (by any process) either in full, or of extracts, may be made **only** in accordance with instructions given by the Author and lodged in the John Rylands University Library of Manchester. Details may be obtained from the Librarian. This page must form part of any such copies made. Further copies (by any process) of copies made in accordance with such instructions may not be made without the permission (in writing) of the Author.
- (2) The ownership of any intellectual property rights which may be described in this thesis is vested in the University of Manchester, subject to any prior agreement to the contrary, and may not be made available for use by third parties without the written permission of the University, which will prescribe the terms and conditions of any such agreement.

Further information on the conditions under which disclosures and exploitation may take place is available from the Head of the Department of Engineering.

Acknowledgement

The author would like to thank the following people whose contributions have proved invaluable to this work.

My office buddies Ahdy Prayitno, Zhang Hongwei, Guo Yinghao and Ji Gong, for their support, and the pleasant working environment I have enjoyed throughout the duration of this study,

The Control Technology Centre Ltd, for their keen attention and advice concerning practical issues, and Dr Barry Lennox for his help with improvements and corrections.

Dr Uwe Kruger, for help with the programming of the algorithms used in this project, and also the wealth of technical support that he has supplied to this project,

My supervisor, Professor Jonathan E. Cooper, for the guidance he has given throughout my study,

The EngD Crew, for the enlightening professional training experiences, and the memorable social occasions that we shared.

This work was supported by Control Technology Centre Ltd, and the Engineering and Physical Sciences Research Council.

This work is dedicated to my parents, and to the memory of my grandparents.

Operators and Notational Conventions

A, B, C, D state space system matrices

A_{TLS}, B_{TLS}, C_{TLS}, D_{TLS}

A, B, C, D polynomials of the Box-Jenkins model structure (Ch. 2)

$A \otimes B$ Kronecker product of *A* and *B*

B diagonal matrix for inner PLS model

C the experimental condition, describing how the experiment was carried out

C_r crest factor (Ch. 2)

$\gamma^2 \tilde{\mathbf{C}}_Y$ vector containing output variable contributions to the T^2 statistic

$\gamma^2 \tilde{\mathbf{C}}_U$ vector containing input variable contributions to the T^2 statistic

$T^2 \tilde{\mathbf{C}}_{\bar{X}_k}$ vector containing \bar{X}_k contributions to the T^2 statistic

$\gamma^2 \tilde{\mathbf{C}}_{\bar{X}_{k+1}}$ vector containing \bar{X}_{k+1} contributions to the T^2 statistic

$\gamma^2 \tilde{\mathbf{C}}_{\xi}$ contributions of all the process variables and the states to the T^2 statistic

$(Q) \tilde{\mathbf{C}}_{\xi, k}$ vector containing the values of the k^{th} contribution of each of the process variables, and the states, to the Q statistic

$PCA \mathbb{C}_{T^2}$ contributions to the PCA T^2 statistic

$PCA \mathbb{C}_Q$ contributions to the PCA Q statistic

D_M number of free parameters in the model

E expectation operator

E process noise sequence, see (6.3)

E PCA model residuals, see (7.3)

F measurement error of the system inputs \mathbf{U}_f

$F_{M, K-\gamma, \alpha}$ confidence limit of an *F*-distribution.

\in belongs to, is an element of

G output error vector of the measurements \mathbf{Y}_f

$G(q, \theta)$ transfer function corresponding to the parameter value θ , see (2.1)

$H(q, \theta)$ analogous to *G* but for the transfer function from *e* to *y*, see (2.1)

$H(q)$ transfer function for the state equations, see (2.22)
 \mathbf{H} Hankel matrix of impulse response parameters see (3.14)
 \mathbf{H}_c lower Toeplitz matrix containing impulse response samples of a controller
 I an identification method used for system identification
 \mathbf{K} Kalman gain
 K number of recorded samples of the reference data set, see (6.28)
 L measurement delay in the system
 M number of principal components retained in the subspace condition monitor
 \mathcal{M} a model used to describe the input-output relationship of a physical system
 \mathbf{M}_{X_k} least squares regression model for the state sequence X_k
 $\mathcal{M}(\theta)$ a given model structure, corresponding to the parameter vector θ
 \mathcal{M}_1 the subid.m algorithm, see also [1]
 \mathcal{M}_2 the com_stat.m algorithm, see also [1]
 \mathcal{M}_3 algorithm based on canonical variate analysis
 \mathcal{M}_4 the n4sid.m “moesp” algorithm from the Matlab system identification toolbox
 \mathcal{M}_5 the n4sid.m “cva” algorithm from the Matlab system identification toolbox
 \mathcal{M}_6 the arx.m algorithm from the Matlab system identification toolbox
 M^{-1} inverse of a matrix M
 M^\dagger pseudoinverse of a matrix M
 M^T transpose of a matrix M
 \hat{M} estimate of a matrix M
 \tilde{M} alternative estimate of a matrix M
 M_k Number of free parameters in a particular model structure
 $M_{a,b,c}$ matrix M where the a, b, c are indices for the first element, the number of rows and the number of columns respectively
 $\text{rank}(M)$ the rank of a matrix M
 $\text{trace}(M)$ the sum of the diagonal elements in M
 $\|M\|_F^2$ Frobenius norm of a matrix M
 $\text{vec}(M)$ vector operation of stacking the columns of M on top of each other

$ m $	absolute value of the scalar m
N	the number of columns in the block Hankel matrix structure
\mathbf{Q}	RQ decomposition matrix containing orthogonal rows, see (3.36)
\mathbf{Q}	PLS projection matrix, see (7.15)
\mathbf{R}	lower triangular matrix of RQ decomposition, see (3.36)
\mathbf{R}	PLS projection matrix, see (7.14)
\mathbf{R}^l	vector space of l -dimension real vectors
$\mathbf{R}^{n \times m}$	vector space of $n \times m$ -dimension real matrices
$\hat{R}_{eu}(\kappa)$	covariance estimate between model residuals and inputs for time lag κ
R_{uu}	covariance estimate for the system inputs
R_{yu}	cross-variance estimate between the inputs u and outputs y
\mathbf{R}_{U_p}	regression matrix describing the relationship $\hat{\mathbf{Y}}_f = f(\mathbf{U}_p)$
\mathbf{R}_{Y_p}	regression matrix describing the relationship $\hat{\mathbf{Y}}_f = f(\mathbf{Y}_p)$
\mathbf{R}_{U_f}	regression matrix describing the relationship $\hat{\mathbf{Y}}_f = f(\mathbf{U}_f)$
S	the physical process that provides the experimental data
T_c^2	confidence limit for Hotelling's T^2 statistic
\mathbf{T}	similarity transformation matrix
U_k	input sequence with the vector u_k as the first element, see (2.26)
\mathbf{U}_f	block Hankel matrix of the “future” inputs of the system, see (3.40)
\mathbf{U}_p	block Hankel matrix of the “past” inputs of the system, see (3.40)
\mathbf{U}_k'	PCA model of the system inputs \mathbf{U}_k
\mathbf{U}_k''	residual matrix of the PCA model of the system inputs \mathbf{U}_k
$V(\theta)$	optimisation criterion for model structure $\mathcal{M}(\theta)$
X_k	state sequence with first element \mathbf{x}_k
X_{k+1}	state sequence with first element \mathbf{x}_{k+1}
\mathbf{X}_k	state sequence for TLS solution, see (6.14)
$\bar{\mathbf{X}}_k$	scaled state sequence for TLS solution, see (6.14)
\mathbf{X}_f	analogous to \mathbf{X}_k

\mathbf{X}_{j+1}	analogous to X_{k+1}
$\bar{\mathbf{X}}_k^I$	PCA model of the scaled state sequences $\bar{\mathbf{X}}_k$
$\bar{\mathbf{X}}_k^H$	residual matrix of the PCA model of the scaled state sequences $\bar{\mathbf{X}}_k$
$\bar{\mathbf{X}}_{k+1}^I$	PCA model of the scaled state sequences $\bar{\mathbf{X}}_{k+1}$
$\bar{\mathbf{X}}_{k+1}^H$	residual matrix of the PCA model of the scaled state sequences $\bar{\mathbf{X}}_{k+1}$
$X_k^{(m)}$	analogous to $\bar{\mathbf{X}}_k$
$X_k^{(p)}$	subspace method model prediction of $\bar{\mathbf{X}}_k$
$X_{k+1}^{(m)}$	analogous to $\bar{\mathbf{X}}_{k+1}$
$X_{k+1}^{(p)}$	subspace method model prediction of $\bar{\mathbf{X}}_{k+1}$
Y_k	output sequence with the vector y_k as the first element, see (2.26)
\mathbf{Y}_f	block Hankel matrix of the “future” outputs of the system, see (3.41)
\mathbf{Y}_k	the system outputs
\mathbf{Y}_p	block Hankel matrix of the “past” outputs of the system, see (3.41)
\mathbf{Y}_k^s	the stochastic part of a combined deterministic-stochastic system
\mathbf{Y}_k^I	PCA model of the system outputs \mathbf{Y}_k
\mathbf{Y}_k^H	residual matrix of the PCA model of the system outputs \mathbf{Y}_k
\mathbf{Z}	$= (\bar{\mathbf{X}}_{k+1}^T \quad \mathbf{Y}_k^T \quad \bar{\mathbf{X}}_k^T \quad \mathbf{U}_k^T)$, subspace method data matrix on which PCA is applied
\mathbf{Z}_{PCA}	matrix corresponding to a PCA analysis of the system
$\hat{\mathbf{Z}}$	PCA model predictions
\mathbf{Z}	PCA model residuals
$\hat{\mathbf{Z}}_N$	PCA model with N principal components
$diag(a, b, c, \dots)$	a diagonal matrix, with a, b, c, \dots on the main diagonal
\max_i	The maximum value of a vector sequence
\in	is an element of
r_{mm}	covariance estimate for a single system input of a MIMO system
Λ	weighting matrix (Ch. 2)
Λ_M	diagonal matrix, containing the inverse values of the normal operation variance of the principal components for the subspace method

Λ_N	analogous to Λ_M , but used in PCA analysis
Φ_r	Toeplitz matrix with r block rows, containing the deterministic Markov parameters $\mathbf{D}, \mathbf{CB}, \mathbf{CAB}, \dots$, see (3.44)
Φ	loading matrix from PCA analysis
Φ_M	subspace method model containing the first M principal directions of the process
Φ_N	PCA model containing the first N principal directions of the process
Φ_U	loading matrix containing the weightings for the system inputs.
$\Phi_{\bar{\mathbf{X}}_k}$	loading matrix containing the weightings for the state sequence $\bar{\mathbf{X}}_k$
$\Phi_{\bar{\mathbf{X}}_{k+1}}$	loading matrix containing the weightings for the state sequence $\bar{\mathbf{X}}_{k+1}$
Φ_Y	loading matrix containing the weightings for the system outputs.
Γ_r	extended observability matrix with r block rows
Ω_r	extended controllability matrix with r block columns
Π	Projection operator – projects the row space of a matrix onto the row space of a matrix B ;
Π_B	$= B^T (BB^T)^{-1} B$
$\hat{\theta}$	estimate of the parameters for the model structure $M(\theta)$
Σ_{FF}	covariance matrix of of the future outputs in CVA, see (3.24)
Σ_{PP}	analogous to Σ_{FF} , but for the past system measurements, see (3.25)
Σ_{FP}	cross variance between future outputs and past system measurements in CVA
\mathbf{T}	principal components matrix of PCA analysis
\mathbf{T}_M	matrix containing the first M principal components corresponding to the model of the process
T_k	vector containing the values of \mathbf{T}_M for time instant k
$\chi^2(K)$	Chi squared density function, with K degrees of freedom
α	confidence limit
β	size of the reference data set, see (7.12)
γ	number of degrees of freedom, see (6.28)
$\delta_{k,i}$	Kronecker Delta
$e(t)$	disturbance at time t

ε_k	subspace method Q statistic residuals
$\varepsilon(k, \theta)$	model residual at instant k , for a model with parameter vector θ
$\varepsilon_{k,U}$	prediction error of the k^{th} instance of the system input
$\varepsilon_{k,X}$	prediction error of the k^{th} instance of the state sequence
$\varepsilon_{k,Y}$	prediction error of the k^{th} instance of the system outputs
$\phi_{i,j}$	the element in the i^{th} row and the j^{th} column of the matrix Φ_N
φ_α	value of the confidence limit for Q statistic, with a confidence of α .
f_I	vector of future system measurements as used in CCA, section 1.4.1
λ_i	normalising factor = variance of the i^{th} state
μ	mean value
μ_φ	mean value of PRESS, for reference data set, see (6.39)
$\rho(a, b)$	correlation estimate as a function of the parameters a and b
σ	standard deviation
σ_φ	variance of PRESS, for reference data set, see (6.39)
τ_k	row vector of the co-ordinates of the k^{th} T score, analogous to T_k
v_k	score corresponding to the projection of the k^{th} response variables
ω	modal frequency (Ch. 4)
ξ	damping ratio (Ch. 4)
ξ_k	k^{th} sample of the predictor variables
ψ_k	k^{th} sample of the predicted variables
ζ_k	data vector, containing the k^{th} sample of the process variables
$\zeta_{k,n,m}$	data vector defining ARX model structure of order n , input delay spread m
$T^2 c_{i,j}$	contribution of the j^{th} process variable to the i^{th} T score
\mathbf{e}_k	residuals of the PLS predicted variables
\mathbf{f}_k	residuals of the PLS predictor variables
$g_i(\theta)$	the i^{th} value of the parameter vector θ , for the time-series model g , see (4.18)
l	number of system outputs
m	number of system inputs

n	state space model order
n_{ζ}	number of process variables, see (7.5)
n_{PCA}	number of principal components, see (7.12)
n_{PLS}	number of latent variables used in PLS, analagous to n_{PCA}
p_t	vector of past system measurements as used in CCA, section 1.4.1
q, q^{-1}	forward and backwards shift operator
r	number of block rows used in block Hankel matrices for subspace methods
$u(t)$	input variable at time t
\mathbf{v}_k	k^{th} instance of the process noise, see (3.38)
\mathbf{w}_k	k^{th} instance of the measurement noise, see (3.38)
\mathbf{x}_k	k^{th} instance of the state \mathbf{x} .
$y(t)$	output variable at time t
$\hat{y}(k, \theta)$	model prediction at time k , based on the model $\mathcal{M}(\theta)$

Abbreviations and Acronyms

<i>lv</i>	Latent Variable
AIC	Akaike's Information Criterion
ARX	<u>A</u> u <u>t</u> o <u>r</u> e <u>r</u> e <u>s</u> sive with <u>E</u> xogenous Inputs Model
BIC	Bayesian Information Criterion
CCA	Canonical Correlation Analysis
CVA	Canonical Variates Analysis
DOF	Degrees of Freedom
DPCA	Dynamic Principal Components Analysis
EIV	Error-In-Variables
ERA	Eigensystem Realisation Algorithm
FDLTI	Finite Dimension Linear Time Invariant
FIR	Finite Impulse Response
IV	Instrumental Variables
LHS	Left Hand Side
LTI	Linear Time Invariant
MSE	Mean Squared Error
MIMO	Multi Input Multi Output
MISO	Multiple Input Single Output
MOESP	<u>M</u> IMO <u>O</u> utput- <u>E</u> rror <u>S</u> tate Space Model
MSIT	Matlab System Identification Toolbox
MSPE	Mean Squared Prediction Error
N4SID	<u>N</u> umerical Method <u>f</u> or <u>S</u> ystem <u>I</u> dentification
OLS	Ordinary Least Squares
PCA	Principal Components Analysis
PCA _{CV}	PCA Cross-validation Procedure
PEM	Prediction Error Method
PLS	Partial Least Squares or Projection to Latent Structures
PRBS	Pseudo Random Binary Sequence
PRESS	Prediction Sum of Squares
RHS	Right Hand Side
RLS	Recursive Least Squares
RMS	Root Mean Squared
SISO	Single Input Single Output
SM	Subspace Method
SPE	Squared Prediction Error
SVD	Singular Value Decomposition
TLS	Total Least Squares
4SID	A family of algorithms closely related to the N4SID algorithm

Chapter 1

Project Background

The sponsoring company is introduced, and the benefits of this work to the sponsoring company are outlined. A project background is provided, this includes a summary of the company's product and a statement of the aims of the project. The Engineering Doctorate is described, then finally the field of subspace system identification is introduced. The chapter ends with a brief survey of the thesis.

1.1 Introduction

Continuous processes are essential for the supply of goods and services in a global economy. Industrial plants operate 24 hours a day, 365 days a year, to supply the constant demand for energy and material goods. These include oil and gas refineries, nuclear power plants, chemical, pharmaceutical and paper plants, and automatic processes in the food and beverage industries.

The constant demand for cheaper, more reliable products, manufactured to higher specifications, means that there is continual effort to reduce costs and to improve standards. The general trend is for greater numbers and a higher levels of automation and increasing complexity. Cleaner processes that are more efficient, with less wastage are attractive from the point of view of environmental friendliness and commercial

competitiveness. Legislation concerning emissions and demands for environmentally friendly operation further add to the demands on modern process managers. Above all is the need for process optimisation which leads to reduced operating costs, cheaper products and increased market share or increased returns for share holders and/or organic growth within the company.

The need to implement advanced control and maintenance strategies is therefore more important than ever. Such strategies require good models. Subspace methods represent a significant advance in linear system identification techniques. In contrast to traditional approaches, subspace methods identify state sequences directly from the process measurements. The model parameters are then calculated to provide a linear fit to the state sequences.

This thesis reports on the application of novel mathematical modelling techniques, i.e. subspace system identification, for the modelling of complex industrial processes. The improved modelling methods will enhance the modelling capabilities of the industrial process condition monitoring software package monitorMV which is developed and marketed by the project sponsors Control Technology Centre Ltd.

1.2 Control Technology Centre Ltd

The Control Technology Centre Ltd (CTC Ltd), for the transfer of control technologies to industry, is part of the Manchester Innovation Holdings Ltd group of companies. Manchester Innovation, the commercial arm of the University of Manchester, was formed in 1999 by the merger of Vuman Ltd. and Manchester Biotech.

The focus of CTC Ltd. is the development, support and industrial exploitation of condition monitoring technologies through the monitorMV software package. In support of these goals, CTC Ltd. maintains close links with the Manchester School of Engineering and other academic institutions. In addition, CTC Ltd. works closely with a number of companies on individual monitorMV applications.

Trials of monitorMV are ongoing in conjunction with the following companies

- Falconbridge Ltd. (Canada)

- SSAB (Swedish Steel)

Proposals are under consideration for monitorMV trials with the following companies

- Unilever (UK)
- Aylesford Newsprint (UK)

1.2.1 MonitorMV

MonitorMV is a toolbox of technologies for process condition monitoring, fault detection and diagnosis. The monitorMV software includes a range of methods which fall under the heading of Multivariate Statistical Process Control (MSPC). Technologies included in monitorMV include PCA and/or PLS modelling, statistical modelling using either Gaussian or Kernel-based methods and multiple modelset handling for real-time monitoring of complex processes. In support of these technologies, monitorMV offers a range of visualisation options including 2D/3D contour plots and quality control charts plus the traditional MSPC plots.

The monitorMV software currently exists as a stand-alone product, offering real-time fault detection and diagnosis capabilities available for Windows NT/2000. The monitorMV software features the following technologies:

Process modelling

Principal Component Analysis (PCA) is the traditional method of modelling highly correlated process data for MSPC. A Partial Least Squares (PLS) model is also available which allows processes to be modelled and monitored using cause-and-effect structures.

Statistical modelling

Gaussian (normal) statistics, which lead to elliptical confidence bounds, are the standard approach to MSPC. MonitorMV introduces Kernel-based density estimation (KDE) which can more accurately describe the distribution of complex process data.

Real-time system

The condition of a process is monitored in real-time providing instantaneous fault detection and the capacity for on-line diagnosis.

Multiple modelset capability

Real-time process monitoring can be performed against a number of distinct models and/or condition monitors. Auto-classification selects the most suitable model available and provides useful information about the operating state of the process.

Process visualisation

Process analysis and monitoring is enabled through a range of graphical displays:

2D and 3D contour plots of statistical models

2D and 3D scatter plots of PCA/PLS scores

Trends of PCA/PLS scores

Modified Hotelling's T-squared and Squared Prediction Error (SPE)

Quality control charts based on probabilistic measures, score metric and error metric

Error and variance contribution charts

PCA / PLS loadings chart

Classification chart

Batch Processing

Additional tools, including PCA/PLS projection, have been developed to tackle condition monitoring issues for specifically batch processes

1.3 The Engineering Doctorate

This Engineering Doctorate dissertation has been made possible by the support of Control Technology Centre Ltd who develop and market the process condition monitoring package MonitorMV.

The research in this thesis has been undertaken as the principal component of the requirements of an Engineering Doctorate Degree (Eng. D). A stated objective of the Eng. D programme is to provide an intensive, broadly based training in collaboration with major companies. The Eng. D aims to combine the best aspects of the traditional Ph.D. research and reporting with the practical implications of linking the research to the specific needs of the collaborating company.

The application for Eng. D is supported by a Diploma in Management studies and additional managerial courses aimed at giving engineers the necessary skills for today's industrial management needs.

1.3.1 Project Background

The project is sponsored by the Control Technology Centre Ltd (CTC Ltd). Throughout the nineties CTC and its commercial arm Predictive Control Limited (PCL) developed and marketed the model predictive control software product Connoisseur. The project began in October 1998, as part of Control Technology Centre's drive to consolidate the Connoisseur as a market leader in model predictive control. The original project description was "Subspace system identification for model predictive control".

PCL was bought by the Siebe Group in February 1997. Siebe merged with Birmingham Tyre and Rubber (BTR) in 1999 to form Invensys plc, the largest engineering consortium in the U.K. The merger created the world's biggest provider of intelligent automation systems for manufacturing, process controls, power systems and industrial drives (sharing the market with Siemens, ABB and Emerson). Initial work on subspace system identification was in support of Invensys solutions and their intelligent automation division. Restructuring at Invensys led to responsibility for marketing and development of Connoisseur moving to Foxboro (USA), with initial development of monitorMV running in parallel to the development work on Connoisseur in the US. However further changes have diverted the focus of the research at CTC Ltd exclusively to applications in multivariate statistical process control.

The layout of this thesis reflects the way the market has changed. The first part of the thesis deals with subspace system identification. The main contribution comes in part II.

Subspace methods have been adapted and applied to develop a subspace process modelling methodology tailored to the process condition monitoring environment.

1.4 Subspace System Identification

Subspace system identification refers to a group of closely related methods for identifying linear state space models from measured time-series data. A core application is the calculation of models for prediction/simulation and model predictive control. Process automation is of primary importance in production and manufacture. Computerised control algorithms enable process operators to handle a process in a stable manner through disturbances, to control and optimise a process and reduce the demand for constant operator involvement. This means a more stable process, with reduced variation, improved quality, larger profits and process operators that are free for other tasks. MPC provides a means of making significant reductions in energy usage and at the same time optimises the appropriate outputs. One way to further develop the efficacy of MPC methodologies is to expand and improve process modelling capability.

The first part of the thesis focuses on subspace identification and the modelling of linear systems. In 1996 the paper “N4SID : Subspace algorithms for the identification of combined deterministic-stochastic systems” by Peter Van Overschee and Bart De Moor won the 1996 Automatica Paper of the Year. At around the same time Professor Lennart Ljung, a leading expert in system identification proclaimed the newly developed subspace methods as “the most exciting thing that has happened to system identification in the last five years or so.”

Clearly a call to action. State space models are naturally suited to modelling high dimension multi-input multi-output (MIMO) systems and there exists a wealth of literature for control using state space models. The subspace methods have been developed as a consequence of considerable research effort in Belgium, Holland and Sweden and been proven, subject to certain “mild” conditions for robustness and reliability. They were considered to be a significant advance in identification technology but they were largely unproven in practice. Although well received in the academic community, subspace methods are largely untried and unused in industry.

More recently subspace methods have been described as “state of the art linear model predictive control (MPC) system identification technology” (Qin and Badgwell 2001).

In this report, following a review of time series modelling and system identification methodology, the mechanics of the subspace algorithms are described, then simulation studies are used to benchmark the models against industry standard linear modelling methods. The motivation for these studies is to assess the advantages that subspace methods would bring to model predictive control. For example, the current method employed in Connoisseur involves the indirect calculation of a state space model. It is expected that a direct subspace approach will lead to a more parsimonious model. Furthermore, subspace methods use robust algorithms and well understood matrix algebra. Finally, the N4SID algorithm has been programmed into the C language and is now compatible with the code used in Connoisseur.

The simulation studies have also been used to demonstrate the effect of user choices on model accuracy. The first user choice involves choosing the number of block rows that are used to build the input and output matrices. This governs the way in which the calculations are performed; where there is often a trade off between the accuracy obtained, and the size of the computation.

A second decision is required, regarding the determination of the order of dynamics to be possessed by the model, i.e. the number of states that are used to model the process. A number of methods have been proposed in the literature, several of which are investigated in the simulation study.

However, changes in the business environment have led to a new market and the need for applications in multivariate statistical process control (MSPC). How can direct access to the state sequences of the process be incorporated into a MSPC environment and how could this improve on existing methodologies? The main contribution from this thesis is the adaptation of subspace methods to develop process models tailored to the process condition monitoring environment.

1.4.1 Subspace methods for monitorMV

In Part II of the thesis, a novel technique for process condition monitoring of continuous industrial processes is developed using subspace system identification techniques. The

efficacy of the novel subspace method is demonstrated by benchmarking it against the current process condition monitoring methods in monitorMV (PCA and PLS). Several case studies are provided and user guidelines are suggested for the subspace method.

The subspace method provides an improved dynamic modelling capability for MonitorMV. Dynamic models are used for process condition monitoring where there is a need accommodate dynamic transients or where autocorrelated data has an adverse effect on the statistical analysis. The current dynamic approach in MonitorMV involves building a data matrix from time-shifted process data. The disadvantage of this approach is the amount of data and the size of the computations that are involved, which leads to cumbersome fault diagnosis due to the number of process variables that are included in the calculations. In contrast, the subspace method condenses the process data into time-shifted state sequences. This leads to a simpler fault diagnosis procedure using fewer process variables.

The subspace method brings two advantages to monitorMV:

- (1) The method models both dynamic and static relationships within the data by including the same data sequences used in traditional PCA. These are augmented with time shifted state sequences that model the dynamics of the process. In fact the state sequences contain all the information necessary to obtain a state space model of the system. This added dynamic information allows the model to cope better with dynamic transients in process streams, which often lead to false alarms. This claim is backed up with a simulation study in Chapter 7.
- (2) The current dynamic approach in MonitorMV is Dynamic PCA, which builds a data matrix from time-shifted process data. This usually involves first identifying an appropriate ARX structure for the system, then incorporating the ARX data structure into a PCA analysis. The disadvantage of this approach is the amount of data and the size of the computations that are involved, due to the large dimensionality that is created. This leads to a cumbersome contributions analysis due to the process variables and time-shifted versions of the same being included in the calculations. In contrast, the subspace method condenses the ARX description of the system into time-shifted state sequences. This is expected to lead to a reduced computational load for the ensuing

PCA analysis and for simpler fault diagnosis procedure because of a more streamlined contributions analysis.

1.4.2 Benefits to the company

In this study, a novel method for the identification of a dynamic model for multivariate statistical process control is described. Several advantages enjoyed by the new method are outlined, which provide the MSPC software product monitorMV with a competitive advantage:

- (1) More robust modelling of processes, where dynamic transients lead to excessive false alarms.
- (2) A more flexible model that deals with shifts in operating point.
- (3) An alternative contributions analysis to the dynamic modelling method currently used in monitorMV.

Advantages in terms of improved market share for monitorMV, are contingent on further trials and testing on the appropriate industrial data sets and full integration of the code into the software product.

The market for commercial monitoring packages for industrial processes has only been established over the last few years. The development of MSPC for the process industry, was started at the end of the 1980s, where early versions of MSPC technologies in commercial packages were introduced by Aspen Technology Inc. and MDC Technology in 1998.

The commercial benefits to the process industry from the application of monitoring packages include [133,134]:

- reduction in complexity of the control problem, resulting in improved product specification; the net effect is estimated to be an increase in capacity, estimated at 0.5% of rated yield or capacity for continuous processing facilities such as petrochemical or refinery distillation processes ,

- a simplified view of complex processes, and about 10% increase in the operational efficiency, leading to a US\$100,000 per year per operating area savings in manpower for a medium sized chemical facility,
- earlier and more reliable detection of faults; eliminates operational shutdowns, (estimated at US\$400,000 each for a typical large-scale batch facility),
- assessment of past experience can provide a learning process for fault handling and detection leading to eliminating similar conditions in the future,
- reduction of the number of key variables to be tracked,
- enhanced process understanding through the application of well established statistical techniques,

Surveys of the loss due to inappropriate reaction to anomalous process situations are available for the US based petrochemical industry. Nimmo [133] and Vedam [134] refer to a report published by the Abnormal Situation Management Committee revealing that if appropriate action was taken, the US based petrochemical industry could save up to US\$10b per annum. It is also highlighted that the same industry loses over US\$20b per annum as a consequence of inappropriate reaction to abnormal process behaviour.

The motivation to provide detection and diagnosis tools for the petrochemical industry alone, is therefore apparent. Further reasons for implementing monitoring packages for industrial processes include the competitive advantage gained from marketing a green image, as well as revelation of otherwise unnoticed problem areas. These include ineffective operation units or poorly tuned controllers, which may be revealed through process monitoring and diagnosis [136].

1.5 Statement of aims

- To develop understanding of linear models built from time-series data and the algorithms used to parameterise them.
- To develop and understand subspace system identification techniques.

- To compare and contrast subspace methods with industry standard methods and draw conclusions as regards their accuracy and efficacy
- To understand how improvements to linear modelling capability can address the needs of the modern process industries.
- To understand the needs of the CTC regarding the evolving multivariate statistical process control market and develop the product accordingly
- To develop and improve on the current dynamic model capability in the company's product monitorMV.
- To deliver a report on the new methodology that will provide an assessment of its viability and form a sound the basis for decisions regarding implementation of subspace methods into monitorMV.

1.6 A brief survey the thesis

Chapter 1

Project Background

Chapter 2

The field of system identification is introduced. Linear time-series and state space model structures are described; and least squares approaches to the identification of these model structures are introduced.

Chapter 3

An outline of subspace system identification is provided. A literature review is followed by an introduction to realisation theory, and the development of subspace system identification algorithms. The N4SID algorithm is described in depth. This provides the platform on which the subspace model for process condition monitoring is built, in the second part of the dissertation.

Chapter 4

A mass-spring-damper simulation is used to demonstrate the properties of subspace system identification. The aim is to present a transparent study with which to evaluate the methodology. Several subspace algorithms are compared and contrasted. A autoregressive ARX model structure is also identified and used as a comparison to the state space model structure.

Chapter 5

A simulation of complex industrial plant and an industrial process are identified using subspace system identification. The performance of subspace algorithms is compared with ARX and FIR model structures, on the basis of prediction accuracy. User choices associated with the subspace algorithms and aspects of model order reduction are also considered.

Chapter 6

Subspace methods are used to develop a dynamic model for multivariate statistical process control. A procedure is outlined for optimising user choices concerning the structure of the model. Finally, an online condition monitoring scheme for the subspace method is outlined, including definitions for Hotelling's T^2 and Q statistics.

Chapter 7

The subspace method is compared with other linear dynamic modelling methods on the basis of a continuous process simulation. PCA, DPCA and PLS modelling procedures are outlined. Several connections between the subspace method and dynamic principal components analysis (DPCA) are made. Possible advantages enjoyed by the subspace method over DPCA are suggested.

Chapter 8

The Subspace Method and DPCA are compared. Key similarities and differences are outlined. A possible advantage of using the Subspace Method in conjunction with PCA is demonstrated.

Chapter 9

Conclusions, further work and publications.

Chapter 2

System Identification

The field of system identification is introduced. Further aspects of the system identification procedure are outlined, including experimental design, and model optimisation criteria. The linear time-series and state space model structures to be used in this study are described; and least squares approaches to the identification of these model structures are introduced.

2.1 Introduction

In this chapter, various aspects of system identification including the linear model structures used in this study are described. System identification refers to a group of procedures for estimating mathematical models of physical processes from measured data. It has wide applications in science and technology such as in economics, chemistry, physics and the biological sciences. In the process industries, system identification provides models for simulation, operator training, analysis, condition monitoring, fault detection, prediction, optimisation, control system designs and quality control [1]. The main goal for this study is to use the science of system identification to develop and identify linear models for process condition monitoring. In the first part of the study, linear state space models are identified using subspace methods. These are

compared with difference equation models, using simulated and industrial data. In the second part of the study, the subspace system identification methods are adapted for application in a process condition monitoring context.

2.2 The System

This study is focused on the application of linear modelling methods for monitoring the condition of automatic processes. The systems under study are assumed to be linear-time-invariant (LTI) and fully controllable and observable.

In its most general form, a system may be regarded as any family of trajectories [2]. However in the case of the industrial process data sets under consideration here, the system is considered in an input-output framework, where the process is an “object” in which different input variables interact to produce measured outputs. The relationship of the system outputs to the system inputs is often described using transfer functions [6]. Figure 1 shows a system where the “inputs” have been classified into three types: *manipulated variables* - those that can be manipulated by the process operator (or an automatic controller); *measured disturbances* - these are inputs that can be measured but over which there is no control; and *unmeasured disturbances* - these are not measured directly but may be observed by their effect on the output.

The complexity of many industrial processes presents a particular challenge to obtaining an accurate model. For example, the highly correlated nature of process data requires the application of latent variable techniques such as principal components analysis (PCA), which is covered in the second part of this study. Therefore, numerical methods are an indispensable part of the process of system identification. In fact, the complexity of most industrial processes means that they cannot be modelled perfectly by finite dimension models, however, in many applications finite dimension linear time-invariant FDLTI models have been shown to be sufficient [3, 4].

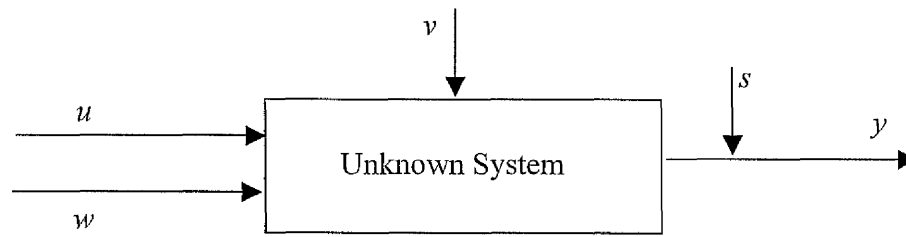


Figure 2.1 A process with outputs y and measured inputs u (the manipulated variables), measured disturbance w , output measurement noise s , and unmeasured disturbance v .

2.3 System Identification

The methodology for obtaining mathematical models of systems encompasses a broad range of activities which are banded together here, under four headings. These four categories are based loosely on the description of system identification given in [5]. They are: The System S , the Model Structure \mathcal{M} , the Identification Method I , and the Experimental Condition C .

2.3.1 The system S

The system S is the physical process that provides the experimental data. The systems considered in this study are dynamic and admit a finite dimension linear time-invariant representation. The system may be open-loop or under closed-loop control. In the first part of this study, the following systems will be considered: S_1 is a single input single output (SISO) sixth order process and S_2 is a simulation of a multi input multi output (MIMO) petrochemical fluid catalytic cracking unit (FCCU).

2.3.2 The model \mathcal{M}

The model \mathcal{M} may be a *parametric* or *non-parametric model*. These two modelling methods are complementary and provide alternative means for describing a process. Non-parametric modelling methods include correlation analysis, transient analysis (e.g. step response plots) and frequency response plots (e.g. bode plots). Non-parametric

models do not impose any structural assumptions about the system (other than that it is linear).

In contrast, parametric models are characterised by their model structure $\mathcal{M}(\theta)$. Parametric modelling methods require that a particular model structure is specified and then an algorithm is chosen to estimate parameters that give a good fit to the data. In some cases parametric models can be based on the laws of Newtonian physics in which case they are derived as a function of continuous time using differential equation (e.g. a model of a mass-spring-damper system). However, it is more common that samples of the process are taken at discrete intervals, then discrete time models are developed by fitting the parameters of a difference equation model. For the most part, this dissertation concerns itself with the application of parametric models, in particular discrete time state space models and difference equation models.

Any given linear model $\mathcal{M}(\theta)$ for a system S can be represented in the form

$$\mathcal{M}: \mathcal{M}\{G(q, \theta), H(q, \theta)\}, \quad (2.1)$$

where q is the backward shift operator and θ is a vector of parameters to be determined [6]. The polynomial terms G and H define the transfer function of the system. For simplicity of notation, a SISO system is described here however exactly the same relations apply to MIMO systems. Given any LTI system, a complete description of the system is given by

$$y(t) = G(q)u(t) + H(q)e(t), \quad (2.2)$$

$$H(q) = 1 + \sum_{m=1}^{\infty} h(m) q^{-m}, \quad (2.3)$$

$$G(q) = \sum_{m=1}^{\infty} g(m) q^{-m}, \quad (2.4)$$

where $e(t)$ is a zero mean, independent, random vector sequence. The general approach to identifying a parametric model is to specify a set of models, known as the model set $\mathcal{M}(\theta)$ from which a final model is chosen. (Specific model structures are discussed in more detail in the next section). An algorithm is applied to find the model parameters θ

that give the best description according to a prespecified optimisation criterion which is chosen according to the engineering requirements.

Most parametric techniques are “black box” methods, meaning that the parameters are found to satisfy a specific optimisation criterion, with no regard for the physical laws which govern the process. This is in contrast to “white box” methods that use the laws of physical science to define a system of differential equations. A third set of parametric models, the so called “grey box” techniques, pre-set some parameters in the model structure and then curve-fit the remaining parameters [6].

Linear models can be transposed between parametric and non-parametric forms, for example, a step response diagram can be obtained by calculating the step response of a difference equation model. These methods are complementary and are often used together, for example, in Kung’s method described in the next chapter, a state space model is identified from the results of a correlation analysis.

First a model set $\mathcal{M}(\theta)$ is defined, then a particular model \mathcal{M} is chosen. The choice of \mathcal{M} is made with consideration given to experience of the process, the type and amount of data available (e.g. frequency domain or time domain), computing power available, the required accuracy and most importantly the intended end use of the model. There may also need to be a trade off between model accuracy and model complexity. All are considered in the choice of \mathcal{M} . For a chosen model-type \mathcal{M} , the next step is to decide on the parameterisation of \mathcal{M} .

In processes where transport delays exist, the model structure needs to incorporate the pure delay time of the system in order to capture the process dynamics. The question of dealing with transport delays when identifying state space models is further considered in Chapter 4.

2.3.3 The identification procedure I

With the model structure \mathcal{M} decided upon, an identification method I is then chosen to find the parameters to best fit the data. For example, a least squares procedure can be used, however, several choices of I and \mathcal{M} can be tried on the same data set until a satisfactory result is obtained [5]. In the case where ordinary least squares (OLS)

performs poorly because of highly correlated industrial data, partial least squares (PLS) or recursive least squares (RLS) provide alternative procedures that avoid the matrix inversion problem that plagues OLS. To obtain a good model with a reasonable amount of work - the user tries various structures and parameterisations before deciding on the best one. A degree of trial and error is required, combined with past experience and knowledge of the process. From a condition monitoring point of view, it is not the accuracy of the model alone which determines its usefulness, but especially its ability to detect and diagnose faults.

All data dealt with in this study is first scaled to unit variance and zero mean. Scaling the data is important for a number of reasons; it provides numerical conditioning and also compensates for the range of engineering units that are invariably encountered when dealing with industrial process data. Scaling of the model residuals is also important when comparing and contrasting model accuracy, again because of the range of engineering units under consideration.

A model should generalise well. Beyond a certain optimum number of parameters, a further increase in model complexity only leads to the model fitting the noise in the data, so that it becomes less accurate on unseen data. Several methods [6] have been developed to obtain the optimum number of model parameters that should be used, e.g. cross-validation, which involves dividing the data into two sets (or more) and using one for training the model and the second for validating the model according to a pre-specified optimisation criterion. In the case of limited data being available, it has been shown that "leave-one-out" cross-validation is statistically the optimum way to parameterise a model $M(\theta)$. For a given N data points, this involves choosing $N-1$ points (hence "leave-one-out") and then calculating the prediction error for the final point. The procedure is repeated N times and the results averaged to find the optimum model. An alternative to cross-validation is the Akaike Information Criterion (AIC) [7]. This provides a quantitative assessment of the trade-off between model complexity and prediction error. The model complexity is determined by the number of free parameters M_k in the model structure. The aim is to find a parsimonious representation, that gives a good fit to the data $AIC = f(M_k, \text{prediction error})$.

A first evaluation of the accuracy of a parametric model is to plot its step response against the process step response, where the residuals are the difference between the predicted values, $\hat{y}(k)$, and the measured values $y(k)$, of the outputs,

$$\varepsilon(k, \theta) = y(k) - \hat{y}(k, \theta). \quad (2.5)$$

The optimum model M_{opt} is the solution that finds the values of θ that minimise an optimisation criterion $V(\theta)$, where Λ is a positive definite matrix that allows for certain outputs to be weighted for, according to the relative importance of their accuracy,

$$V(\theta) = \sum_{k=1}^N \varepsilon(k, \theta)^T \Lambda \varepsilon(k, \theta). \quad (2.6)$$

Equation 2.6 represents a non-linear least squares optimisation problem, on which a variety of techniques have been used [3]. When cross-validation is used, first a model structure $M(\theta)$ is chosen, then the number of parameters is varied so as to minimise the objective function (2.6) on the training data set. The model is then tested on one or more validation data sets for final assessment. Once an analytical solution to the parameterisation problem is found, an iterative search may be used to further optimise the parameters of $M(\theta)$. Subspace methods complement the non-linear optimisation methods because they calculate an analytical solution that is near optimal [8]. The user then has the option of attempting further optimisation using a prediction error method (PEM) based on Eq. 2.6, although an improvement in accuracy is not guaranteed. Finding an appropriate model is generally part of an iterative scheme, so that even after a model that seems acceptable according to the optimisation criteria $V(\theta)$ has been found, the user may choose to go back to the first step and try an alternative model structure $M(\theta)$ or an alternative parameterisation θ .

Several statistical tests can be used to evaluate and compare the quality of a set of models. One test is to calculate the covariance between the residuals and the past inputs

$$\hat{R}_{\varepsilon u}(k) = \frac{1}{N} \sum_{m=1}^N \varepsilon(m) u(m-k). \quad (2.7)$$

The covariance values $\hat{R}_{\varepsilon u}(\kappa)$ can be plotted for various values of κ to check for correlation between the inputs and residuals. In addition to evaluation of prediction error and residual analysis there is a further issue to consider - does the model fulfil the need for which it is intended? For example, in the context of process condition monitoring, a model with significant bias may still be able to detect and locate faults in terms of sudden changes in the squared prediction error (SPE).

2.3.4 The Experimental Condition C

The experimental condition C defines how the identification is carried out. This involves setting up the experiment with consideration for an appropriate operating range and choosing appropriate excitation for the process. A question is asked: Is the system stable and is the experiment to be carried out in closed-loop or open-loop? Other issues include setting the sampling rate, deciding upon the option of pre-filtering, data collection, defining training and validation data sets and removing outliers. In addition, the input signal must have a rich enough spectral content [9]. Clearly the choice of excitation, and where it is applied, has a substantial influence on the process measurements. This will determine how each of the modes of the process is excited and also the operating point of the process. It is important that all the modes of interest in the plant are properly excited. Careful choice of an appropriate operating point is also necessary if a non-linear system is to be linearised about a certain operating point.

For industrial processes it may not be possible to manipulate the process in production mode and even if this is possible there will generally be restricting conditions for applying a system identification experiment. However, when using simulations, the user has the freedom to choose any input sequence deemed appropriate to excite the dynamics of the system. This may be filtered Gaussian white noise, random or pseudo-random binary signals (PRBS), step inputs or sums of sinusoids [6]. Assuming a linear response, increasing input power leads to more accurate parameter estimates [6]. When the signal to noise ratio rises, the variance of the identified model parameters (assumed to be a normally distributed random variable with mean $\mu_{I, M(\theta)}$ and standard deviation $\sigma_{I, M(\theta)}$) is reduced. However there are generally tight restrictions on the maximum permissible input amplitudes that may be applied to a process. For a given input

amplitude, the crest factor, C_r , is a measure of relative signal power. For a zero mean input signal $u(t)$

$$C_r^2 = \frac{\max(u^2(t))}{E(u^2(t))}. \quad (2.8)$$

Zero mean, binary, symmetric signals such as PRBS have a crest factor of 1, the theoretical lower bound for C_r , and as such they are ideal candidates for many identification experiments because they maximise the input signal power. In the industrial case, it may be possible to apply a PRBS signal about the mean operating point of each of the system inputs. For an in-depth discussion of the advantages and disadvantages of various system excitation strategies see [5, 6].

Another consideration is that the model is required to operate over certain frequency ranges. The sampling rate can be calculated according to the rise time of the fastest dynamics of interest [6]. The optimum choice for sampling rate will lie in the range of the (unknown) time constants of the system. As a heuristic: ten times the band width of the system is about right, although there may be the option of sampling as fast as possible during the experiment and then digitally pre-filtering the data [6].

2.4 Model Structures

Three families of linear model structures are used in this study. They are State Space, Finite Impulse Response (FIR) and Autoregressive with Exogenous Variables (ARX) model structures. Each model structure is characterised as $M(\theta)$. In Chapters 4 and 5 the usefulness and applicability of subspace methods for system identification has been demonstrated by benchmarking the performance of the subspace methods against industry standard methods, using FIR and ARX model structures.

2.4.1 ARX Model Structure

ARX models are identified using “black box” methods [5]. Consider a LTI, SISO system sampled at regular sampling interval T , for notational simplicity assume ($T=1$).

amplitude, the crest factor, C_r , is a measure of relative signal power. For a zero mean input signal $u(t)$

$$C_r^2 = \frac{\max(u^2(t))}{E(u^2(t))}. \quad (2.8)$$

Zero mean, binary, symmetric signals such as PRBS have a crest factor of 1, the theoretical lower bound for C_r , and as such they are ideal candidates for many identification experiments because they maximise the input signal power. In the industrial case, it may be possible to apply a PRBS signal about the mean operating point of each of the system inputs. For an in-depth discussion of the advantages and disadvantages of various system excitation strategies see [5, 6].

Another consideration is that the model is required to operate over certain frequency ranges. The sampling rate can be calculated according to the rise time of the fastest dynamics of interest [6]. The optimum choice for sampling rate will lie in the range of the (unknown) time constants of the system. As a heuristic: ten times the band width of the system is about right, although there may be the option of sampling as fast as possible during the experiment and then digitally pre-filtering the data [6].

2.4 Model Structures

Three families of linear model structures are used in this study. They are State Space, Finite Impulse Response (FIR) and Autoregressive with Exogenous Variables (ARX) model structures. Each model structure is characterised as $M(\theta)$. In Chapters 4 and 5 the usefulness and applicability of subspace methods for system identification has been demonstrated by benchmarking the performance of the subspace methods against industry standard methods, using FIR and ARX model structures.

2.4.1 ARX Model Structure

ARX models are identified using “black box” methods [5]. Consider a LTI, SISO system sampled at regular sampling interval T , for notational simplicity assume ($T=1$).

Assume that all the dynamics of the system are properly excited and measured. An ARX model structure for the system is the linear difference equation

$$y(t) = \sum_{i=1}^n a_i y(t-i) + \sum_{j=1}^p b_j u(t-L-j) + e(t). \quad (2.9)$$

Equation 2.9 relates the current value of the system output, $y(t)$, as a function of the previous n outputs, and p previous inputs starting with the L^{th} previous input. L is the time it takes for the output to begin to respond to the input (the transport delay). The model structure $M_{ARX}(\theta)$ is entirely defined by the parameters n , p and L . The system has n poles and p zeros, and the total number of parameters required for this SISO model is $n+p$. For the MIMO case, with m inputs and l outputs, each output $y_r(t)$ is defined in a similar way to Eq. 2.9 so that a MIMO ARX model is of the form

$$y_r(t) = \sum_{i=1}^n a_i y_r(t-i) + \sum_{g=1}^m \sum_{j=1}^p b_{j,g} u_g(t-L-j) + e_r(t). \quad (2.10)$$

While it is possible to specify different values for n , p and L for each output, they are often set to the same value. One disadvantage of ARX models is the number of user defined choices required to fully specify the model structure. This may involve trying a large number of different models. Such calculations are well within the capability of modern computers, however, for large-scale MIMO systems, the number of possible models could be prodigious. In contrast, state space models provide parsimonious representations where only the model order needs to be specified.

ARX models are a subset of the more general Box Jenkins Model Structure [10]. For a SISO model, the Box Jenkins structure is

$$y(t) = \left(\frac{B(q)}{A(q)} \right) u(t-k) + \left(\frac{C(q)}{D(q)} \right) e(t). \quad (2.11)$$

A , B , C , and D are polynomials in the backwards shift operator q and $e(t)$ describes the unmeasured disturbances to the system and is usually considered to be zero mean, random and normally distributed.

ARX models have $A(q)$ monic, $A=D$ and $C=1$, i.e.

$$A(q)y(t) = B(q)u(t-k) + e(t). \quad (2.12)$$

Note the subtle difference between Eq. 2.12 and Eq. 2.9. The noise term now affects the autoregression terms meaning that Eq. 2.12 is an equation error model structure [6].

2.4.2 ARX Identification Procedure

The coefficients of the polynomials A and B in Eq. 2.12 can be found using a linear regression. First write Eq. 2.9 as

$$\hat{y}(t) = \sum_{i=1}^n a_i y(t-i) + \sum_{j=1}^p b_j u(t-k-j). \quad (2.13)$$

The measured data sequences are arranged into the appropriate regression matrices, and then an over-determined system of equations is solved for the unknown θ , where θ contains the coefficients of the polynomials A and B for the model prediction Eq. 2.13 i.e.

$$\mathbf{Y} = \Phi \hat{\theta}. \quad (2.14)$$

For a MIMO system, with m inputs, l outputs and N data points, an ARX model with n autoregressive terms, p lagged inputs, and a pure time delay of k samples,

$$\mathbf{Y} \in \mathbb{R}^{N-k-p+1 \times l} \quad \Phi \in \mathbb{R}^{N-k-p+1 \times (mp+nl)}, \quad \hat{\theta} \in \mathbb{R}^{mp+nl \times l}.$$

The ordinary least squares solution to Eq. 2.14 is

$$\hat{\theta} = (\Phi^T \Phi)^{-1} \Phi^T \Phi \mathbf{Y}. \quad (2.15)$$

In the case of highly correlated process data, the expression $\Phi^T \Phi$ in Eq. 2.15 may be singular or poorly conditioned, leading to difficulty in finding the inverse. One way to avoid this problem is to avoid the explicit calculation of $(\Phi^T \Phi)^{-1}$, either by using recursive least squares (RLS) or PLS, or alternatively, solve Eq. 2.14 using an orthogonal matrix approach such as QR or SVD [11].

2.4.3 FIR Model Structure

The Finite Impulse Model (FIR) is a commonly used modelling approach for industrial processes with time delays. It is robust, and easy to understand and apply, however, for systems with long delays and/or settling times, many parameters are required to describe the system response. For a SISO system, the FIR model structure is a difference equation of the form

$$y(t) = \sum_{p=1}^m b_p u(t - L - p) + e(t). \quad (2.16)$$

p is a measure of the delay spread and L is the pure time delay in the system. This is an “output error” model structure [6] with the white noise term $e(t)$ representing the mismatch between the predicted and measured output. As with the ARX model, a least squares estimate yields an unbiased estimate if $e(t)$ is a normally distributed random variable with zero mean. The FIR model structure belongs to the Box Jenkins group of model structures and is a special form of the ARX model (Eq. 2.12) with $A=1$:

$$y(t) = B(q)u(t - k) + e(t). \quad (2.17)$$

The parameters to fit the FIR model structures can be found using the same linear regression techniques that are applied to the ARX problem described above.

2.4.4 State Space Model Structure

This study deals with the identification of state space models using subspace system identification methods. In this section, the structure of linear state space models in discrete time is presented. State space models have been applied to a great variety of problems since the groundwork of Kalman [12] in the 1960s on prediction and linear quadratic control. They have proved useful for human endeavour in a range of fields; for modelling and simulating a wide range of physical processes, with applications including control engineering, econometrics and dynamics [1]. The properties of state space models are well understood, and a large amount of control and systems literature exists for them [17].

Assuming a stable LTI system for which there is input-output data sampled at discrete instants, kT , $k = 1, 2, 3, \dots$. The discrete-time state space model is written as a system of first order difference equations. A deterministic system of finite dimension, n , may be described by the n^{th} order state-space model:

$$\begin{aligned} \mathbf{x}_{(k+1)T} &= \mathbf{A}\mathbf{x}_{kT} + \mathbf{B}\mathbf{u}_{kT} \\ \mathbf{y}_{kT} &= \mathbf{C}\mathbf{x}_{kT} + \mathbf{D}\mathbf{u}_{kT} \end{aligned} \quad (2.18)$$

The parameters to be identified are the system matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$. These system matrices describe the evolution of the state vector $\mathbf{x}_{kT} \in \mathbb{R}^n$ over time using n first order difference equations. For an n^{th} order MIMO system, with m inputs and l outputs, then $\mathbf{u}_{kT} \in \mathbb{R}^m$, $\mathbf{y}_{kT} \in \mathbb{R}^l$, $\mathbf{x}_{kT} \in \mathbb{R}^n$, $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{B} \in \mathbb{R}^{n \times m}$, $\mathbf{C} \in \mathbb{R}^{l \times n}$, $\mathbf{D} \in \mathbb{R}^{l \times m}$. Only the input and the output variables are measured, this implies that the state variables need to be estimated in some way, in order to identify $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$. On occasions, the states of the system may have physical meaning (e.g. position and velocity of a spacecraft) however Kalman [13] stresses that the states of the process are to be viewed as an abstract quantity. The state of the system is “the minimal amount of information about the past history of the system which suffices to predict the effect of the past on the future”.

Assuming a sampling interval $T=1$, Eq. 2.18 becomes

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k \\ \mathbf{y}_k &= \mathbf{C}\mathbf{x}_k + \mathbf{D}\mathbf{u}_k \end{aligned} \quad (2.19)$$

The description (2.19) is not unique. The state basis is changed by any non-singular matrix $\mathbf{T} \in \mathbb{R}^{n \times n}$, without affecting the input - output behaviour of the system. By substituting $\mathbf{z}_k = \mathbf{T}\mathbf{x}_k$, Eq. 2.19 can also be expressed as

$$\begin{aligned} \mathbf{z}_{k+1} &= \bar{\mathbf{A}}\mathbf{z}_k + \bar{\mathbf{B}}\mathbf{u}_k \\ \mathbf{y}_k &= \bar{\mathbf{C}}\mathbf{z}_k + \bar{\mathbf{D}}\mathbf{u}_k \end{aligned} \quad (2.20)$$

It follows directly from Eq. 2.20 that

$$\bar{\mathbf{A}} = \mathbf{T}\mathbf{A}\mathbf{T}^{-1}; \quad \bar{\mathbf{B}} = \mathbf{T}\mathbf{B}; \quad \bar{\mathbf{C}} = \mathbf{C}\mathbf{T}^{-1}; \quad \bar{\mathbf{D}} = \mathbf{D}. \quad (2.21)$$

The transfer function $\mathbf{H}(q)$ remains the same for the different state space realisations Eq. 2.20 and Eq. 2.21:

$$\mathbf{H}(q) = \mathbf{C}(q\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}. \quad (2.22)$$

A realistic model of an industrial system needs to take into account the effect of noise. This may come from a variety of sources – including uncertainty due to measurement error, process irregularities and the effect of unmeasured disturbances. The following linear stochastic model of the system is therefore proposed:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{v}_k \\ \mathbf{y}_k &= \mathbf{C}\mathbf{x}_k + \mathbf{D}\mathbf{u}_k + \mathbf{w}_k \end{aligned} \quad (2.23)$$

The stochastic noise sequences \mathbf{v}_k and \mathbf{w}_k are from a least squares point of view, the model residuals. A condition for unbiased estimates is that \mathbf{v}_k and \mathbf{w}_k are uncorrelated, zero mean, white noise processes.

Eq. 2.23 can also be expressed in the innovations form [1]:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{K}\mathbf{e}_k \\ \mathbf{y}_k &= \mathbf{C}\mathbf{x}_k + \mathbf{D}\mathbf{u}_k + \mathbf{e}_k \end{aligned} \quad (2.24)$$

where \mathbf{K} is the Kalman gain [14].

2.4.4.1 Identifying a state space model using least squares

Consider the case where the state vectors are known, for example the measured position and velocity of an object. Then write Eq. 2.19 as

$$\begin{bmatrix} \mathbf{X}_{k+1} \\ \mathbf{Y}_k \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{X}_k \\ \mathbf{U}_k \end{bmatrix}. \quad (2.25)$$

The respective vector sequences are defined as

$$\begin{aligned} \mathbf{X}_{k+1} &= (\mathbf{x}_{k+1} \quad \mathbf{x}_k \quad \cdots \quad \mathbf{x}_1), \\ \mathbf{Y}_k &= (\mathbf{y}_{k+1} \quad \mathbf{y}_k \quad \cdots \quad \mathbf{y}_0), \end{aligned} \quad (2.26)$$

$$\mathbf{X}_k = \begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}_{k-1} \end{bmatrix},$$

$$\mathbf{U}_k = \begin{bmatrix} \mathbf{u}_k \\ \mathbf{u}_{k-1} \end{bmatrix}.$$

Eq. 2.25 can be written as

$$\mathbf{Y} = \Theta \Phi. \quad (2.27)$$

Since all the values in Eq. 2.25 are known, Eq. 2.27 is an over determined system of equations with unknowns $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$, that can be solved using an ordinary least squares procedure. Unfortunately, in the case of industrial process data, the states are abstract quantities that bear little relation to the external measurements of plant. Therefore the states can't be measured. However it is possible to find a valid representation of the state sequences using the subspace methods described in Chapter 3. In the next chapter, a full explanation of subspace theory is provided. Three algorithms that estimate the states from the external measurements of the system will be outlined.

2.5 Conclusion

In this chapter, a brief introduction to the field of system identification has been provided. The field has been classified into 4 constituents: The system S , The model structure

identification procedure is in identifying the unknown state sequences. In the next chapter, subspace system identification methods are described. These methods provide a means for calculating the states directly from the system measurements.

In Chapter 4, the performance of subspace and time-series models is compared on a simple mass-spring-damper system, then finally, in Chapter 5, on a more complex industrial simulation. In Part Two, dynamic process condition monitoring models will be considered. A subspace model is used as a condition monitor and its performance is compared to currently available methods in monitorMV.

Chapter 3

Subspace System Identification

An outline of subspace system identification is provided. The chapter begins with a literature review that provides a brief summary of the main contributions to the development of the methodology. State space realisation theory is presented as a precursor to the development of subspace system identification algorithms. A brief summary of the CVA and MOESP algorithms is presented, followed by a more in depth treatment of the N4SID identification procedure. The N4SID algorithm provides the platform on which the subspace model for process condition monitoring is built, in the second part of the dissertation.

3.1 Introduction

Subspace system identification methods identify LTI state space models of systems as a function of discrete time. In some cases, state space models can be built with knowledge of the physical process in mind. This approach usually involves the construction of a continuous time state space model based on Newton's laws of physics. However the task to build a useful model of a continuous industrial process, with many inputs and outputs, would be prodigious if not impossible (given time constraints). The subspace approach enables easy identification of linear state space models of complex MIMO

processes, by adopting a black box approach. In the first step, the process outputs are modelled as a linear function of the past inputs and outputs of the process. In the second step, a singular value decomposition (SVD) is calculated and used to estimate the model order best to describe the system. Finally, the state sequences are estimated, and then the model parameters are calculated. The use of the SVD leads to a balanced realisation of the process [15, 16], which makes it easy for model order reduction. This leads to low order models, that are useful for process condition monitoring because they greatly reduce the number of dimensions that need to be monitored. The use of SVD also provides the identification procedure with robust numerical properties at low computational cost [17].

3.2 Literature Review

Subspace identification methods have attracted a great deal of research interest over the past decade. The three main algorithms are the canonical variate analysis (CVA) method of Larimore [18-20], the MIMO output-error state space model (MOESP) of Verhaegen [21-24], and the N4SID algorithm of Van Overschee [1, 25]. These subspace methods and other variations of them, are closely connected to the state space realisation theory of the 1960's [6, 17]. One of the earliest of the realisation algorithms was Ho's Algorithm [26], which delivers a method for recovering the state matrices from a Hankel matrix, constructed from impulse response data of a linear deterministic system. A rank analysis of this Hankel matrix, containing the Markov parameters, reveals the order of dynamics of the system. Then the Hankel matrix is factorised to reveal the state matrices. Zeiger [27] proposed the singular value decomposition (SVD) in the case where the impulse response matrix data is contaminated by noise, and hence no exact determination of the rank (and therefore the system order) is possible. Kung [28] highlighted the utility of the SVD as a practical tool for approximate linear realisation of stable linear systems. Particularly attractive is that the Hankel matrix reflects the order of the system, and that the SVD displays a set of singular values that not only reflect the order of the system, but that the magnitudes of the singular values give a quantitative measure of the distance of the matrix to a lower rank one, for example, an r^{th} order approximation suggests that the level of noise on the system is of the order of the $(r + 1)^{th}$ singular value. Kung also demonstrates that the SVD provides

an internally balanced realisation, which is very useful for determining lower order models. Kung's algorithm extends Ho's algorithm to the case where the excitation is Gaussian noise, in which case the Markov parameters can be obtained by calculating the correlation between the measured input and output sequences, leading to identification based on an approximate impulse response sequence.

The subspace methods of the last decade [1, 18-25] have been shown to have much in common with the realisation methods described above [17]. In common with Ho's and Kung's algorithms, they construct Hankel matrices from external measurements of the system. An SVD factorisation is applied which reveals quantitatively the likely order of the system dynamics, then the shift invariance structure of the left singular vectors is used to calculate the observability matrix of the system.

Subspace methods develop realisation theory further by extending the analysis to systems with any combination of inputs, including measurement noise and/or process noise. One of the earliest of the so-called subspace methods to be published is the CVA algorithm [18] of Larimore (1990). He built on from the work of Akaike [29], who constructed a minimal realisation procedure for a Markov process using canonical correlation analysis (CCA). A second distinct family of subspace algorithms is Verhaegen's [24] Multivariable Output-Error State Space (MOESP) class of algorithms, based on an instrumental variables approach (1992). The MOESP algorithms are different in that the key subspace identified is the extended observability matrix rather than an explicit state sequence. MOESP uses an RQ decomposition and the column space of specific sub-matrices of the R factor, to approximate the column space of the extended observability matrix. Verhaegen [30] points out that CVA [18] treats the identification problem in a statistical setting, whereas the N4SID algorithm [25] derives the solution in terms of standard linear algebra. N4SID, which is an abbreviation for "Numerical Method for Subspace System Identification", belongs to a family of algorithms that are grouped together under the banner of "4SID" [8] and was published by Van Overschee and De Moor (1994). 4SID methods calculate the state sequences from an explicit projection of the future output Hankel matrix onto a compound matrix of the past and the future input Hankel matrices. It is shown in [31], that the initial calculation in N4SID corresponds to a least squares fit for an ARX model structure, where each of the rows of the projection, $\hat{\mathbf{Y}}_{future}$, are k -step ahead predictors. Overschee

also proves in [25] that the state sequences calculated using N4SID are equivalent to the states of a non-steady state Kalman filter, i.e. they are optimum predictions in the sense that the covariance of the state error is minimised. These three algorithms are summarised in the following sections, with particular emphasis on N4SID, then in the second part of the dissertation, a 4SID method is adapted for use as a condition monitor.

A general and comprehensive overview of the subspace methods appears in the essay by Viberg [17]. Also useful is the summary of subspace methods given by Ljung [6] and Favoreel [8]. Viberg [17] provides an introduction to state space realisation theory and a description of the aforementioned algorithms from a statistical point of view (i.e. their consistency in the face of noise). He goes on to cast each of the subspace methods into an instrumental variables (IV) framework. Overschee [32] provides a “unifying theorem” for the CVA, MOESP and N4SID algorithms. He proves that each of the algorithms uses weightings of exactly the same subspace to estimate the order of the system and to identify the controllability matrix $f(A, C)$. In the initial “projection” stage, CVA uses canonical correlations, whereas MOESP uses an RQ decomposition to compress the data into orthogonal subspaces, while N4SID uses a least squares projection. The unifying theorem [32] demonstrates that each algorithm is based on weighted versions of the same subspace, which is defined by covariance matrices of the input-output Hankel matrices. Favoreel [8] compared the performance of N4SID with respect to prediction error methods (PEM) [6], on the basis of ten industrial data sets. He concluded that N4SID was sub-optimal with respect to PEM in some cases, however, he described the following attractive qualities of subspace methods: they deliver an analytical solution, thereby avoiding the computational uncertainties of the non-linear optimisation problem; they yield near optimal predictions, where a single decision as regards the model order leads to the identification of a parsimonious state space model. Favoreel concludes that subspace and PEM methods are complementary, for example, where appropriate, N4SID could provide an initial estimate, and then further optimisation be attempted using PEM. In Chapter 4, N4SID will be compared directly with MOESP, and a time-series model structure (ARX), on the basis of a simple mass-spring-damper system. The results will be used to highlight various aspects associated with subspace system identification user choices, and as a vehicle for measuring the state space model against its closest relative, the ARX model structure.

DeMoor [33] developed two algorithms; the first uses least squares for systems with noise only on the outputs; the second uses total least squares to deal with noise on both inputs and outputs. He stresses the importance of “long” data sequences for accurate identification. Bauer [34, 35] discusses user choices in running subspace methods. He investigates the ideal number of block rows used in the initial projections and also compares methods for automating the model order selection, based on the magnitude of the singular values. In [35], Bauer analyses the effect of data pre-processing methods on the variance of the estimates. He shows that the removal of trends or periodic components from the data affects the variance of the estimates in the same way as if extra signals are added as inputs. Delgado [36] presents a technique for implementing N4SID online. He uses a Householder transformation [11] to update the initial projections, and also shows how the left singular vectors of the SVD can be computed in a recursive way, then finally the state sequences are calculated using RLS.

Three papers that treat the problem of uncertainty in both input and output measurements, known as the error-in-variables (EIV) problem, are those of Moonen [37], Chou [38] and Gustafsson [39]. In [37], two schemes are considered. The first treats the case where inputs and outputs are corrupted with white noise. The solution is described in terms of finding the intersection between the row spaces of the past and future Hankel matrices. A second SVD is applied to the rank deficient intersection in order to estimate the system order. The second scheme deals with identification problems involving pre-filtered data of the system. This leads to the analysis of data containing coloured noise sequences of known coloration, due to the characteristics of the filter. Moonen [40] delivers a revised computational procedure for [37], so that the algorithm gives a balanced realisation. Chou [38] delivers a EIV subspace algorithm, which he claims gives unbiased estimates in open or closed-loop operation. Extending the work in [37], he describes the solution in an IV framework. Chou generalises to the case where there is input measurement error, and the output is corrupted by a sum of white measurement noise and white process noise. Gustaffson [39] describes an EIV method in an IV framework, where he suggests using a weighting matrix which improves the accuracy of [38].

Ljung [31, 41] gives a least squares interpretation of subspace methods. These papers describe a defining feature of subspace methods, compared with traditional approaches to state space modelling: they calculate the state sequences *before* estimating the state

matrices. In [31, 41], it is shown that the k^{th} row of the initial projection in N4SID is a k -step ahead prediction based on an ARX-type model structure (identified using least squares). Ljung describes the SVD as the numerical technique that provides a robust method for choosing the linear combinations of the rows that determine the state sequence. Finally, once the states are known, determining the state matrices is reduced to a simple linear regression. In [42] Ljung uses the results from [31, 41], to show why the k -step predictors produce biased results when operating in closed-loop. He suggests a method for calculating the k -step predictors recursively, based on a one-step predictor to overcome this problem.

Verhaegen [43] describes the application of his MOESP algorithm in closed loop. He casts the problem in an open loop framework by injecting reference signals into the loop, and calling these the system inputs. He then develops a global model based on the response of the system to these reference signals. He demonstrates that information concerning the form of the controller isn't necessary for his scheme to work. However several challenges exist, such as the case of unstable pole-zero cancellations between the controller and the plant, however this is not considered a problem for much of industrial plant – where many systems are open-loop stable. Overschee [44] suggests an alternative solution for the closed-loop subspace identification problem. He addresses some of the drawbacks of Verhaegens method [43] that arise due to identifying a global model: e.g. in the case where the order of the controller is high, such that the joint plant-controller model leads to a very complicated problem, and also difficulties associated with the model-reduction step. Overschee's method assumes that a set of Markov parameters are available for the controller. He then casts the entire problem in a N4SID framework, where he uses the lower Toeplitz matrix containing the Markov parameters of the controller to create the necessary instruments for an IV approach. Subspace methods [1, 18-25] rely on the measurement and noise sequences being uncorrelated with the states and the system inputs. This condition is violated due to the presence of feedback. Overschee shows how the Markov parameters of the controller can be used to create instruments that remove the effect of the future inputs from the initial projection.

In [45], Overschee describes an algorithm for determining the state matrices when the system is driven only by unmeasured disturbances – the so-called stochastic identification problem. He proves that the estimated state sequences correspond to the predictions of a non-steady state Kalman filter. This algorithm contains some of the

results that are later used in the development of N4SID. In the case where few time lags are used in the building of the input-output Hankel matrices, the new algorithm performs better than Akaike's CCA method [46] and Larimore's CVA [19]. In [47] Overschee further develops his work on the unifying theorem [32] by demonstrating that each of the algorithms [18, 24, 25] uses weighted projections that are frequency weighted balanced according to the framework presented in Enns [48]. This has important consequences regarding a comparative analysis of the performance of the algorithms in the frequency domain.

Maciejowski [49] points out that the distinguishing feature of the subspace methods is that a state sequence is estimated before the system matrices are identified. He suggests a method for guaranteeing the stability of the A matrix, by guaranteeing that the eigenvalues of the estimate of A are less than or equal to unity. Ottersten [50] also presents a method for estimating the poles of the system, where the poles are calculated directly from the extended observability matrix, without explicitly calculating A . He obtains the extended observability matrix using an IV method that is similar, in essence, to N4SID.

In his tutorial survey paper of CVA, Larimore [18] describes CVA as a "black box method" that makes no assumption concerning the system order. He cites advantages of "batch" identification, e.g. that there is no need for a start-up procedure, then asserts that the AIC criterion [7] guarantees optimum estimation of the model order (provided enough data is present). Larimore addresses the issue of adaptive models for the cases where the system dynamics change (slowly) with time. He illustrates the effectiveness of CVA using an aerospace application and two applications involving chemical processes.

Verhaegen [30] describes two further identification schemes that are presented in an IV framework; the first treats the case where the input is white noise and the second treats the more general input case (subject to the requirements for well designed identification experiments). In [51] Verhaegen presents a recursive implementation of his MOESP algorithm. He shows how exact updates to an online model can be applied in an adaptive context. He points out that the algorithm could be useful where RLS schemes are currently employed.

Juang [52] proposes the eigensystem realisation algorithm (ERA) to deal with noisy data and inexact measurements of impulse response data. Cooper describes several time series methods including an improvement to the ERA method [9, 53]. Hunter [54] contrasts the CVA and ERA methods, and provides a direct comparison of their performance on the basis of a high order state space simulation.

The preceding literature review indicates that subspace system identification has attracted a great deal of research interest over the past few years. However most of the published literature has centred on a theoretical treatment of the subject. The following section provides some background to the development of subspace theory; then popular subspace algorithms are summarised, followed by a more in depth treatment of the N4SID subspace system identification method. In Chapters 4 and 5 the methodology is investigated from a practical point of view, using both simulated and real industrial data.

3.3 Realisation theory

In this section, Ho's algorithm [26] and Kung's algorithm [28] are outlined. These algorithms identify the state matrices from Hankel matrices built up from the external measurements of the system. They are the precursors of subspace methods with which they have the following three operations in common: (1) The formation of Hankel matrices based on external measurements of the system, (2) the estimation of the system order based on the rank of the "weighted" Hankel matrix, and (3) the use of an orthogonal factorisation (SVD), to obtain an estimation of the controllability matrix of the system.

3.3.1 Realisation theory using impulse excitation

Consider a stable, open loop, LTI system of finite dimension, n , sampled at a regular sampling interval ($T = 1$). The system is deterministic, causal, and is exactly described by the state equations

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k \\ \mathbf{y}_k &= \mathbf{C}\mathbf{x}_k \end{aligned} \quad (3.1)$$

Note that this system is exactly described by the model, i.e. the properties of the model are also the properties of the system. If the assumption is made that the model describes the system *exactly*, then the terms “properties of the model” and “properties of the system” are used interchangeably.

The system (3.1) generates a time series, with input and output vectors of dimension m and l respectively: $\mathbf{U}_k = (\mathbf{u}_k \ \mathbf{u}_{k+1} \ \dots)$ and $\mathbf{Y}_k = (\mathbf{y}_k \ \mathbf{y}_{k+1} \ \dots)$.

The problem is to identify the system matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ belonging to the state space description Eq. 3.1. The aim is to find a minimal realisation, i.e. the lowest order model possible that fully describes the system.

The system is excited by applying an impulse at time instant $(k=0)$. As a direct consequence of the *model structure* (3.1), the output of the model for $(k=1,2,3,\dots)$ is

$$\mathbf{y}_k = \mathbf{C}\mathbf{A}^{k-1}\mathbf{B}. \quad (3.2)$$

These are the so called Markov parameters of the system. A block Hankel matrix $\mathbf{H}_{0,r,N}$ is constructed from the system outputs \mathbf{y}_k , where the first subscript of H refers to the index of first element in H, and the second and third subscripts of H refer to the number of block rows in H and number of columns in H respectively. For Ho's algorithm, $r = N$, so that H contains the first $2r - 1$ Markov parameters of the system.

$$\mathbf{H}_{0,r,N} = \begin{bmatrix} \mathbf{y}_0 & \mathbf{y}_1 & \dots & \mathbf{y}_{N-1} \\ \mathbf{y}_1 & \mathbf{y}_2 & \dots & \mathbf{y}_N \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{y}_{r-1} & \mathbf{y}_r & \dots & \mathbf{y}_{N+r-2} \end{bmatrix} = \begin{pmatrix} \mathbf{C} \\ \mathbf{C}\mathbf{A} \\ \vdots \\ \mathbf{C}\mathbf{A}^{r-1} \end{pmatrix} (\mathbf{B} \ \mathbf{A}\mathbf{B} \ \mathbf{A}^2\mathbf{B} \ \dots \ \mathbf{A}^{N-1}\mathbf{B}),$$

or

$$\mathbf{H}_{0,r,N} = \mathbf{\Gamma}_r \mathbf{\Omega}_r, \quad (3.3)$$

where

$$\mathbf{\Gamma}_r = \begin{pmatrix} \mathbf{C} \\ \mathbf{C}\mathbf{A} \\ \vdots \\ \mathbf{C}\mathbf{A}^{r-1} \end{pmatrix}, \quad (3.4)$$

and

$$\mathbf{\Omega}_r = (\mathbf{B} \quad \mathbf{AB} \quad \cdots \quad \mathbf{A}^{r-1}\mathbf{B}). \quad (3.5)$$

$\mathbf{\Gamma}_r$ is defined as the extended observability matrix, and $\mathbf{\Omega}_r$ is defined as the extended controllability matrix.

A key property of \mathbf{H} is that $\text{rank}(\mathbf{H}_{0,r,N}) = n$, for all $r \geq n$, such that a minimal realisation of the n^{th} order system has $r = n$. Furthermore, a full rank factorisation is possible, $\mathbf{H}_n = \mathbf{\Gamma}_n \mathbf{\Omega}_n$, that yields $\mathbf{\Gamma}_n$ and $\mathbf{\Omega}_n$ up to within a similarity transformation of the system equations Eq. 3.1.

The system matrices \mathbf{B} and \mathbf{C} are read directly from the first m columns of $\mathbf{\Gamma}_r$ and the first l rows of $\mathbf{\Omega}_r$ respectively. \mathbf{A} is computed by exploiting the shift invariant structure of $\mathbf{\Gamma}_r$: delete the first and last block rows from $\mathbf{\Gamma}_r$ to form the following two matrices, $\mathbf{\Gamma}_{2:r}$ and $\mathbf{\Gamma}_{1:r-1}$. The system matrix \mathbf{A} is then calculated using linear regression to solve

$$\mathbf{\Gamma}_{2:r} = \mathbf{\Gamma}_{1:r-1} \mathbf{A}, \quad (3.6)$$

i.e.

$$\mathbf{A} = \mathbf{\Gamma}_{1:r-1}^\dagger \mathbf{\Gamma}_{2:r}, \quad (3.7)$$

where $(\cdot)^\dagger$ denotes the pseudoinverse (since $\mathbf{\Gamma}$ in general, is not square).

Ho's algorithm provides the theoretical basis for determining the state matrices based on the exact measurements of the impulse response (Markov parameters). However the algorithm does not address the practical problem of dealing with noisy measurements. Zeiger [27] delivered the "ultimate weapon" for applying Ho's algorithm to noisy data-sets: SVD can be used to factorise \mathbf{H} . The presence of noise means that \mathbf{H} is full rank, however a lower order model can be identified, where the size of the singular values are used to distinguish the deterministic part of the system from the noise:

$$\mathbf{H}_{0,r,N} = \mathbf{USV}^T, \quad (3.8)$$

where $\mathbf{U} \in \mathbb{R}^{r \times r}$ and $\mathbf{V} \in \mathbb{R}^{r \times r}$ are orthogonal matrices, and $\mathbf{S} \in \mathbb{R}^{r \times r}$ contains the eigenvectors of $\mathbf{H}^T \mathbf{H}$. \mathbf{S} is a diagonal matrix of non negative singular values, in non decreasing order, on the main diagonal [11]. In the case of Ho's algorithm, \mathbf{H} is of rank n , and the last $r-n$ singular values are zero. In the presence of measurement noise, all the singular values are non-zero, however provided the signal to noise ratio is high, the first n singular values will be significantly higher than the final $r-n$. This allows the user to decide the cut off point that determines the system order.

$$\mathbf{H} = [\mathbf{U}_1 \quad \mathbf{U}_2] \begin{bmatrix} \mathbf{S}_1 & 0 \\ 0 & \mathbf{S}_2 \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{bmatrix}, \quad (3.9)$$

or
$$\hat{\mathbf{H}} = \mathbf{U}_1 \mathbf{S}_1 \mathbf{V}_1^T. \quad (3.10)$$

The major dynamics are captured in the first n singular values, while most of the noise is confined to the "minor" singular values. The estimates of the observability and controllability matrices are then

$$\mathbf{\Gamma} = \mathbf{U}_1 \mathbf{S}_1^{1/2}. \quad (3.11)$$

$$\mathbf{\Omega} = \mathbf{S}_1^{1/2} \mathbf{V}_1^T. \quad (3.12)$$

3.3.2 Realisation theory using white noise excitation

Impulse excitation has proven useful for some applications [9], however the method is unsuitable for continuous industrial process applications, for example, due to the highly damped response, impracticality or economic considerations. Kung [28] demonstrated an alternative to impulse excitation of the system: to excite the system using a white noise excitation. A white noise input allows for a finite number of Markov parameters to be estimated from the nonparametric estimates of the cross variance function $r_{yu}^{i,j}(p)$, where the superscripts (i,j) refer to the cross correlation between the i^{th} input and the j^{th} output. Using a well known relation from linear system theory [55]: the output of a LTI system at sample instant k can be fully described by the convolution sum

$$\mathbf{y}_k = \sum_{r=0}^{\infty} \mathbf{H}_r \mathbf{u}_{k-r}. \quad (3.13)$$

If the system (3.1) is corrupted by measurement noise \mathbf{v}_k then:

$$\mathbf{y}_k = \sum_{r=0}^{\infty} \mathbf{H}_r \mathbf{u}_{k-r} + \mathbf{v}_k. \quad (3.14)$$

Assuming the white noise excitation \mathbf{u}_k is uncorrelated with the measurement noise \mathbf{v}_k , then for sufficiently long data sequences the following approximation can be applied:

$$r_{yu}^{i,j}(p) = E(\mathbf{y}_k^j (\mathbf{u}_{k-p}^i)^T) = h_p^{i,j} r_{uu}, \quad (3.15)$$

where

$$E(u_t^i (u_s^i)^T) = r_{uu}^i \delta_{t,s}, \quad (3.16)$$

i.e. the cross variance $R_{yu}(p)$ can be used to estimate \mathbf{H} , the Hankel matrix of the impulse response of the system, according to the relationship

$$H_p = \frac{R_{yu}(p)}{R_{uu}}, \quad (3.17)$$

leading to the full rank matrix $\mathbf{H}_{0,r,r}$, containing the estimated Markov parameters:

$$\mathbf{H}_{0,r,r} \stackrel{\text{def}}{=} \begin{bmatrix} H_0 & H_1 & H_2 & \cdots & H_{r-1} \\ H_1 & H_2 & H_3 & \cdots & H_r \\ H_2 & H_3 & H_4 & \cdots & H_{r+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ H_{r-1} & H_r & H_{r+1} & \cdots & H_{2r-1} \end{bmatrix} \in \mathbb{R}^{lr \times mr}. \quad (3.18)$$

$\mathbf{H}_{0,r,r}$ is factorised using SVD to yield

$$\mathbf{H}_{0,r,r} = \mathbf{U} \mathbf{S} \mathbf{V}^T. \quad (3.19)$$

The user must decide the cut off point that determines the system order. This is considered a non trivial task, with no proven theoretical optimum value. In practice, different choices can be tried, and the results compared using cross-validation [6], however, other methods for determining the optimum model order exist, for example, AIC [7] and BIC [56]. The major dynamics are captured in the first n singular values, while most of the noise is confined to the “minor” singular values. The estimates of the observability and controllability matrices are as before,

$$\mathbf{\Gamma} = \mathbf{U}_1 \mathbf{S}_1^{1/2}. \quad (3.20)$$

$$\mathbf{\Omega} = \mathbf{S}_1^{1/2} \mathbf{V}_1^T. \quad (3.21)$$

The state space model obtained from the aforementioned procedure is balanced [17] in the sense that that

$$\mathbf{\Gamma}^T \mathbf{\Gamma} = \mathbf{\Omega} \mathbf{\Omega}^T = \mathbf{S}_1. \quad (3.22)$$

Moore [15] demonstrated that the principal components analysis of \mathbf{H} leads to the controllability and observability matrices being in a co-ordinate system which is internally balanced. This is ideal for addressing the model order reduction problem. An internal balanced realisation assures that the input-output properties of the model are reflected by the internal principal components, in a way such that in terms of using a lower order model, the infinity norm of the difference between the original and reduced order model is upper bounded by the largest of the neglected weighted singular values [15].

3.4 Subspace Methods

One drawback of the realisation methods outlined above is the difficulty in obtaining accurate impulse response measurements. Generally it is not practical to directly deliver an impulse on major industrial plant, and cross variance analysis using a white noise input presents difficulty due to the noise level in the measurements [52]. However subspace methods can be applied using any type of inputs, subject to the condition of persistent excitation [6]. Subspace methods extract the desired information directly from the data – without explicitly forming the impulse responses. In this section, the three most widely cited subspace methods, CVA [18], MOESP [24], and N4SID [25] are summarised. These have been shown in [32] to calculate weighted versions of the same input-output vector space to determine the order and the extended observability matrix of the system.

The notation used is as follows: if a block Hankel matrix $\mathbf{M}_{a,b,c}$ is described with subscripts a, b, c : then a is the index of the first element of \mathbf{M} ; b is the number of block

rows in \mathbf{M} , and c is the number of columns in \mathbf{M} . The subscript f refers to future and the subscript p refers to the past.

3.4.1 CVA Approach

CVA adopts a statistical approach to solving the system identification problem [30]. Larimore [18] claims to find the optimal combination of the past to predict the future. His method evolved from Akaike's [29] work on canonical correlation analysis (CCA). CVA identifies the deterministic part of LTI systems described by the state space model

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k; \mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{D}\mathbf{u}_k, \quad (3.23)$$

where the outputs \mathbf{y}_k , inputs \mathbf{u}_k and state variables \mathbf{x}_k are of dimension l , m , and n respectively. The system variables are divided into past and future vectors each with finite lags of length L , $p_i^T = (y_{i-L}^T \cdots y_i^T \ u_{i-L}^T \cdots u_i^T)$ and $f_i^T = (y_i^T \cdots y_{i+L-1}^T)$, where the length L is chosen at least as large as the maximum model order required.

CCA is applied to the past and future vectors to find linear combinations of the past vector p_i , that has maximum correlation with linear combinations of the future vector f_i . In effect this involves rotating the past and future vector spaces to find directions with maximum correlation. Larimore calculated a state vector based on past information, to optimally predict the future, as measured by the quadratic prediction error criterion. This method may be viewed as a statistical approach to solving the identification problem, where the covariance and cross-covariance of the data is calculated, then canonical correlations are used to estimate the state sequences X_k and X_{k+1} . Finally a simple least squares regression is applied to calculate the estimate of the state matrices. The method proceeds by first forming Hankel matrices of past and future:

$$\mathbf{F}_{k+1,L,N} = \begin{pmatrix} & & & & y_{k+N} \\ & & & & y_{k+N+1} \\ y_{k+1} & y_{k+2} & \cdots & \cdots & y_{k+N} \\ y_{k+2} & y_{k+3} & \cdots & \cdots & y_{k+N+1} \\ \vdots & \vdots & \cdots & \cdots & \vdots \\ y_{k+L} & y_{k+L+1} & \cdots & \cdots & y_{k+N+L-1} \end{pmatrix} \in \mathbb{R}^{Ll \times N}, \quad (3.24)$$

$$\mathbf{P} = \begin{pmatrix} \mathbf{Y}_{k,L,N} \\ \mathbf{U}_{k,L,N} \end{pmatrix} = \begin{pmatrix} y_k & y_{k+1} & \cdots & \cdots & y_{k+N-1} \\ y_{k-1} & y_k & \cdots & \cdots & y_{k+N-2} \\ \vdots & \vdots & \cdots & \cdots & \vdots \\ y_{k-L+1} & y_{k-L} & \cdots & \cdots & y_{k+N-L} \\ u_k & u_{k+1} & \cdots & \cdots & u_{k+N-1} \\ u_{k-1} & u_k & \cdots & \cdots & u_{k+N-2} \\ \vdots & \vdots & \cdots & \cdots & \vdots \\ u_{k-L+1} & u_{k-L} & \cdots & \cdots & u_{k+N-L} \end{pmatrix} \in \mathbb{R}^{L(m+l) \times N}. \quad (3.25)$$

First estimate the variance and covariance of \mathbf{F} and \mathbf{P} , i.e.

$$\Sigma_{\mathbf{FF}} = N^{-1} \mathbf{F} \mathbf{F}^T, \quad \Sigma_{\mathbf{PP}} = N^{-1} \mathbf{P} \mathbf{P}^T, \quad \Sigma_{\mathbf{FP}} = N^{-1} \mathbf{F} \mathbf{P}^T.$$

The aim is to find the canonical correlations of the past and future [83]. This involves finding the linear combination of the rows of \mathbf{P} that have maximum correlation with linear combination of the rows of \mathbf{F} . This will be the first canonical correlation. Then find the second canonical correlation subject to the condition that it is orthogonal to the first and so on.

Consider the two linear combinations $\boldsymbol{\eta} = \mathbf{a}^T \mathbf{F}$ and $\boldsymbol{\phi} = \mathbf{b}^T \mathbf{P}$. The correlation between $\boldsymbol{\eta}$ and $\boldsymbol{\phi}$ is

$$\rho(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a}^T \Sigma_{\mathbf{FP}} \mathbf{b}}{(\mathbf{a}^T \Sigma_{\mathbf{FF}} \mathbf{a} \mathbf{b}^T \Sigma_{\mathbf{PP}} \mathbf{b})^{1/2}}. \quad (3.26)$$

The value of $\rho(\mathbf{a}, \mathbf{b})$ in (3.26) does not depend of the scaling of the values of \mathbf{a} or \mathbf{b} , therefore \mathbf{a} and \mathbf{b} are found by solving the following optimisation problem:

$$\max_{\mathbf{a}, \mathbf{b}} (\mathbf{a}^T \Sigma_{\mathbf{FP}} \mathbf{b}), \quad (3.27)$$

subject to

$$\mathbf{a}^T \Sigma_{\mathbf{FF}} \mathbf{a} = \mathbf{b}^T \Sigma_{\mathbf{PP}} \mathbf{b} = 1. \quad (3.28)$$

This problem can be formulated in terms of the CVA Theorem [20]. Let $\Sigma_{PP} \in \mathbb{R}^{L(m+L) \times L(m+L)}$ and $\Sigma_{FF} \in \mathbb{R}^{L \times L}$ be covariance matrices based on the past and future Hankel matrices of the input-output data. Then there exists matrices $\mathbf{J} \in \mathbb{R}^{L(m+L) \times L(m+L)}$ and $\mathbf{L} \in \mathbb{R}^{L \times L}$ such that

$$\mathbf{J} \Sigma_{PP} \mathbf{J}^T = \mathbf{I}_{r_{PP}}, \quad (3.29)$$

$$\mathbf{L} \Sigma_{FF} \mathbf{L}^T = \mathbf{I}_{r_{FF}}, \quad (3.30)$$

$$\mathbf{J} \Sigma \mathbf{L}^T = \mathbf{D} = \text{diag}(\gamma_1, \gamma_2, \dots, \gamma_r, 0, \dots, 0). \quad (3.31)$$

where $r_{PP} = \text{rank}(\Sigma_{PP})$, $r_{FF} = \text{rank}(\Sigma_{FF})$, and γ_i are the canonical correlations. Matrices \mathbf{J} and \mathbf{L} are obtained by applying SVD to the cross variance matrix Σ_{PF} . Calculate the SVD

$$\Sigma_{PP}^{-1/2} \Sigma_{PF} \Sigma_{FF}^{-1/2} = \mathbf{U} \mathbf{W} \mathbf{V}^T. \quad (3.32)$$

The transformation matrix \mathbf{T} , which maps from the past the information critical to the prediction of the future is calculated as

$$\mathbf{T} = \mathbf{U}^T (\mathbf{P} \mathbf{P}^T)^{-1/2}. \quad (3.33)$$

The state sequence is then calculated as

$$\mathbf{X}_k = \mathbf{T} \mathbf{P}. \quad (3.34)$$

The next step is to obtain an estimate of \mathbf{X}_{k+1} . This can be done by shifting the matrix \mathbf{T} forward a step in time and recalculating (3.34). Finally least squares regression is used to obtain the state matrices by solving

$$\begin{pmatrix} \mathbf{X}_{k+1} \\ \mathbf{Y}_{k,1,N} \end{pmatrix} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix} \begin{pmatrix} \mathbf{X}_k \\ \mathbf{U}_{k,1,N} \end{pmatrix}. \quad (3.35)$$

3.4.2 The MOESP Algorithm

The MOESP approach [21, 22, 24] centres on a QR decomposition which creates 4 orthogonal subspaces.

Assuming that the process noise and the measurement noise are zero mean, and normally distributed sequences independent of each other, and independent of the input sequence; and that the input \mathbf{U} is sufficiently exciting, and assuming a stable system, then the orthogonality of the rows of \mathbf{Q} can be exploited in the following manner.

First, block Hankel matrices are built, corresponding to Eqs. 3.40 and 3.41. Note the different order to N4SID, where the Hankel Matrices are arranged with \mathbf{U}_f on top. A QR decomposition is then applied:

$$\begin{pmatrix} \mathbf{U}_f \\ \mathbf{U}_p \\ \mathbf{Y}_p \\ \mathbf{Y}_f \end{pmatrix} = \begin{pmatrix} \mathbf{R}_{11} & 0 & 0 & 0 \\ \mathbf{R}_{21} & \mathbf{R}_{22} & 0 & 0 \\ \mathbf{R}_{31} & \mathbf{R}_{32} & \mathbf{R}_{33} & 0 \\ \mathbf{R}_{41} & \mathbf{R}_{42} & \mathbf{R}_{43} & \mathbf{R}_{44} \end{pmatrix} \begin{pmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \\ \mathbf{Q}_3 \\ \mathbf{Q}_4 \end{pmatrix} \quad (3.36)$$

Verhaegen [23, 24] proves that the column space of the extended observability matrix is shared by both \mathbf{R}_{42} and \mathbf{R}_{43} . Singular value decomposition is applied to obtain

$$\begin{bmatrix} \mathbf{R}_{42} & \mathbf{R}_{43} \end{bmatrix} = \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{bmatrix} \begin{bmatrix} \mathbf{S}_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{bmatrix} \quad (3.37)$$

Where the column space of \mathbf{U}_1 approximates Γ_r . This is followed by a series of algebraic manipulations leading eventually to the extraction of the state matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$. For a detailed account of the MOESP subspace system identification algorithm see also [21, 22, 30, 57].

3.4.3 N4SID Approach

Numerical methods for subspace system identification (N4SID) methods [1, 25, 47, 58, 59] use techniques from linear algebra to solve the identification problem. The algorithm begins by building Hankel matrices from the external measurements of the system. These are then divided into “past” and “future” halves. References to past and

future in this context are relative to each other - because 4SID algorithms are described in terms of a single batch of data, where the first half of the data sequences are referred to as “past”, and the second half “future”.

In contrast to the statistical approach of CVA, which uses the covariance matrices to compute the states, 4SID methods calculate a least squares prediction by projecting the rows of the future outputs matrix onto the rows of the past inputs, past outputs and future inputs matrices. The influence of the future inputs is then removed to calculate a so-called k -step predictor matrix [31]. An SVD is then used to approximate the extended observability matrix and the state sequence. Finally, once the state matrices (\mathbf{A}, \mathbf{C}) and the states have been estimated, (\mathbf{B}, \mathbf{D}) are estimated by solving an over-determined system of equations.

N4SID identifies the state matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ of linear models with the form:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{v}_k, \\ \mathbf{y}_k &= \mathbf{C}\mathbf{x}_k + \mathbf{D}\mathbf{u}_k + \mathbf{w}_k, \end{aligned} \quad (3.38)$$

where the outputs \mathbf{y}_k , inputs \mathbf{u}_k , and state variables \mathbf{x}_k are of dimension l , m , and n respectively, and \mathbf{v}_k and \mathbf{w}_k are uncorrelated zero mean white noise processes of dimensions n and l . N4SID [25] has been proven to deliver consistent, unbiased estimates subject to the following conditions: an observable and controllable LTI system, where the process and measurement noise is zero mean and Gaussian, and is uncorrelated with the deterministic inputs to the system. The covariance matrix of the process, and the measurement noise, is calculated from the model residuals as

$$E_j((\mathbf{v}_k \mathbf{w}_j) (\mathbf{v}_k \mathbf{w}_j)^T) = \begin{pmatrix} \mathbf{Q} & \mathbf{S}^T \\ \mathbf{S} & \mathbf{R} \end{pmatrix} \delta_{kj}. \quad (3.39)$$

First define the block Hankel matrices $\mathbf{U}_p, \mathbf{U}_f, \mathbf{Y}_p, \mathbf{Y}_f \dots$

$$\begin{aligned}
\begin{pmatrix} \mathbf{U}_p \\ \dots \\ \mathbf{U}_f \end{pmatrix} &= \begin{pmatrix} \mathbf{U}_{1,r,N} \\ \dots \\ \mathbf{U}_{r+1,r,N} \end{pmatrix} = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_N \\ \mathbf{u}_2 & \mathbf{u}_3 & \dots & \mathbf{u}_{N+1} \\ \vdots & \vdots & \dots & \vdots \\ \mathbf{u}_r & \mathbf{u}_{r+1} & \dots & \mathbf{u}_{r+N-1} \\ \dots & \dots & \dots & \dots \\ \mathbf{u}_{r+1} & \mathbf{u}_{r+2} & \dots & \mathbf{u}_{r+N} \\ \vdots & \vdots & \dots & \vdots \\ \vdots & \vdots & \dots & \vdots \\ \mathbf{u}_{2r} & \mathbf{u}_{2r+1} & \dots & \mathbf{u}_{2r+N-1} \end{bmatrix} \\
&= \begin{pmatrix} \mathbf{U}_{1,r+1,N} \\ \dots \\ \mathbf{U}_{r+2,r-1,N} \end{pmatrix} = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_N \\ \mathbf{u}_2 & \mathbf{u}_3 & \dots & \mathbf{u}_{N+1} \\ \vdots & \vdots & \dots & \vdots \\ \mathbf{u}_{r+1} & \mathbf{u}_{r+2} & \dots & \mathbf{u}_{r+N} \\ \dots & \dots & \dots & \dots \\ \mathbf{u}_{r+2} & \mathbf{u}_{r+3} & \dots & \mathbf{u}_{r+N+1} \\ \vdots & \vdots & \dots & \vdots \\ \vdots & \vdots & \dots & \vdots \\ \mathbf{u}_{2r} & \mathbf{u}_{2r+1} & \dots & \mathbf{u}_{2r+N-1} \end{bmatrix} \in \mathbb{R}^{2rm \times N}.
\end{aligned} \tag{3.40}$$

$$\begin{aligned}
\begin{pmatrix} \mathbf{Y}_p \\ \dots \\ \mathbf{Y}_f \end{pmatrix} &= \begin{pmatrix} \mathbf{Y}_{1,r,N} \\ \dots \\ \mathbf{Y}_{r+1,r,N} \end{pmatrix} = \begin{bmatrix} \mathbf{y}_1 & \mathbf{y}_2 & \dots & \mathbf{y}_N \\ \mathbf{y}_2 & \mathbf{y}_3 & \dots & \mathbf{y}_{N+1} \\ \vdots & \vdots & \dots & \vdots \\ \mathbf{y}_r & \mathbf{y}_{r+1} & \dots & \mathbf{y}_{r+N-1} \\ \dots & \dots & \dots & \dots \\ \mathbf{y}_{r+1} & \mathbf{y}_{r+2} & \dots & \mathbf{y}_{r+N} \\ \vdots & \vdots & \dots & \vdots \\ \vdots & \vdots & \dots & \vdots \\ \mathbf{y}_{2r} & \mathbf{y}_{2r+1} & \dots & \mathbf{y}_{2r+N-1} \end{bmatrix} \\
&= \begin{pmatrix} \mathbf{Y}_{1,r+1,N} \\ \dots \\ \mathbf{Y}_{r+2,r-1,N} \end{pmatrix} = \begin{bmatrix} \mathbf{y}_1 & \mathbf{y}_2 & \dots & \mathbf{y}_N \\ \mathbf{y}_2 & \mathbf{y}_3 & \dots & \mathbf{y}_{N+1} \\ \vdots & \vdots & \dots & \vdots \\ \mathbf{y}_{r+1} & \mathbf{y}_{r+2} & \dots & \mathbf{y}_{r+N} \\ \dots & \dots & \dots & \dots \\ \mathbf{y}_{r+2} & \mathbf{y}_{r+3} & \dots & \mathbf{y}_{r+N+1} \\ \vdots & \vdots & \dots & \vdots \\ \vdots & \vdots & \dots & \vdots \\ \mathbf{y}_{2r} & \mathbf{y}_{2r+1} & \dots & \mathbf{y}_{2r+N-1} \end{bmatrix} \in \mathbb{R}^{2rl \times N}.
\end{aligned} \tag{3.41}$$

It is relatively straight forward [1, 60] to verify that the recursive equations Eq. 3.1 can be written explicitly as

$$\mathbf{Y}_k = \mathbf{\Gamma}_r \mathbf{X}_k + \mathbf{\Phi}_r \mathbf{U}_k + \mathbf{Y}_k^s, \quad (3.42)$$

where \mathbf{Y}_k^s is the stochastic part of the system that represents the effect of the process and measurement noise on the external measurements, and

$$\mathbf{\Gamma}_r = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{r-1} \end{bmatrix} \in \mathbf{R}^{lr \times n}, \quad (3.43)$$

$$\mathbf{\Phi}_r = \begin{bmatrix} \mathbf{D} & 0 & 0 & 0 & 0 \\ \mathbf{CB} & \mathbf{D} & 0 & 0 & 0 \\ \mathbf{CAB} & \mathbf{CB} & \mathbf{D} & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ \mathbf{CA}^{r-2}\mathbf{B} & \dots & \dots & \dots & \mathbf{D} \end{bmatrix} \in \mathbf{R}^{lr \times mr}, \quad (3.44)$$

$$\mathbf{Y}_k = \begin{bmatrix} \mathbf{y}_k & \mathbf{y}_{k+1} & \dots & \mathbf{y}_{k+N-1} \\ \mathbf{y}_{k+1} & \mathbf{y}_{k+2} & \dots & \mathbf{y}_{k+N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{y}_{k+r-1} & \mathbf{y}_{k+r} & \dots & \mathbf{y}_{k+N+r-2} \end{bmatrix} \in \mathbf{R}^{lr \times N}, \quad (3.45)$$

$$\mathbf{X}_k \stackrel{\text{def}}{=} [\mathbf{x}_k \quad \mathbf{x}_{k+1} \quad \dots \quad \mathbf{x}_{k+N-1}] \in \mathbf{R}^{n \times N}, \quad (3.46)$$

where $\mathbf{U}_k = \mathbf{U}_{k,r,N} \in \mathbf{R}^{mr \times N}$ and $r > n$ is the number of block rows in $\mathbf{\Gamma}_r$, $\mathbf{\Phi}_r$, \mathbf{Y}_k and \mathbf{U}_k . Eq. 3.42 is the basis on which the 4SID algorithms are built. The aim is to estimate the quantity $\mathbf{\Gamma}_r \mathbf{X}_k$, then use SVD in Eq. 3.42 to calculate the principal directions of the process. Provided that the number of block rows, r , is chosen larger than the order of the system, n , then the SVD reveals n principal components (or states) that describe the principal dynamics of the system. Applying SVD to the expression $\mathbf{\Gamma}_r \mathbf{X}_k$ yields (in the case of a deterministic system)

$$\mathbf{\Gamma}_r \mathbf{X}_k = [\mathbf{U}_1 \quad \mathbf{U}_2] \begin{bmatrix} \mathbf{S}_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{bmatrix}, \quad (3.47)$$

where

$$\Gamma_n = \mathbf{U}_1 \mathbf{S}_1^{1/2}. \quad (3.48)$$

A valid state sequence is $\mathbf{X}_k = \mathbf{S}_1^{1/2} \mathbf{V}_1^T$. The matrices \mathbf{A}, \mathbf{C} can be extracted from Γ_n by exploiting the shift invariant structure of Γ_r , i.e. form the matrices $\Gamma_{2:r}$ and $\Gamma_{1:r-1}$, then solve

$$\mathbf{A} = \Gamma_{1:r-1}^\dagger \Gamma_{2:r}. \quad (3.49)$$

The calculation of $\Gamma_r \mathbf{X}_k$ is the key to unlocking the column space of the extended observability matrix and the row space of the states. N4SID calculates this expression by estimating

$$\Gamma_r \mathbf{X}_k = \mathbf{Y}_k - \Phi_r \mathbf{U}_k - \mathbf{Y}_k^s. \quad (3.50)$$

The initial projection used in the 4SID methods corresponds to an ordinary least squares solution for an ARX model structure, where the number of lags used in the model is the number of block rows, r . The k^{th} row in $\tilde{\mathbf{Y}}_f$ in Eq. 3.52 below, is composed of k -step ahead ARX predictions of the system, where $\tilde{\mathbf{Y}}_f = f(\mathbf{P}, \mathbf{U}_f)$. More details appear in Ljung [31]. In the presence of correlated data, the least squares procedure encounters difficulty in calculating the inverse of the covariance matrix in Eq. 3.52. One way to guarantee that the required inverse exists is to calculate the SVD of the covariance matrix, then discard the singular values below a pre-set threshold. In contrast, Overschee [1] calculates an RQ decomposition, and then uses the R factors to obtain the required solution. To help simplify the notation first define:

$$\mathbf{P} = \begin{bmatrix} \mathbf{U}_p \\ \mathbf{Y}_p \end{bmatrix}, \quad (3.51)$$

then,

$$\tilde{\mathbf{Y}}_f = \mathbf{Y}_f \left[\mathbf{P}^T \mathbf{U}_f^T \right] \begin{bmatrix} \mathbf{P} \mathbf{P}^T & \mathbf{P} \mathbf{U}_f^T \\ \mathbf{U}_f \mathbf{P}^T & \mathbf{U}_f \mathbf{U}_f^T \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{P} \\ \mathbf{U}_f \end{bmatrix}. \quad (3.52)$$

Note that Eq. 3.52 is comprised of linear combinations of the past inputs, past outputs and future inputs Hankel matrices. To see this more clearly, 3.52 can be written as

$$\tilde{\mathbf{Y}}_f = [\mathbf{L}_1 \quad \mathbf{L}_2 \quad \mathbf{L}_3] \begin{bmatrix} \mathbf{U}_p \\ \mathbf{Y}_p \\ \mathbf{U}_f \end{bmatrix}. \quad (3.53)$$

Overschee proves in [25] that the estimate corresponding to Eq. 3.53 can be equated to Eq. 3.42, where the estimated state sequence $\hat{\mathbf{X}}_k$ is equivalent to the state estimates of a non-steady state Kalman Filter, i.e. Eq. 3.53 can be used to estimate Eq. 3.42:

$$\tilde{\mathbf{Y}}_f = \mathbf{L}_1 \mathbf{U}_p + \mathbf{L}_2 \mathbf{Y}_p + \mathbf{L}_3 \mathbf{U}_f = \mathbf{\Gamma}_r \hat{\mathbf{X}}_f + \mathbf{\Phi}_r \mathbf{U}_f. \quad (3.54)$$

Note that each side of Eq. 3.54 contains linear combinations of the future inputs Hankel matrix. By subtracting the linear of combinations of \mathbf{U}_f from Eq. 3.54, an estimate of $\mathbf{\Gamma}_r \hat{\mathbf{X}}_k$ is obtained, i.e. calculate $\tilde{\mathbf{Y}}_f - \mathbf{L}_3 \mathbf{U}_f = \mathbf{L}_1 \mathbf{U}_p + \mathbf{L}_2 \mathbf{Y}_p$, which is done as follows:

$$\hat{\mathbf{Y}}_f = \mathbf{L}_1 \mathbf{U}_p + \mathbf{L}_2 \mathbf{Y}_p = \mathbf{Y}_f (\mathbf{P}^T \mathbf{U}_f^T) \begin{pmatrix} \mathbf{P} \mathbf{P}^T & \mathbf{P} \mathbf{U}_f^T \\ \mathbf{U}_f \mathbf{P}^T & \mathbf{U}_f \mathbf{U}_f^T \end{pmatrix}_{\text{first } (lr+nr) \text{ rows}}^{-1} \begin{pmatrix} \mathbf{U}_p \\ \mathbf{Y}_p \end{pmatrix}. \quad (3.55)$$

Eq. 3.55 defines the subspace projection used in 4SID methods, from which is extracted the state space model. It provides an estimate of the non steady state Kalman states (as proved in [25]), i.e.

$$\mathbf{\Gamma} \hat{\mathbf{X}}_f = \mathbf{L}_1 \mathbf{U}_p + \mathbf{L}_2 \mathbf{Y}_p. \quad (3.56)$$

Note the similarity between Eq. 3.56 and Eq. 3.34 of the CVA algorithm. Both expressions use linear combinations of the past Hankel Matrices to estimate the states. In [32], Overschee describes the exact connection between the two methods, where N4SID and CVA are shown to be weighted versions of exactly the same subspace. A further weighting is applied to Eq. 3.56 in the “robustified” N4SID algorithm [8]. Overschee claims that improved accuracy can be obtained by further removing the influence of the “future outputs” from Eq. 3.56, by calculating the orthogonal complement to the row space of the future inputs $\mathbf{\Pi}_{U_f}^\perp$, which is used as an instrument for removing any remaining effect of \mathbf{U}_f . A numerically efficient way to do this appears in [1]:

$$\Pi_{U_f}^\perp = \mathbf{I}_{N \times N} - \mathbf{U}_f^T (\mathbf{U}_f \mathbf{U}_f^T)^{-1} \mathbf{U}_f. \quad (3.57)$$

The final projection from which the state sequences are estimated is

$$\mathbf{Q} = (\mathbf{L}_1 \mathbf{U}_p + \mathbf{L}_2 \mathbf{Y}_p) \Pi_{U_k}^\perp, \quad (3.58)$$

or

$$\mathbf{Q} = \Gamma \hat{\mathbf{X}}. \quad (3.59)$$

SVD is applied to Eq. 3.59. This provides a stable and robust numerical routine for extracting a state basis, leading to a balanced realisation, where the magnitude of the singular values provides a measure of the distance of the model to lower order models [28, 61]. In addition, the controllability and observability Grammians are equal, which guarantees that the input-state path in the model is balanced with the state-output path [15]. Therefore, the principal directions that correspond to the largest singular values capture the most important dynamics in the system. Eq. 3.59 is factorised as

$$\Gamma_r \hat{\mathbf{X}}_f = [\mathbf{U}_1 \quad \mathbf{U}_2] \begin{bmatrix} \mathbf{S}_1 & 0 \\ 0 & \mathbf{S}_2 \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{bmatrix}, \quad (3.60)$$

where

$$\Gamma_r = \mathbf{U}_1 \mathbf{S}_1^{1/2}. \quad (3.61)$$

The state matrices (\mathbf{A}, \mathbf{C}) can be calculated by exploiting the shift invariant structure of the extended observability matrix $\Gamma_r \in \mathbb{R}^{n \times n}$. The procedure is outlined in Eqs. 3.6 and 3.7, other methods are suggested in [49, 50].

N4SID proceeds by calculating a new state sequence $\hat{\mathbf{X}}_{f+1}$, by shifting the division between “past” and “future” in Eqs 3.40 and 3.41 down a row, and recalculating Eq. 3.55. First define:

$$\mathbf{P}_* = \begin{bmatrix} \mathbf{U}_{1,r+1,N} \\ \mathbf{Y}_{1,r+1,N} \end{bmatrix}, \quad \begin{pmatrix} \mathbf{U}_{f*} \\ \mathbf{Y}_{f*} \end{pmatrix} = \begin{pmatrix} \mathbf{U}_{r+2,r-1,N} \\ \mathbf{Y}_{r+2,r-1,N} \end{pmatrix}. \quad (3.62)$$

Then a matrix of k -step estimates based on projecting $\mathbf{Y}_{r+2,r-1,N}$ is calculated:

$$\hat{\mathbf{Y}}_{r+2,r-1,N} = \mathbf{Y}_{f*} [\mathbf{P}_*^T \mathbf{U}_{f*}^T \begin{bmatrix} \mathbf{P}_* \mathbf{P}_*^T & \mathbf{P}_* \mathbf{U}_{f*}^T \\ \mathbf{U}_{f*} \mathbf{P}_*^T & \mathbf{U}_{f*} \mathbf{U}_{f*}^T \end{bmatrix}^{-1}]_{\text{first } (l+m)(r+1) \text{ rows}} \begin{bmatrix} \mathbf{U}_{p*} \\ \mathbf{Y}_{p*} \end{bmatrix}. \quad (3.63)$$

The shift invariance structure of the estimate of the extended observability matrix Γ_r from Eq. 3.61 is exploited, to find an estimate for $\Gamma_{r-1} \in \mathbb{R}^{(r-1)l \times n}$. This follows directly by subtracting the last l rows from Γ_r . The final extraction of the state matrices takes the following relationships within the model into account, where from Eq. 3.54 it follows directly:

$$\hat{\mathbf{X}}_f = \Gamma_r^\dagger (\hat{\mathbf{Y}}_f - \Phi_r \mathbf{U}_f), \quad (3.64)$$

and in a similar fashion for $\hat{\mathbf{X}}_{f+1}$, using Eq. 3.63

$$\hat{\mathbf{X}}_{f+1} = \Gamma_{r-1}^\dagger (\hat{\mathbf{Y}}_{r+2,r-1,N} - \Phi_{r-1} \mathbf{U}_{r+2,r-1,N}) \quad (3.65)$$

where $\hat{\mathbf{X}}_f = [\hat{\mathbf{x}}_f \quad \hat{\mathbf{x}}_{f+1} \quad \cdots \quad \hat{\mathbf{x}}_{f+N-1}]$, and $\hat{\mathbf{X}}_{f+1} = [\hat{\mathbf{x}}_{f+1} \quad \hat{\mathbf{x}}_{f+2} \quad \cdots \quad \hat{\mathbf{x}}_{f+N}]$.

The following relationship is constructed from Eq. 3.38, where the white noise sequence terms have been dropped, as they correspond to the residuals of a least squares fit of the model parameters.

$$\begin{aligned} \hat{\mathbf{X}}_{f+1} &= \mathbf{A} \hat{\mathbf{X}}_f + \mathbf{B} \mathbf{U}_{r+1,1,N} \\ \mathbf{Y}_{r+1,1,N} &= \mathbf{C} \hat{\mathbf{X}}_f + \mathbf{D} \mathbf{U}_{r+1,1,N} \end{aligned} \quad (3.66)$$

The expressions for $\hat{\mathbf{X}}_f$ and $\hat{\mathbf{X}}_{f+1}$ in Eq. 3.64 are substituted into Eq. 3.66 and then simple algebraic manipulations lead to:

$$\Gamma_{r-1}^\dagger (\hat{\mathbf{Y}}_{r+2,r-1,N} - \Phi_{r-1} \mathbf{U}_{r+2,r-1,N}) = \mathbf{A} \Gamma_r^\dagger (\hat{\mathbf{Y}}_f - \Phi_r \mathbf{U}_f) + \mathbf{B} \mathbf{U}_{r+1,1,N}, \quad (3.67)$$

$$\Gamma_{r-1}^\dagger \hat{\mathbf{Y}}_{r+2,r-1,N} = \mathbf{A} \Gamma_r^\dagger \hat{\mathbf{Y}}_f - \mathbf{A} \Gamma_r^\dagger \Phi_r \mathbf{U}_f + \mathbf{B} \mathbf{U}_{r+1,1,N} + \Gamma_{r-1}^\dagger \Phi_{r-1} \mathbf{U}_{r+2,r-1,N}, \quad (3.68)$$

$$\Gamma_{r-1}^\dagger \hat{\mathbf{Y}}_{r+2,r-1,N} = \mathbf{A} \Gamma_r^\dagger \hat{\mathbf{Y}}_f + [\mathbf{B} \mid \Gamma_{r-1}^\dagger \Phi_{r-1} - \mathbf{A} \Gamma_r^\dagger \Phi_r] \begin{bmatrix} \mathbf{U}_{r+1,1,N} \\ \mathbf{U}_{r+2,r-1,N} \end{bmatrix}. \quad (3.69)$$

The expression for $\hat{\mathbf{X}}_f$ in Eq. 3.63 is substituted into Eq. 3.66 to obtain

$$\mathbf{Y}_{r+1,1,N} = \mathbf{C}\Gamma_r^\dagger (\hat{\mathbf{Y}}_f - \Phi_r \mathbf{U}_f) + \mathbf{D}\mathbf{U}_{r+1,1,N}. \quad (3.70)$$

$$\mathbf{Y}_{r+1,1,N} = \mathbf{C}\Gamma_r^\dagger \hat{\mathbf{Y}}_f + [\mathbf{D} \mid 0 - \mathbf{C}\Gamma_r^\dagger \Phi_r] \mathbf{U}_f. \quad (3.71)$$

Taking note that $\begin{bmatrix} \mathbf{U}_{r+1,1,N} \\ \mathbf{U}_{r+2,r-1,N} \end{bmatrix} = \mathbf{U}_f$; and combining Eq. 3.69 and Eq. 3.71 leads to

$$\begin{pmatrix} \Gamma_{r-1}^\dagger \hat{\mathbf{Y}}_{r+2,r-1,N} \\ \mathbf{Y}_{r+1,1,N} \end{pmatrix} = \begin{pmatrix} \mathbf{A} \\ \mathbf{C} \end{pmatrix} \Gamma_r^\dagger \hat{\mathbf{Y}}_f + \begin{pmatrix} \mathbf{B} \mid \Gamma_{r-1}^\dagger \Phi_{r-1} - \mathbf{A}\Gamma_r^\dagger \Phi_r \\ \mathbf{D} \mid 0 - \mathbf{C}\Gamma_r^\dagger \Phi_r \end{pmatrix} \mathbf{U}_f. \quad (3.72)$$

$$\begin{pmatrix} \Gamma_{r-1}^\dagger \hat{\mathbf{Y}}_{r+2,r-1,N} \\ \mathbf{Y}_{r+1,1,N} \end{pmatrix} = \begin{pmatrix} \mathbf{A} \\ \mathbf{C} \end{pmatrix} \Gamma_r^\dagger \hat{\mathbf{Y}}_f + \Delta \mathbf{U}_f. \quad (3.73)$$

Eq. 3.73 is now solved using least squares to estimate \mathbf{A}, \mathbf{C} and Δ . The term Δ is linear in (\mathbf{B}, \mathbf{D}) , to see this write

$$\Delta = \begin{bmatrix} \Delta_1 \\ \Delta_2 \end{bmatrix} \stackrel{\text{def}}{=} \begin{pmatrix} \mathbf{B} \mid \Gamma_{r-1}^\dagger \Phi_{r-1} - \mathbf{A}\Gamma_r^\dagger \Phi_r \\ \mathbf{D} \mid 0 - \mathbf{C}\Gamma_r^\dagger \Phi_r \end{pmatrix}, \quad (3.74)$$

with

$$\Delta_1 = \mathbf{B} \mid \Gamma_{r-1}^\dagger \begin{bmatrix} \mathbf{D} & 0 & \dots & 0 \\ \mathbf{CB} & \mathbf{D} & \dots & 0 \\ \vdots & \dots & \ddots & \vdots \\ \mathbf{CA}^{r-3}\mathbf{B} & \mathbf{CA}^{r-2}\mathbf{B} & \dots & \mathbf{D} \end{bmatrix} - \mathbf{A}\Gamma_r^\dagger \begin{bmatrix} \mathbf{D} & 0 & \dots & 0 \\ \mathbf{CB} & \mathbf{D} & \dots & 0 \\ \vdots & \dots & \ddots & \vdots \\ \mathbf{CA}^{r-2}\mathbf{B} & \mathbf{CA}^{r-1}\mathbf{B} & \dots & \mathbf{D} \end{bmatrix}. \quad (3.75)$$

$$\Delta_2 = \mathbf{D} \mid [0] \begin{bmatrix} \mathbf{D} & 0 & \dots & 0 \\ \mathbf{CB} & \mathbf{D} & \dots & 0 \\ \vdots & \dots & \ddots & \vdots \\ \mathbf{CA}^{r-3}\mathbf{B} & \mathbf{CA}^{r-2}\mathbf{B} & \dots & \mathbf{D} \end{bmatrix} - \mathbf{C}\Gamma_r^\dagger \begin{bmatrix} \mathbf{D} & 0 & \dots & 0 \\ \mathbf{CB} & \mathbf{D} & \dots & 0 \\ \vdots & \dots & \ddots & \vdots \\ \mathbf{CA}^{r-2}\mathbf{B} & \mathbf{CA}^{r-1}\mathbf{B} & \dots & \mathbf{D} \end{bmatrix}. \quad (3.76)$$

$$\mathbf{G}_{1j} \in \mathbb{R}^{n \times l}, \quad \mathbf{G}_{2j} \in \mathbb{R}^{l \times l}, \quad \mathbf{H}_j \in \mathbb{R}^{n \times l}, \quad \Delta_{1j} \in \mathbb{R}^{n \times m}, \quad \Delta_{2j} \in \mathbb{R}^{l \times m},$$

$$\begin{aligned}
\begin{bmatrix} \mathbf{G}_{11} & \mathbf{G}_{12} & \cdots & \mathbf{G}_{1r} \\ \mathbf{G}_{21} & \mathbf{G}_{22} & \cdots & \mathbf{G}_{2r} \end{bmatrix} &= \begin{pmatrix} \mathbf{A} \\ \mathbf{C} \end{pmatrix} \mathbf{\Gamma}_r^\dagger \quad \in \mathbb{R}^{(n+l) \times lr} \\
[\mathbf{H}_1 & \mathbf{H}_2 & \cdots & \mathbf{H}_{r-1}] &= \mathbf{\Gamma}_{r-1}^\dagger \quad \in \mathbb{R}^{n \times l(r-1)} \\
\begin{bmatrix} \Delta_{11} & \Delta_{12} & \cdots & \Delta_{1r} \\ \Delta_{21} & \Delta_{22} & \cdots & \Delta_{2r} \end{bmatrix} &= \begin{pmatrix} \mathbf{B} | \mathbf{\Gamma}_{r-1}^\dagger \mathbf{\Phi}_{r-1} - \mathbf{A} \mathbf{\Gamma}_r^\dagger \mathbf{\Phi}_r \\ \mathbf{D} | 0 - \mathbf{C} \mathbf{\Gamma}_r^\dagger \mathbf{\Phi}_r \end{pmatrix} \quad \in \mathbb{R}^{(n+l) \times mr}
\end{aligned} \tag{3.77}$$

In the next step, the vector operation is performed on Δ , i.e. the columns of Δ are stacked on top of each other to define the following equalities:

$$\begin{aligned}
\begin{pmatrix} \Delta_{11} \\ \Delta_{12} \\ \Delta_{13} \\ \vdots \\ \Delta_{1r} \\ \hline \Delta_{21} \\ \Delta_{22} \\ \Delta_{23} \\ \vdots \\ \Delta_{2r} \end{pmatrix} &= \begin{pmatrix} \mathbf{B} | & -\mathbf{G}_{11} & \mathbf{H}_1 - \mathbf{G}_{12} & \cdots & \mathbf{H}_{r-2} - \mathbf{G}_{1,r-1} & \mathbf{H}_{r-1} - \mathbf{G}_{1r} \\ & \mathbf{H}_1 - \mathbf{G}_{12} & \mathbf{H}_2 - \mathbf{G}_{13} & \cdots & \mathbf{H}_{r-1} - \mathbf{G}_{1r} & 0 \\ & \mathbf{H}_2 - \mathbf{G}_{13} & \mathbf{H}_3 - \mathbf{G}_{14} & \cdots & 0 & 0 \\ & \vdots & \vdots & \vdots & \vdots & \vdots \\ & \mathbf{H}_{r-1} - \mathbf{G}_{1r} & 0 & \cdots & 0 & 0 \\ \hline & \mathbf{D} | & -\mathbf{G}_{21} & -\mathbf{G}_{22} & \cdots & -\mathbf{G}_{2,r-1} & -\mathbf{G}_{2r} \\ & & -\mathbf{G}_{22} & -\mathbf{G}_{23} & \cdots & -\mathbf{G}_{2r} & 0 \\ & & -\mathbf{G}_{23} & -\mathbf{G}_{24} & \cdots & 0 & 0 \\ & & \vdots & \vdots & \vdots & \vdots & \vdots \\ & & -\mathbf{G}_{2r} & 0 & \cdots & 0 & 0 \end{pmatrix} \begin{bmatrix} \mathbf{D} \\ \mathbf{CB} \\ \mathbf{CAB} \\ \vdots \\ \mathbf{CA}^{r-2}\mathbf{B} \end{bmatrix} \quad \in \mathbb{R}^{r(n+l) \times m} \\
\end{aligned} \tag{3.78}$$

$$\begin{pmatrix} \mathbf{E}_{11} \\ \vdots \\ \vdots \\ \vdots \\ \mathbf{E}_{1r} \\ \hline \mathbf{E}_{21} \\ \vdots \\ \vdots \\ \vdots \\ \mathbf{E}_{2r} \end{pmatrix} = \begin{pmatrix} \mathbf{B} | -\mathbf{G}_{11} & \mathbf{H}_1 - \mathbf{G}_{12} & \cdots & \mathbf{H}_{r-2} - \mathbf{G}_{1,r-1} & \mathbf{H}_{r-1} - \mathbf{G}_{1r} \\ \mathbf{H}_1 - \mathbf{G}_{12} & \mathbf{H}_2 - \mathbf{G}_{13} & \cdots & \mathbf{H}_{r-1} - \mathbf{G}_{1r} & 0 \\ \mathbf{H}_2 - \mathbf{G}_{13} & \mathbf{H}_3 - \mathbf{G}_{14} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{H}_{r-1} - \mathbf{G}_{1r} & 0 & \cdots & 0 & 0 \\ \hline \mathbf{D} | -\mathbf{G}_{21} & -\mathbf{G}_{22} & \cdots & -\mathbf{G}_{2,r-1} & -\mathbf{G}_{2r} \\ -\mathbf{G}_{22} & -\mathbf{G}_{23} & \cdots & -\mathbf{G}_{2r} & 0 \\ -\mathbf{G}_{23} & -\mathbf{G}_{24} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -\mathbf{G}_{2r} & 0 & \cdots & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{I}_l & 0 \\ 0 & \mathbf{\Gamma}_{r-1} \end{pmatrix} \in \mathbb{R}^{r(n+l) \times (n+l)} \quad (3.79)$$

Eqs. 3.78 and 3.79 are linked through the relationship

$$\mathbf{\Delta} = \mathbf{E} \begin{pmatrix} \mathbf{D} \\ \mathbf{B} \end{pmatrix} \in \mathbb{R}. \quad (3.80)$$

Eq. 3.80 expresses $\mathbf{\Delta}$ as a linear function of (\mathbf{B}, \mathbf{D}) . $\mathbf{\Delta} = f(\mathbf{B}, \mathbf{D})$. To estimate (\mathbf{B}, \mathbf{D}) from Eq. 3.80, find (\mathbf{B}, \mathbf{D}) so as to minimise

$$\left\| \begin{pmatrix} \mathbf{\Gamma}_{r-1}^\dagger \hat{\mathbf{Y}}_{r+2,r-1,N} \\ \mathbf{Y}_{r+1,1,N} \end{pmatrix} - \begin{pmatrix} \mathbf{A} \\ \mathbf{C} \end{pmatrix} \mathbf{\Gamma}_r^\dagger \hat{\mathbf{Y}}_f - \begin{pmatrix} \mathbf{B} | \mathbf{\Gamma}_{r-1}^\dagger \mathbf{\Phi}_{r-1} - \mathbf{A} \mathbf{\Gamma}_r^\dagger \mathbf{\Phi}_r \\ \mathbf{D} | 0 - \mathbf{C} \mathbf{\Gamma}_r^\dagger \mathbf{\Phi}_r \end{pmatrix} \mathbf{U}_f \right\|_F^2. \quad (3.81)$$

First, define the following term (which is known):

$$\mathbf{\Omega} = \begin{pmatrix} \mathbf{\Gamma}_{r-1}^\dagger \hat{\mathbf{Y}}_{r+2,r-1,N} \\ \mathbf{Y}_{r+1,1,N} \end{pmatrix} - \begin{pmatrix} \mathbf{A} \\ \mathbf{C} \end{pmatrix} \mathbf{\Gamma}_r^\dagger \hat{\mathbf{Y}}_f. \quad (3.82)$$

Then solve the following equation in a least squares sense:

$$\mathbf{\Omega} = \begin{pmatrix} \mathbf{B} | \mathbf{\Gamma}_{r-1}^\dagger \mathbf{\Phi}_{r-1} - \mathbf{A} \mathbf{\Gamma}_r^\dagger \mathbf{\Phi}_r \\ \mathbf{D} | 0 - \mathbf{C} \mathbf{\Gamma}_r^\dagger \mathbf{\Phi}_r \end{pmatrix} \mathbf{U}_f, \quad (3.83)$$

i.e. minimise:

$$\left\| \mathbf{\Omega} - \begin{pmatrix} \mathbf{B} | \mathbf{\Gamma}_{r-1}^\dagger \mathbf{\Phi}_{r-1} - \mathbf{A} \mathbf{\Gamma}_r^\dagger \mathbf{\Phi}_r \\ \mathbf{D} | 0 - \mathbf{C} \mathbf{\Gamma}_r^\dagger \mathbf{\Phi}_r \end{pmatrix} \mathbf{U}_f \right\|_F^2. \quad (3.84)$$

Eq. 3.83 is a least squares problem that is linear in \mathbf{B} and \mathbf{D} . One way to determine Δ , is to multiply Eq. 3.83 by \mathbf{U}_f^T , then solve Eq. 3.80 using least squares, to get the values for \mathbf{B} and \mathbf{D} . However, in the case of badly conditioned input Hankel matrices, this has been shown to produce poor results [1]. To avoid having to compute the inverse of \mathbf{U}_f , \mathbf{B} and \mathbf{D} can be computed directly using Eq. 3.79 by writing

$$\begin{pmatrix} \mathbf{B} | \mathbf{\Gamma}_{r-1}^T \mathbf{\Phi}_{r-1} - \mathbf{A} \mathbf{\Gamma}_r^T \mathbf{\Phi}_r \\ \mathbf{D} | 0 - \mathbf{C} \mathbf{\Gamma}_r^T \mathbf{\Phi}_r \end{pmatrix} \mathbf{U}_f = \sum_{k=1}^r \begin{pmatrix} \mathbf{E}_{1k} \\ \mathbf{E}_{2k} \end{pmatrix} \begin{pmatrix} \mathbf{D} \\ \mathbf{B} \end{pmatrix} \mathbf{U}_{r+k,1,N}. \quad (3.85)$$

Now rewrite Eq. 3.85 using the equality $(\text{vec}(\mathbf{ABC}) = (\mathbf{C}^T \otimes \mathbf{A}) \text{vec}(\mathbf{B}))$, where \otimes denotes the Kronecker product, and $\text{vec}(\mathbf{A})$ denotes the vector operation of stacking the columns of \mathbf{A} on top of each other [62]. Then Eq. 3.84 may be expressed as

$$J_{\min \mathbf{B}, \mathbf{D}} = \left\| \text{vec } \mathbf{\Omega} - \left(\sum_{i=1}^r \mathbf{U}_{r+k,1,N}^T \otimes \begin{pmatrix} \mathbf{E}_{1k} \\ \mathbf{E}_{2k} \end{pmatrix} \right) \text{vec} \begin{pmatrix} \mathbf{D} \\ \mathbf{B} \end{pmatrix} \right\|_{F^r}^2. \quad (3.86)$$

The solution for which is found by solving the least squares regression.

$$\text{vec} \begin{pmatrix} \mathbf{D} \\ \mathbf{B} \end{pmatrix} = \left(\sum_{i=1}^r \mathbf{U}_{r+k,1,N}^T \otimes \begin{pmatrix} \mathbf{E}_{1k} \\ \mathbf{E}_{2k} \end{pmatrix} \right)^{\dagger} \text{vec } (\mathbf{\Omega}). \quad (3.87)$$

The preceding summary of the N4SID algorithm is drawn on the considerable achievements of van Overschee and De Moor, for which they won the Automatica Prize for Paper of the Year 1994. The skeletal outline provided here, does not even begin to address some of the more complicated issues regarding a numerical implementation, for example, N4SID uses only the \mathbf{R} factor of a QR decomposition of the input-output Hankel matrices to calculate the state space model [1, 25]. However a more simple implementation of the algorithm, based wholly on using the SVD to perform the least squares calculations, is applied in part 2 of the dissertation, to calculate state sequences for a subspace method (SM) condition monitor.

3.5 Conclusion

In this chapter, an outline of subspace system identification has been provided. The chapter began with a literature review, in which many of the important recent publications in the arena of subspace system identification were mentioned. A background to the development of state space realisation theory from the 1970s was presented as a precursor to the subspace methods that have appeared in the past decade. A brief summary of the CVA and MOESP algorithms was presented, followed by a more in depth treatment of the N4SID identification procedure. The N4SID identification procedure provides the basis for the state space model identification used in part 2 of this dissertation to develop a dynamic modelling procedure for process condition monitoring applications.

Chapter 4

Identification of a simple simulated system

A mass-spring-damper simulation is used to demonstrate the properties of subspace system identification. The aim is to present a transparent study with which to evaluate the methodology. Several subspace algorithms are compared and contrasted. An ARX model structure is identified using the Matlab System Identification Toolbox, and compared to state space models identified with subspace methods, using the Akaike Information Criterion.

4.1 Introduction

In Chapter 3, subspace system identification was introduced as a method for calculating the parameters of state space models of LTI systems. The subspace methods are a recent innovation in the field of system identification. A recent study conducted by Qin and Badgwell [4], describes them as the next generation of linear identification techniques for model predictive control. Although well known in academic circles, their application in industry is yet to take hold. The work in this, and the next chapter, is aimed at clarifying procedures, and assessing aspects of their utilisation on industrial data. A very likely application of subspace system identification is to identify models for use in a control context. The aim of this study is to compare and demonstrate subspace

models, compared to currently preferred linear models used in the Model Predictive Control software package Connoisseur, namely ARX and FIR model structures. The use of subspace methods in a control context is addressed in [63]. For application of state space models in a control context see also [64]. It is expected that the summary and conclusions in Chapter 4 and Chapter 5 will help provide a platform for decisions concerning the software development program within the Control Technology Centre.

A suite of subspace algorithms based on N4SID [25] has been included in the Matlab System Identification Toolbox (MSIT). These have been proven to deliver accurate estimates of properly excited linear systems [1], however there are few published results concerning applications on industrial data with the notable exception of [8]. Unfortunately, the use of industrial data to analyse the properties and accuracy of linear models is fraught with difficulty, due to the complexity of industrial process data, and the uncertainties that exist, such as the presence of unmeasured disturbances, and lack of knowledge of the true system. Rather than clouding the results with unknowns, the aim in this chapter is to provide a more transparent comparison study of the application of subspace methods to the system identification problem. The platform that was developed in Chapter 2 and Chapter 3 is used in this chapter to demonstrate, assess and analyse subspace methods on the basis of a relatively simple linear system. The properties of the system used for this study are well understood, so that the results of the experiment can be more easily analysed. Four subspace algorithms are compared and contrasted with currently preferred linear methods in the software package Connoisseur, i.e. an ARX model is also applied to the same data sets and the results are compared. The two model structures compared in this study can therefore be summarised as (1) state space models, and (2) ARX models. The main basis for the comparison is their prediction accuracy across an infinite prediction horizon. Throughout the discussion the expression “subspace model” may sometimes be used to describe using a subspace algorithm to identify the parameters of a state space model. Subspace methods are, of course, identification procedures, not models, however, the expression should be unambiguous in the context in which it is applied.

The identification methods behind the five models in the comparison study are summarised in Table 4.1. The first two algorithms are taken from Overschee [1] and the

final three algorithms come from MSIT [66]. Note that $\mathcal{M}_2(\theta)$ is adapted for condition monitoring of industrial processes in Part II of this dissertation.

MODEL	NAME	DETAILS
$\mathcal{M}_1(\theta)$	subid.m	See Chapter three, pp 47-53, see also [1].
$\mathcal{M}_2(\theta)$	com_stat.m	This algorithm forms the basis for the subspace method developed in Part II of this dissertation.
$\mathcal{M}_3(\theta)$	n4sid.m	N4SID Property: “n4sid weight” = “moesp”. See [66].
$\mathcal{M}_4(\theta)$	n4sid.m	N4SID Property: “n4sid weight” = “cva”. See [66].
$\mathcal{M}_5(\theta)$	arx.m	model order = $n = p$ in Eq. 2.10. See [66].

Table 4.1 The methods used to model the mass-spring-damper system.

Each of the above subspace algorithms requires user choices that affect both the computational load and the accuracy of the identified model. However, the entire identification procedure can be automated, an option which is available in MSIT. The automation procedure involves performing an iterative search of a range of model/data structures, then choosing the best model according to the Akaike Information Criterion (AIC). However, the manipulation of user choices (as opposed to automation) allows for a more flexible approach to the modelling procedure, and also provides insight into the mechanics of the methods. The user choices are (i) the way in which the Hankel matrices built from the external measurements of the system are defined, and (ii) the decision concerning the order of dynamics to be possessed by the model. The best way to determine the model order remains an open question. Larimore [18] first identifies an ARX model to determine the appropriate dimensions for the Hankel matrices, and then uses the AIC to determine optimum system order. In contrast the N4SID algorithm [25] uses the relative magnitude of the singular values of the Q matrix in Eq. 3.59 to find an appropriate model order. A third method is cross-validation, where the final choice is based solely on the mean squared error of the model predictions. Each of these methods provides a quantitative measure of the system order.

A further analysis in this chapter involves the introduction of a time delay into the system. The aim of this experiment is to evaluate the efficacy of different strategies for modelling a system with a pure time delay. The ability to deal with time delays is an important issue, since many systems in industry contain significant time delays.

The data sets in this study are based on linear relationships, therefore each of the linear models is expected to produce good results. However the final comparison will highlight the mechanics of the identification process and advantages and drawbacks of each method. A description of the relatively simple and well-understood SISO system employed in this study now follows.

4.2 Simulation of a mass-spring-damper system

Figure 4.1 shows three masses joined in series to a rigid platform. Each mass is linked by a spring and damper in parallel, which provide proportional stiffness and damping. The system parameters are as follows: M_1 , M_2 and M_3 are each 1 Kg; C_1 , C_2 and C_3 are 15, 4, and 5 Nsecs/m respectively. The spring constants K_1 , K_2 and K_3 are 5000, 4500 and 6000 N/m respectively.

The system was simulated in Simulink, then sampled at a rate of 100 samples per second. The system has three modes with natural frequencies at 5, 15 and 20 Hertz. Figures 4.3 and 4.4 show the input and response of the system in the time domain. Figure 4.5 shows the system response in the frequency domain. A list of the modal frequencies and dampings and further details of the simulation are given in the Appendix.

The excitation signal is a normally distributed random sequence, corresponding to a force of 0.4 Newtons, applied at M_3 , and the system output is the displacement of M_2 , measured in meters. Five sets of 200 Monte Carlo experiments were performed. The data sets are as follows:

- Data set (**C1**) has white noise appended to the outputs only. The system and model are therefore : $S = f(u_k, v_k)$, $M(\theta) = f(u_k, \tilde{y}_k)$

- Data set (**C2**) has white noise appended to both inputs and outputs. The system and model are therefore $S = f(u_k, y_k)$, $M(\theta) = f(\tilde{u}_k, \tilde{y}_k)$.
- Data set (**C3**) has white noise appended to the outputs only. A pure time delay of 0.05 seconds is introduced into the data. The system and model are therefore: $S = f(u_k, v_k)$, $M(\theta) = f(u_{k-5}, \tilde{y}_k)$.
- Data set (**C4**) has white noise appended to the outputs only. A pure time delay of 0.15 seconds is introduced into the data. The system and model are therefore: $S = f(u_k, v_k)$, $M(\theta) = f(u_{k-15}, \tilde{y}_k)$.
- Data set (**C5**) has a filtered noise sequence appended to the outputs only. The system and model are therefore: $S = f(u_k, v_k)$, $M(\theta) = f(u_k, \tilde{y}_k)$. The noise sequence was generated by sending a white noise sequence, e_k , through a linear filter:

$$u_k = 0.85u_{k-1} + 0.02e_k - 0.04e_{k-1}. \quad (4.1)$$

The system is illustrated in Figure 4.2, where the appended noise sequences are derived from normally distributed random sequences and provide a noise to signal ratio (NTS) = 10% rms value of the deterministic signals.

For example, the NTS for **C1** is calculated as

$$\% \text{ NTS} = \frac{\text{rms value of the noise}}{\text{rms value of the uncorrupted output}} \times 100\%. \quad (4.2)$$

The high frequency noise sequences have constant variance, while the variance of the deterministic signals of the system changes with time (see Figure 4.4), therefore the NTS may be considered as time variant.

4.2.1 The Identification Procedure.

For each of the Monte Carlo experiments, the input and output data sequences were scaled to unit variance and zero mean. The data were also divided, the first half (corresponding to 2000 data points) used to train the model, and the second half used for model validation. The models were assessed using cross-validation, by calculating

the mean squared prediction error (MSPE), and the value of Akaike's Information Criterion (AIC). In addition the eigenvalues of the dynamical \mathbf{A} matrix were used to calculate the primary frequency of the system, where the eigenvalues of \mathbf{A} are z_1, \dots, z_n . Each eigenvalue z_i is a complex conjugate pair of the form $a \pm ib$ where

$$a \pm ib = e^{-\xi\omega T \pm i\omega\sqrt{1-\xi^2}T}. \quad (4.3)$$

Algebraic manipulation of the above expression leads to the following expressions for the modal frequencies and dampings of the system [9]

$$\omega = \frac{\sqrt{\left(\ln \sqrt{a^2 + b^2}\right)^2 + \left(\tan^{-1} \frac{b}{a}\right)^2}}{T}, \quad (4.4)$$

$$\xi = \frac{-\ln \sqrt{a^2 + b^2}}{\omega T}. \quad (4.5)$$

4.2.2 Choosing the model order

Given the stochastic nature of process data, increasing the order of the state space model (i.e. increasing the number of free parameters) leads to a more accurate fit of the model predictions to the training data. However, when the number of free parameters exceeds the minimum required to model the deterministic part of the system, the extra parameters only fit the noise in the measurements. This leads to the identification of models that do not perform as well on new data, and also contributes to the identification of unstable \mathbf{A} matrices. The end use for the model, such as the requirements for accuracy and robustness, will influence the final choice of model order, however low order models are generally desirable. They are robust, and from a control point of view, reducing the complexity of the model simplifies the calculations and usually produces a more robust controller.

Three approaches are considered in this study for finding the best model order with which to fit the data. They are

- (1) Cross-validation [6],
- (2) Akaike's Information Criterion (AIC) [7],

(3) The rank governing the singular values in Eq. 3.60 [1].

4.2.2.1 Cross-Validation

Cross-validation provides a good indication of the likely model order [6]. It works by evaluating the accuracy of the model on unseen data. The most commonly applied validation criterion involves the calculation of the mean squared prediction error (MSPE). Note that the input and output data is first scaled to unit variance, so that the MSPE values correspond to the cost function

$$V(\theta) = \sum_{k=1}^N \varepsilon(k, \theta)^T \lambda^{-1} \varepsilon(k, \theta), \quad (4.6)$$

where λ is the variance of the system output, and N is the number of data points in the validation data set. The attractive feature of cross-validation procedures is the simple and practical way they work - without the need for assumptions concerning the statistics of the noise or the system structure [6]. The risk of choosing too high a model order is limited because the validation procedure is carried out on unseen data.

One way to apply cross-validation is to calculate the prediction sum of squares (PRESS) statistic [67] using all available data, except a single sample that is kept unseen for validating the model. The model is calculated using the remaining $N-1$ points, and then the prediction error is calculated for the remaining data-point. This procedure is repeated N times and then a sum of squares error is calculated. This PRESS statistic can be calculated for all the candidate models.

In contrast, the method used in this study takes validation data from each of 200 Monte Carlo experiments and calculates the MSPE for each. The main drawback is the large amount of computation and that the results are sometimes contradictory, where the performance of different models varies, depending on the particular data set that is used for validation.

The MSPE measure is affected by the initial state (starting point) used for generating the model predictions. To guarantee an accurate initial state for the validation data set, the following procedure has been implemented

- (1) Divide each of the Monte Carlo experiments into a training set ($N/2$ data points) and a validation data set ($N/2$ data points).
- (2) Calculate the model from the training set.
- (3) Generate the model predictions by passing the entire input sequence through the model.
- (4) Assess the prediction error; let $R = N/2$, then

$$\text{MSPE} = \frac{1}{R} \sum_{k=R+1}^N (\hat{y}_k - \tilde{y}_k)^2. \quad (4.7)$$

4.2.2.2 Akaike Information Criterion (AIC)

AIC [7] provides a quantitative way for determining whether increasing the number of free parameters in the model structure has significantly improved the prediction accuracy of the model. When the gain in accuracy isn't large, then a lower order model may be better because it is more likely to be robust, and it will reduce the complexity of the control problem. AIC takes into account the number of free parameters in the model structure, and can be used to compare different parametric model structures. It is based on calculating a maximum likelihood estimator [6], which aims to maximise the probability of an observed event occurring. For each model calculate

$$\text{AIC} = -2(\text{maximum log likelihood}) + 2 D_M. \quad (4.8)$$

If the measurement error on the data is assumed to conform to a zero-mean, Gaussian distribution of constant variance, then the MSPE measure can be used in Eq. 4.8 [6]. D_M is the number of free parameters in the model. Candy [68] showed that the number of functionally independent parameters in a state space model is less than the total number of parameters in the various state space matrices. The number of free parameters in a state space model is

$$D_M = (2r + m)l + ml + \frac{1}{2}(l + 1)l. \quad (4.9)$$

In Eq. 4.9, r is the number of states, l is the number of outputs, and m is the number of inputs to the system. For the SISO system under consideration here, the following calculation for AIC has been employed

$$AIC = \log(\text{MSPE}) + \frac{2}{N}(3r + 2). \quad (4.10)$$

In Eq. 4.10, the system measurements are pre-scaled to zero mean and unit variance, and N is the number of data points in the validation set.

4.2.2.3 Using the rank of Q

N4SID methods estimate the order of the system according to the relative magnitude of the singular values in Eq. 3.45. The determination of the order of dynamics involves finding a cut-off point that differentiates between “system” and “noise”. This method works in the same way as Kung’s Algorithm, where the left singular vectors lead directly to an estimate of the order of the extended observability matrix. Eq. 3.59 in the N4SID algorithm corresponds to Eq. 3.3 in Kung’s algorithm, where the “effective rank” of each is used to estimate the order of dynamics. However the choice of system order is non-trivial because the stochastic nature of process data means that there is often no clear cut-off point. Several possibilities exist for deciding on the cut-off point: visual inspection can be used, or the model order can be chosen based on the percentage variance V , captured by the first n singular values [9]:

$$V = \frac{\sum_{i=1}^n \sigma_i}{\text{trace}(\mathbf{S})} \times 100\%. \quad (4.11)$$

An alternative method [9] is to determine a limiting value for the term

$$\frac{|\sigma_n|}{|\sigma_{n+1}|}, \quad (4.12)$$

where σ_i is the i^{th} singular value on the main diagonal. For a deterministic system, Eq. 4.12 goes to infinity. If a good estimate of the variance of the noise is available then the cut-off point may be indicated by

$$\sigma_{n+1} > kx, \quad (4.13)$$

where x is a measure of the variance of the system and k is a value to be determined. Eq. 4.13 may be useful if a good estimate of NTS is available. Then the NTS may be considered for each of the modes separately. For modes that are governed by small singular values, the NTS will be proportionately higher, meaning that the model parameters describing this mode are more likely to be corrupted.

4.2.3 Residuals Analysis

A residuals analysis includes assessing the mean and autocorrelation function of the residuals:

$$\hat{\mu}_\varepsilon = \frac{1}{N} \sum_{k=1}^N \varepsilon_k, \quad (4.14)$$

$$\hat{r}_\varepsilon^N(\tau) = \frac{1}{N} \sum_{k=1}^N \varepsilon_k \varepsilon_{k-\tau}. \quad (4.15)$$

For a system subject only to white noise, the residuals of a good model will be zero mean and randomly distributed (i.e. the model captures the dynamics and rejects the noise). Various tests for the whiteness of the residuals can be employed [6, 69], for example the autocorrelation function can be used to test whether

$$\frac{N}{(\hat{r}_\varepsilon^N(0))^2} \sum_{\tau=1}^K (\hat{r}_\varepsilon^N(\tau))^2 < \chi^2(K), \quad (4.16)$$

where K is the number of variables, and N is the number of data points, i.e. does the LHS of Eq. 4.16 fall within a Chi Squared distribution? Another test is to calculate the number of sign changes in the residual sequence $\varepsilon(t)$, which should tend to $N/2$ as $N \rightarrow \infty$ [5].

The independence between past inputs and the residuals is indicated by the cross-correlation function:

$$\hat{r}_{\varepsilon u}^N(\tau) = \frac{1}{N} \sum_{k=1}^N \varepsilon_k u_{k-\tau}. \quad (4.17)$$

The correlation function between the residuals and the past inputs indicates the level to which the causal effect of the inputs on the process have been captured by the model. Provided the model has captured the important dynamics of the process then the correlation function will be small. The presence of significant correlation may indicate that the model structure is not correct or perhaps the presence of unmeasured inputs to the process.

4.2.4 Pure Time Delay

Transport delays in petrochemical and other continuous industrial processes are often encountered. It is therefore important to develop a strategy for subspace system identification of processes with transport delays.

A linear model structure Eq. 2.2 admits a pure time delay [5], where for a pure time delay of L samples, the first L parameters of the parameter vector $g(\theta)$ equal zero, i.e.

$$g_i(\theta) = 0, \quad i = 1, 2, \dots, L \quad (4.18)$$

This is equivalent to not including the time lags $1, 2, \dots, L$ in the model structure. However the complexity of process data, or lack of knowledge about the true process, leads to uncertainty concerning the length of the true time delay. If the delay in the model structure is longer than the actual delay in the system, then the model breaks down completely [5, 6].

To investigate the use of subspace system identification of linear systems with pure time delay, Monte Carlo (200) experiments were created containing delays of 50 milliseconds (**C3**) and 150 milliseconds (**C4**). The correlation plot (Figure 4.15) shows how shifting forward the input sequence of **C1** affects the cross-correlation between the input and output of the model. The aim is to assess the effect of the time delay on the efficacy of the subspace procedures, and to develop a strategy for working with subspace methods on systems with pure time delays. Two procedures are considered:

- 1) Increase the size of the Hankel matrices so that the covariance matrix in Eq. 3.52 will model the effect of the delay.
- 2) Introduce time-shifted input sequences into the input Hankel matrices.

4.2.5 User Choices for the Subspace Algorithms

For each of the subspace algorithms, there are three user choices to make:

- (1) The maximum forward prediction horizon, r_1 , used by the algorithm.
- (2) The number of past inputs, r_2 , that are used to build the block Hankel matrix \mathbf{U}_p .
- (3) The number of past outputs, r_3 , that go into the formation of \mathbf{Y}_p .

r_1 is the number of block rows in the matrix \mathbf{Y}_f , where using r block rows in the formation of $\hat{\mathbf{Y}}_f$ corresponds to the calculation of an r -step ahead predictor. r_2 corresponds to the number of input lags used in the r -step predictor [31].

The assumption $r_1 = r_2 = r_3$ can be relaxed, however, subspace algorithms [1, 25, 32, 58] use $r_1 = r_2 = r_3$, so that a single choice of the number of block rows, r_r , is sufficient to fully define the projection matrix in Eq. 3.52. r_r defines the number of singular values in Eq. 3.60 and therefore limits the maximum order of the state space model that can be identified, i.e. the user must specify $r_r >$ the maximum order of the state space model required. The choice of r_r affects the accuracy of the predictions and the computational expense. Specification of r_r is explored further in the next section, where a range of values for r_r is used, and then the results compared according to the maxim “the best model obtainable with a reasonable amount of work” [6].

4.3 Results

The satisfactory performance of linear methods on industrial process data depends on the linearity of the process. Industrial process data is tainted by the presence of various noise sources, problems with sensors, and particularly unmeasured disturbances (e.g. variations in the quality of feed streams). This leads to a degree of uncertainty in measuring the performance of different models. Therefore a simple, well-understood linear system has been used in the study here, as a basis for comparing the linear methods under investigation. Figures 4.3 and 4.4 show the deterministic signals and

corresponding signals with gaussian noise added. Figure 4.5 shows the frequency response of the system from 1-50 Hertz.

The user choices for the subspace methods are as follows:

- (a) The order of model, n .
- (b) The number of block rows, r . Unless specified otherwise, $r_1 = r_2 = r_3$. In this study the number of block rows is set by declaring the value B , where $r = n + B$.

This investigation will consider the following:

- (1) A comparison of the performance of ARX and state space models on the basis of MSPE and AIC.
- (2) Various methods for determining the order of dynamics to be possessed by the model will be considered: using cross-validation, AIC and inspection of the singular values.
- (3) The effect of the number of block rows on model accuracy.
- (4) Procedures for dealing with a pure measurement delay.
- (5) The effect of noise on the stability of the identified model.
- (6) The effect of noise on the covariance of the parameter error matrix.
- (7) Identification of the primary frequency of the system.

4.3.1 Measures to determine the optimum model structure

AIC can be used to determine the best model structure with which to model the system. Subject to the statistical properties of the noise, the minimum value of AIC is generally considered to coincide with an optimum model structure [20]. However, the following results reveal that the minimum value of AIC may sometimes lead to higher order model structures than might be required according to other criterion, for example, a robust, simple model structure that does the job.

The first identification experiment involves system data with white noise appended to the output (dataset **C1**). For each of 50 Monte Carlo experiments, algorithms M_1 and M_2 were used to identify state space models. In each case a range of model orders was tried. To find the model with the best prediction accuracy, the number of block rows used in the algorithms was also varied. The mean and standard deviation of the MSPE across the 50 Monte Carlo experiments, for each model order was also calculated.

Table 4.2 illustrates the breadth of the search for the model with the best prediction accuracy. It contains results from a cross-validation on the basis of a single experiment taken from **C1**. Model orders $n=1:7$ were calculated for $B=0:20$, using subspace methods M_1 and M_2 . Note that the total number of block rows used in each of the algorithms is $n+B$. The minimum MSPE for each model order is indicated by the shaded regions in bold.

Table 4.3 summarises the results from all 50 Monte Carlo experiments. In terms of prediction accuracy, an 8th order model is indicated for algorithm M_1 , however this is only marginally more accurate than 6th order ($mse = 0.9975$) and 4th order ($mse = 0.9994$) models. Table 4.3 also shows the corresponding values of AIC. The minimum MSPE is at $n = 7$ for algorithm M_2 . Note that AIC is minimised at $n = 4$ for models M_1 and M_2 even though higher order models provided the greatest prediction accuracy. Note also that the AIC for 2nd order models produced by algorithms M_1 and M_2 is also very close to the minimum AIC. The shaded regions in bold indicate the model structures corresponding to minimum AIC. It is of some interest that the true system order is 6 however the minimum AIC is found for 4th order state space models. An explanation may lie in the variance of the noise, which is such that the third mode becomes insignificant. For the deterministic system, the minimum value of AIC for all six models was found for 6th order models.

Table 4.4 shows the mean and standard deviation of the mean squared prediction error for models identified using algorithms M_1 and M_2 . This summarises the results for 50 datasets taken from **C1**, where white noise was appended to the deterministic output of the system. The minimum mean squared prediction error was found to correspond to a 4th order state space model for both algorithms M_1 and M_2 . This suggests that while

the true system order is 6, there is relatively little power in the third mode of vibration, which is masked by the noise in the system. The shaded regions in bold indicate the model structures corresponding to minimum *mse*.

Table 4.5 displays results based on 50 experiments from dataset **C2**, where white noise has been appended to both the deterministic input and the deterministic output. This situation corresponds to the case where there is both process and measurement noise on the system. The MSPE for both algorithms \mathbf{M}_1 and \mathbf{M}_2 is now significantly higher than for the **C1** dataset, that includes measurement noise only. The shaded regions in bold indicate the model structure corresponding to minimum mean *mse*. The minimum MSPE is found to correspond to a 3rd order model using algorithm \mathbf{M}_1 , and to a 2nd order model using algorithm \mathbf{M}_2 . It now appears that the presence of process noise has further masked the true order of the system.

4.3.2 The prediction accuracy of state space and ARX models

In this section, the performance of subspace methods is compared with that of ARX models. Cross-validation and AIC are used to determine optimum model structures. Tables 4.6 and 4.7 convey the results of the same cross-validation procedure as previously applied, this time for algorithms \mathbf{M}_3 , \mathbf{M}_4 and \mathbf{M}_5 .

Table 4.6 displays results from modelling the system with algorithms \mathbf{M}_3 , \mathbf{M}_4 and \mathbf{M}_5 from the Matlab System Identification Toolbox. \mathbf{M}_3 and \mathbf{M}_4 are subspace algorithms with weightings “moesp” and “cva” respectively. \mathbf{M}_5 is an ARX model structure. Given the linear nature of the data and the gaussian distribution of the measurement noise, it is expected that providing an appropriate model structure is used, reasonably good models should be obtained using all three algorithms.

The minimum MSPE was found for 6th order models for algorithms \mathbf{M}_3 and \mathbf{M}_4 . As indicated in Table 4.6, the models produced by \mathbf{M}_3 and \mathbf{M}_4 are marginally more accurate than the ARX models produced by \mathbf{M}_5 . A possible explanation for the superior performance of the subspace algorithms is that they exploit the SVD to produce an “inner model”, i.e. the state equation, and use the states, which are derived from linear

combinations of the input-output data. This gives the model an advantage over the ARX model structure, which depends on previous input measurements and output predictions to predict future outputs. Note that the ARX models were constructed with the same number of regressive terms and input delay spread, i.e. $n = p$ (see Eq. 2.9), and it may be possible to obtain a marginal improvement in accuracy if the input delay spread is varied independently of the number of regressive terms.

AIC for M_3 and M_4 on the basis of **C1** indicates that the system order is 4. Note that there is very little difference between the mean MSE and AIC for M_3 and M_4 . Although a 4th order model is indicated by AIC, the values of AIC for second order models is only marginally higher, so that a 2nd order model of the system may also be considered on this basis.

Table 4.7 shows results from the **C2** dataset, i.e. deterministic data with both process and measurement noise appended. The accuracy of the predictions of all three algorithms M_3 , M_4 and M_5 has been compromised by the presence of process noise. A 3rd order state space model now provides the best prediction accuracy in the case of algorithms M_3 and M_4 . This suggests that the higher order states of the process have been corrupted by the addition of process noise.

A range of ARX model structures (M_5) was investigated. The minimum MSPE for M_5 is shown in column 10 of Tables 4.6 and 4.7. This involved using MSIT in a search across 50 Monte Carlo experiments, using ARX model orders 1:61 in each case. The values in brackets indicate the model order for which the relevant statistic is given. A 12th order model is indicated on the basis of **C1** and a 8th order model on the basis of **C2**. However, as with the AIC values for the subspace models, lower order ARX models returned only marginally higher values, indicating that 4th and 6th order models could also be considered.

Figure 4.6 shows a residuals analysis for a 4th order model of the **C1** data identified using algorithm M_1 . The residuals are near zero mean and with no correlation in time, indicating that the model produced by algorithm M_1 has accurately captured the dynamics of the system.

Figure 4.7 summarises the effect of increasing the ARX model order on prediction accuracy. MSPE is plotted with confidence limits representing the mean value $\mu \pm 3\sigma$ for model orders 1:10. Figure 4.7(a) shows results for Monte Carlo (50) data with measurement noise only (**C1**), and Figure 4.7(b) shows results for Monte Carlo (50) data with measurement and process noise. The results show that the prediction accuracy is improved as the order of the ARX models is increased. Prediction accuracy levelled out beyond order 10, with the best model structures being 12th order (**C1**) and 8th order (**C2**).

The Tables (4.2 – 4.7) and Figures (4.6 – 4.7) have presented results from identification experiments using state space and ARX model structures. Prediction accuracy and AIC have been used to assess the quality of each of the models. The results show that the subspace models have performed *at least as well* as the ARX models on each of the noise configurations that have been considered. The results also suggest that using measures across several validation experiments can lead to conflicting results. However, on the basis of AIC, the subspace algorithms \mathbf{M}_1 , \mathbf{M}_2 , \mathbf{M}_3 , and \mathbf{M}_4 have provided more powerful models than the ARX algorithm \mathbf{M}_5 . In control applications, it is considered advantageous to opt for as simple a model structure as possible that achieves the required accuracy. In this respect the subspace methods have been shown to have an advantage over ARX structures because they utilise the states of the process to provide a more parsimonious description of the system. For example, a comparison between Figure 4.7(a) and Table 4.4 reveals that 2nd and 4th order state space models have provided a level of prediction accuracy equivalent to ARX model structures that use a much larger number of parameters.

4.3.3 Using the singular values to determine model order

In this section, the N4SID method for determining an appropriate order of dynamics to be possessed by the model is demonstrated.

The singular values occupy the main diagonal of \mathbf{S} in Eq. 3.60 in non-ascending order. They govern the amplitude of the principal directions (or states) of the process. Figure 4.8(a) shows bar charts of the magnitude of the singular values from algorithm \mathbf{M}_1 , $r = 18$, for single experiments taken from **C1** (left bar chart) and **C2** (right bar chart). In

each case 2 singular values are significantly larger than the rest, so that the cut-off point (obtained by visual inspection) indicates that a 2nd order model is a very likely candidate with which to model the system.

The effect of a pure time delay (experiments taken from **C3** and **C4**) on the singular values is shown in Figure 4.8(b). The pure time delay, if not accommodated into the model structure, acts as a non-linearity. The bar chart shows the magnitude of the singular values for algorithm M_1 , $\beta = 13$, where the apparent order of the system is now 3rd, 4th order or even higher.

4.3.4 The effect of noise

In this section, the advantage of using low order, more robust models (at the possible expense of a slight loss in accuracy) is highlighted. Second, fourth, and sixth order models are investigated in terms of their consistency in estimating the primary frequency of the system.

An iterative search found that some high order models ($n > 6$) the greatest prediction accuracy in terms of MSPE, even though the deterministic system is known to be 6th order. Furthermore, the minimum value for AIC for algorithms M_1 and M_2 indicated that the system may be fourth order. Although the system has three modes, the noise has obscured the presence of the third mode. In fact it was found that AIC for 2nd order models was only marginally higher. Furthermore, a visual inspection of the singular values (Figure 4.8(a)) revealed that a second order model may be best. One of the advantages of using lower order models is that they tend to be more robust where higher order models might be more easily corrupted by the influence of noise and/or unmeasured disturbances. This effect is demonstrated in Figures 4.9-4.12 where Monte Carlo (200) experiments from **C1**, **C2** and **C5** have been performed, in each case using algorithm M_1 , with the number of block rows, $r_r = 18$. The eigenvalues of 2nd, 4th and 6th order model estimates have been plotted to illustrate that lower order subspace models tend to be more robust, because higher order models have modes that are more sensitive to noise. For example, considering Eq. 4.13, the NTS may be considered for each of the 3 modes of the system separately. Where the singular values are relatively large, the NTS is low. However for modes that are governed by small singular values,

the NTS is high, meaning that the model parameters describing this mode are more likely to be corrupted.

Figure 4.9 shows a comparison of the consistency of the model estimates, between the case where white noise has been appended to the system output and the case where coloured noise has been appended to the system output. Algorithm M_1 , with the number of block rows of data used to calculate the state space model, $r_r = 18$, was applied to each of the 200 datasets from **C1** and **C5** to calculate 2nd order state space models. Note that Figures 4.9(a) and 4.9(b) are highly magnified to show tight clusters of eigenvalues centred on $0.943 \pm 0.3i$. Figure 4.9(a) plots the eigenvalues of the dynamical **A** matrix from the 200 experiments in **C1**. Figure 4.9(b) plots the eigenvalues of the dynamical **A** matrix from the 200 experiments in **C5**. Although marginal, it can be seen that the spread of eigenvalues is greater for the **C5** data than for the **C1** data, i.e. the consistency of the estimates has been compromised slightly by the presence of coloured noise. However the tightness of the clusters indicates that the 2nd order models have provided robust and reliable estimates of the primary natural frequency of the system. On the basis of the 200 **C1** experiments, the 2nd order estimates of system had mean $\mu = 4.968$ Hz, and standard deviation $\sigma = 0.0016$ Hz. For the 200 **C5** experiments, the 2nd order estimates of the primary frequency of the system had mean $\mu = 4.968$ Hz, and standard deviation $\sigma = 0.0020$ Hz. (The primary frequency of the true system is in the region of 4.959 Hz).

The same procedure was followed to identify 4th order state space models of the system. 200 estimates of **C1** and 200 estimates of **C2** were obtained using algorithm M_1 ($r_r = 18$). Figure 4.10 shows the results obtained for 4th order models. In particular, the effect of the process noise on the estimates is noticeable. In the case of **C1**, the 200 estimates of the first mode of vibration are all centred on the '+' at $0.94 \pm 0.3i$. The 200 estimates of the 2nd mode of vibration show considerable spread in the region of $0.25 \pm 0.9i$, and 2 spurious modes appear on the real axis. In the case of **C2**, the dramatic effect of the process noise can be seen in Figure 4.10(b). The 200 estimates of the first mode of vibration are all centred on the '+' at $0.94 \pm 0.3i$. However there is a lack of consistency regarding the second mode of vibration. 34 spurious modes appear on the real axis, one outside the unit circle. The remaining 83 pairs of eigenvalues,

corresponding to the oscillatory 2nd mode are now very considerably spread about $0.25 \pm 0.9i$.

Figure 4.11 shows the eigenvalues of the dynamical **A** matrices of 6th order estimates of **C1** and **C2**. Algorithm \mathbf{M}_1 , with the number of block rows of data used to calculate the state space model, $r_p = 18$, was applied to each of the 200 datasets to calculate 6th order state space models. Figure 4.11(a) shows the results for dataset **C1**. A significant number of eigenvalues are real, and 9 others lie outside the unit circle. In the case of dataset **C2**, there are 16 unstable oscillatory modes, i.e. 8% of the models identified are unstable. These unstable and spurious modes are evidence that the presence of 10% NTS on the system measurements has led to severe masking of the tertiary dynamics of the system. The robustness of these higher 6th order estimates is clearly more suspect than for the lower order estimates.

Figure 4.12 plots the eigenvalues of the dynamical **A** matrices of state space model estimates of the 200 experiments in **C5** (data with coloured measurement noise added). The coloured noise case is particularly important, as this is the case most commonly found in industry. In each case, Algorithm \mathbf{M}_1 was applied to each set of the data, with the number of block rows of data used to calculate the state space models, $r_p = 18$. The results indicate how the consistency and robustness of the identification procedure is affected by the identification of higher order models. There is also the opportunity to observe the performance of algorithm \mathbf{M}_1 when faced with data tainted with coloured noise, compared to the white noise case in Figures 4.10 and 4.11.

Figure 4.12(a) plots the eigenvalues of 4th order state space models of the 200 experiments in **C5**. The 200 estimates of the first mode of vibration are all centred on the '+' at $0.94 \pm 0.3i$. These correspond to the cluster shown for the **C5** 2nd order state space models shown in Figure 4.9(b). Of the eigenvalues corresponding to the 2nd mode of vibration, just 6 are oscillatory. This indicates that the coloured noise has almost entirely masked the 2nd mode of vibration. A direct comparison with Figure 4.10(a) which shows the case where white measurement noise has been appended, reveals that the white noise has not masked the second mode of vibration in the way that the coloured noise has.

Figure 4.12(b) plots the eigenvalues of 6th order state space models of the 200 experiments in **C5**. The 200 estimates of the first mode of vibration are all centred on the '+' at $0.94 \pm 0.3i$. These correspond to the cluster shown for the **C5** 2nd order state space models shown in Figure 4.9(b), and represent a vibratory mode in the region of 5 Hertz. A second cluster of 200 eigenvalues appears in the region $0.3 \pm 0.75i$ corresponding to an oscillatory mode in the region of 20 Hertz, i.e. the third mode of vibration (see Eq. 4.4). The second mode of vibration at 15 Hertz has been almost totally masked by the coloured noise. A comparison with Figure 4.11(a) which shows the results for white measurement noise reveals a cluster in the region of $0.58 \pm 0.72i$, which corresponds to the second mode of vibration at 15 Hertz (using Eq. 4.4). The damping effect of the first order, coloured noise sequence is also apparent. In the case of the 6th order models used to identify the **C1** system with white measurement noise (Figure 4.11(a)), there are 9 eigenvalue pairs that lie outside the unit circle, corresponding to unstable oscillatory modes. For the 6th order estimates of data set **C5**, there are no unstable oscillatory modes, however there is the presence of 12 eigenvalues of magnitude greater than 1 lying on the real axis.

4.3.5 The number of block rows used and model accuracy

In this section, the question of how to choose the number of block rows, r , used in the subspace model calculations is considered.

An iterative search for the most accurate model, using $r = n + B$, where $B = 0:20$, $n = (2, 4)$, was conducted. For the 2nd and 4th order models, the value of B_{opt} is defined as that which minimises MSPE. The values of B_{opt} for Monte Carlo (100) **C1** and **C2** are recorded in the relative frequency histograms in Figure 4.13. For **C1**, and 2nd order models, B_{opt} is frequent in the region $B = 7:10$. For **C1**, and 4th order models B_{opt} is more evenly distributed. The structured distribution in these bar charts can be used as an indication for choosing r . Note that increasing the number of block rows when working with **C2** tends to improve the model's performance, where $B = 60$ is very frequent. The indication seems to be that, in the case of measurement noise only, fewer block rows are required, however, if process noise is present, using more block rows improves the accuracy of the resulting model.

The best way to find B_{opt} remains an open area for research, however the idiosyncratic nature of industrial data suggests that an iterative search for each experiment is required.

Figure 4.14 further considers the effect of the number of block rows used in the calculations on model accuracy. Algorithms \mathbf{M}_3 and \mathbf{M}_4 were applied to data tainted with coloured measurement noise (**C5**). In all 200 Monte Carlo experiments were performed to calculate the mean MSPE, as a function of the number of rows. The error bars in Figure 4.14 to show the 99% confidence limits, i.e. $\mu \pm 3\sigma$, where σ represents the standard deviation of the MSPE. \mathbf{M}_3 and \mathbf{M}_4 differ in the way in which the initial projections are calculated. \mathbf{M}_3 uses a MOESP weighting and \mathbf{M}_4 uses a CVA weighting, however the results for methods \mathbf{M}_3 and \mathbf{M}_4 are remarkably similar. Given that the computation expense increases significantly as extra block rows are added to the input and output Hankel matrices, it is important to find a satisfactory trade off between the number of block rows used in the calculations and the prediction accuracy obtained. Figure 4.14 indicates that the prediction accuracy improves by using more block rows. Based on the results in Figure 4.14, $r = 7:10$ seems about right, where the total number of block rows used in the algorithms is $B = n + r$. Note that the results presented in Figure 4.14 tie in with the corresponding bar chart, Figure 4.13(a).

4.3.6 Procedures for dealing with time delays

In this section, strategies for dealing with systems with known transport delays are considered.

Figures 4.15 - 4.18 show the results obtained from identification experiments on systems with a measurement time delay. Figure 4.15 shows the cross-correlation for data segments taken from **C1**, **C3** and **C4**. Measurement delays of 50 milliseconds, and 150 milliseconds, for **C3** and **C4** respectively, are clearly indicated by the shifted values of the cross-correlations.

Figure 4.14 considers the ability of the subspace algorithms to deal with data that contains a measurement time delay. Algorithms \mathbf{M}_3 and \mathbf{M}_4 were applied to the time-shifted dataset **C3**, with a measurement delay of 50 milliseconds. In all 200 Monte Carlo experiments were performed to calculate the mean MSPE, as a function of the

number of rows. The error bars in Figure 4.14 to show the 99% confidence limits, i.e. $\mu \pm 3\sigma$, where σ represents the standard deviation of the MSPE. The 2nd order models have not coped with the time delay, however, the 4th and 6th order models have performed better.

An alternative way to cope with pure time delay, rather than using extra states is by including time-shifted inputs in the model. A SISO system becomes multi input single output (MISO). This leads to an increase in dimension of the **B** matrix, where for a delay spread of L samples, the dimension of the **B** matrix increases from $\mathbf{B} \in \mathbb{R}^{r \times m}$ to $\mathbf{B} \in \mathbb{R}^{r \times ml}$. In Figure 4.17, the results for \mathbf{M}_1 are shown for a measurement delay of 50 milliseconds. The number of input lags on the x-axis indicates the length of the delay spread. For the 2nd order model, the input delay spread needs to span the entire time delay. However, higher order models provide better estimates when the delay spread used on the input side of the model does not sufficiently cover the total delay (which is in this case equivalent to 5 samples).

Providing that the model structure is correct, the parameters of the linear model will reflect the transport delay. Figure 4.18 shows state equations for methods \mathbf{M}_1 and \mathbf{M}_2 , 2nd and 6th order models. The true system is $y_k = f(u_{k-5})$. For 6th order \mathbf{M}_1 , the states x_1 and x_2 depend only on u_k and u_{k-6} . The extra states x_3, x_4, x_5 and x_6 depend only u_k . Including the lagged inputs has effectively linearised the system where the state equations for \mathbf{M}_1 correspond to Eq. 4.18.

4.3.7 MSPE as a random variable

In this study, 200 Monte Carlo experiments have been used in an attempt to provide a statistical analysis of some of the properties of subspace system identification. The aim is to consider the overall performance of different models, as it varies from data set to data set.

The stochastic influence of the white noise leads to uncertainty in the MSPE calculations, i.e. the MSPE can be considered as a normally distributed random variable with mean μ and standard deviation σ . Figures 4.19 – 4.24 show how the MSPE for a single model has varied across the data sets **C1** and **C2**. The MSPE for each of the 200

experiments is plotted, for methods M_1 , M_2 , M_3 and M_4 . For each graph, models ($B = 19$) were identified, on the basis of single data segments from each of **C1** and **C2**.

As can be seen in the figures, the prediction error of each model is clearly a function of the data on which it is tested, where there is a high degree of correlation amongst the models from data segment to data segment. Figure 4.19 shows the results from the 200 data segments that contain measurement noise only (**C1**). The MSPE measures of the 2nd order models, **C1**, indicate that algorithms M_1 and M_2 have consistently performed better than algorithms M_3 and M_4 . This may explain why the eigenvalue plot (used by M_1 and M_2 to determine the system order) indicated that a 2nd order model structure was best, yet AIC (used by M_3 and M_4 to determine the system order) indicated a 4th order model structure was best. For the 4th order models, the prediction error of the M_3 model (Figure 4.20) was the least on many of the data segments, however, the difference in performance of the four algorithms was less pronounced than for the 2nd order models. For 6th order models, each of $M_1(+)$, $M_2(o)$ and $M_3(x)$ had the least prediction error on occasions; however, $M_4(\diamond)$ was often the most inaccurate.

Figures 4.22 - 4.24 show how the **C2** Monte Carlo (200) data set is divided into five data segments (or batches) each containing 40 experiments. Each of the 5 batches contains a different input noise sequence (white noise with the same variance) appended to the deterministic input. Each of the batches has (the same) 40 different output noise sequences appended to the deterministic output to create the 200 different experiments. Figures 4.22 - 4.24 show the extent to which the relative performance of each of the algorithms depends on the appended noise sequences. For example $M_2(o)$ has consistently scored best on segments 2 and 5 in Figure 4.24, but not on segments 1, 3 and 4.

4.4 Conclusion

In this chapter, a SISO, 3DOF mass-spring-damper system has been used to compare the performance of six system identification algorithms. The data from the deterministic system was appended with stochastic sequences, and measurement time delay, to create five data sets, each consisting of 200 Monte Carlo experiments. The mean squared

prediction error (MSPE) was used in conjunction with cross-validation to assess the performance of the algorithms. In addition, the eigenvalues of the \mathbf{A} matrix were used to provide a measure of the covariance of the error of the identified parameter matrices.

In summary:

- (1) A comparison of ARX and state space model structures has been carried out using the Matlab System Identification Toolbox. AIC was calculated on the basis of two data sets: (1) data with measurement noise only, (2) data with measurement and process noise. AIC indicated that on the basis of the trade-off between model complexity and model accuracy, the state space models, identified using subspace system identification methods, provided more accurate models *for a given complexity*, than the ARX models.
- (2) Two methods for obtaining the appropriate order of the state space models were considered. Cross-validation was used, in conjunction with AIC, to determine the model order for methods \mathbf{M}_3 and \mathbf{M}_4 . In contrast, visual inspection of a bar chart showing the relative magnitude of the singular values provided an estimation of the system order in the case of \mathbf{M}_1 and \mathbf{M}_2 . For data with measurement noise only, 4th order models using \mathbf{M}_3 performed best on the basis of mean MSPE. However, \mathbf{M}_1 provided the best second order models.
- (3) The effect of various configurations of noise on the model estimates has been considered. The noise has created difficulty in accurately identifying the secondary and tertiary modes of the system. This lead to an increase in the covariance of the parameter error for 4th and 6th order models. This effect has been illustrated using a series of eigenvalue plots which indicate the robustness of lower order models.
- (4) The work on time delays considered two approaches for subspace identification of process data with transport delays. Increasing the order of the state space model was shown to improve the prediction accuracy. However, a better approach is to incorporate the delay into the model structure by introducing time-shifted input variables into the input Hankel matrices. Providing the time-shift is large enough, this approach was shown to effectively “linearise” the system, leading to an accurate and robust model. It is therefore important to obtain a good estimate of the likely

process time delay, using, for example, correlation analysis and (if possible) knowledge of the dynamics of the process.

- (5) Monte Carlo experiments were conducted to consider the question of the number of block rows to use in the subspace algorithms. The indication is that, in the case of measurement noise only, the number of block rows for the algorithms needs only to be marginally higher than the order of the system dynamics. However, when process noise was present, using more block rows leads to more accurate models.
- (6) Even though the second order statistics of the noise were equivalent for each of the 200 Monte Carlo experiments, the number of block rows that minimised MSPE varied considerably. The idiosyncratic quality of process data means that trial and error will often be required, in order to find the best number of block rows to use in the subspace algorithms. The procedure can, however, still be automated, by iterating through different values of r , then using cross validation. Alternatively a PRESS statistic can be used, for example, an iterative search is adopted by Ljung [66], where the search is carried out on the basis of a single data set. While AIC provides a useful measure of the trade-off between model complexity and prediction accuracy, the results can be misleading when working with process data, for example unmeasured disturbances will corrupt the AIC (and other) measures.

In conclusion:

Provided an appropriate model structure is used, and the system is reasonably linear, all the linear methods considered in this study are expected to identify reasonably accurate models. Keeping this in mind, the following points might be kept in mind by application engineers wishing to exploit the subspace approach:

Overall there is no hard and fast rule for determining the best model structure, even for the simple example given in this chapter. However, because subspace methods use state space models, there is only one choice to make, i.e. the model order. Measures for determining the best model structure will involve the use of the magnitude of the singular values in the form of a bar chart as in algorithms \mathbf{M}_1 and \mathbf{M}_2 . In addition to this, cross validation measures can be used as employed \mathbf{M}_3 and \mathbf{M}_4 . Any or each of these measures can be used for identifying an appropriate model structure. However,

because industrial data is idiosyncratic due to noise sources and unmeasured disturbances, if many measures and comparisons are made, this will often lead to conflicting results. In general the rule remains – choose the least complex model structure possible to do the job. In this respect subspace methods have an advantage over the use of ARX and FIR model structures, as demonstrated in the results in this chapter, because state space models use state directions to describe the system, not just the data measurements. This allows them to offer greater accuracy for a given model complexity. In addition, dimension reduction using subspace methods is easy and intuitive (i.e. just reduce the number of states).

The strengths and possible weaknesses of the subspace methods are noted below:

Strengths:

- (1) Subspace system identification uses well-understood techniques from linear algebra to provide a reliable and robust numerical method for the linear estimation problem.
- (2) Use of the singular value decomposition to identify fully parameterised state space models, in which the (orthogonal) state sequences are able to capture the important dynamics, leads to a less complex model structure than can be obtained using time-series models.
- (3) Subspace methods identify a state space model structure, the properties of which are well understood from a control point of view are well understood from a control point of view, as evidenced by the wealth of literature available on the subject [14].
- (4) The balanced realisation provided by subspace identification makes the state space model order reduction problem both intuitive and easy to execute.

Weaknesses:

- (1) The computational procedure involved is rather complex which may alienate some users, however automation of the procedure has led to a much more user- friendly procedure.

- (2) The procedure is sensitive to the number of block rows used in the calculations. However this part of the procedure is easily automated, so as to find an appropriate number – at the added computational expense.

4.5 Figures and Tables

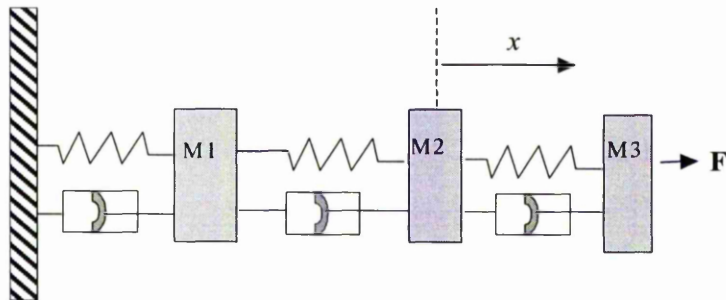


Figure 4.1 Three degrees of freedom mass-spring-damper system.

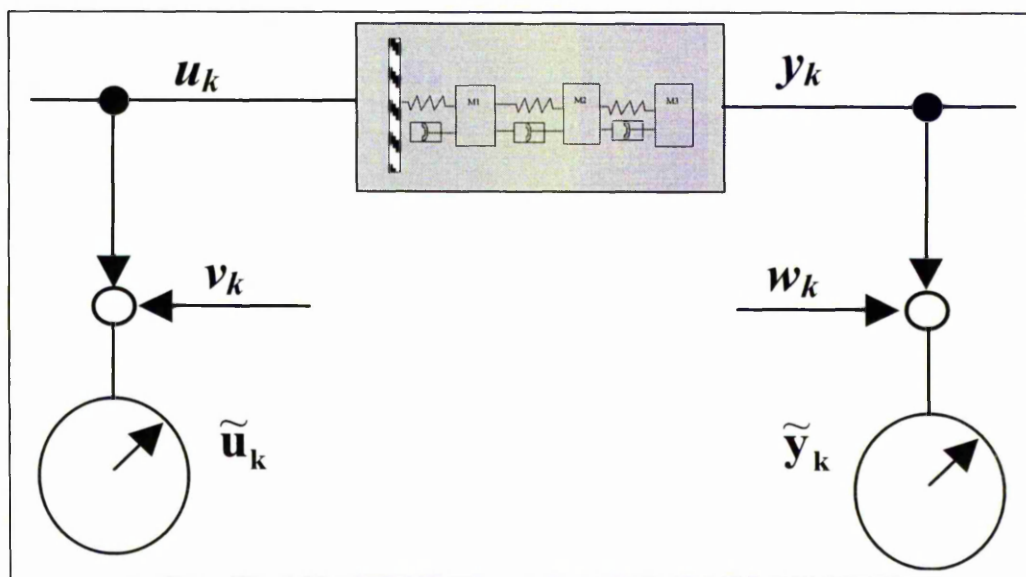


Figure 4.2 The simulated system with noise appended to the system input and the system output.

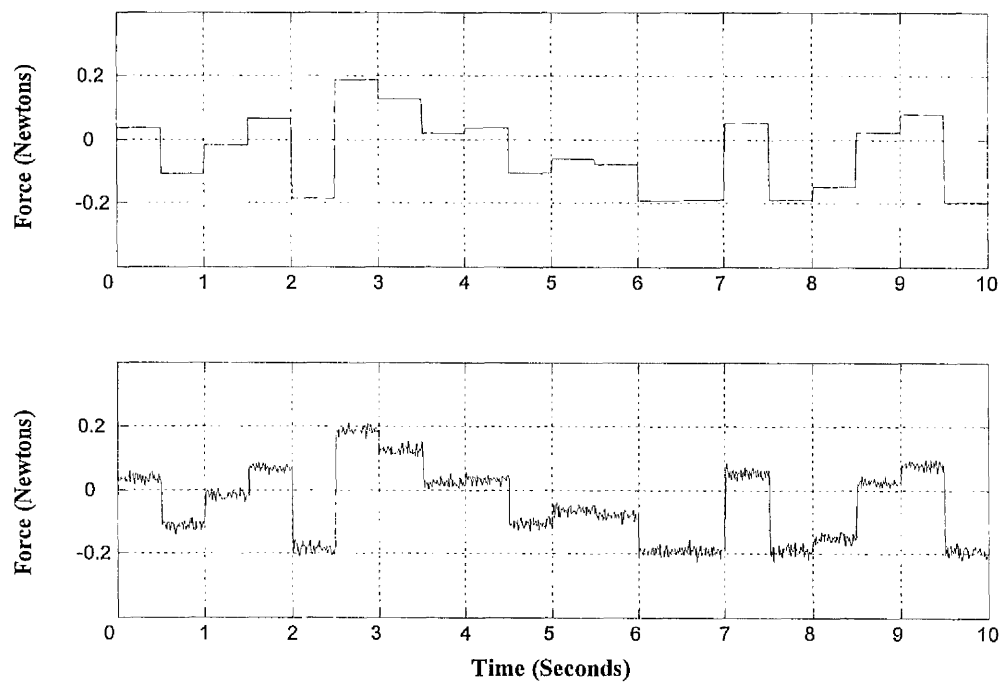


Figure 4.3 Shows the deterministic, uniformly distributed, random input excitation, and the system input with noise added.

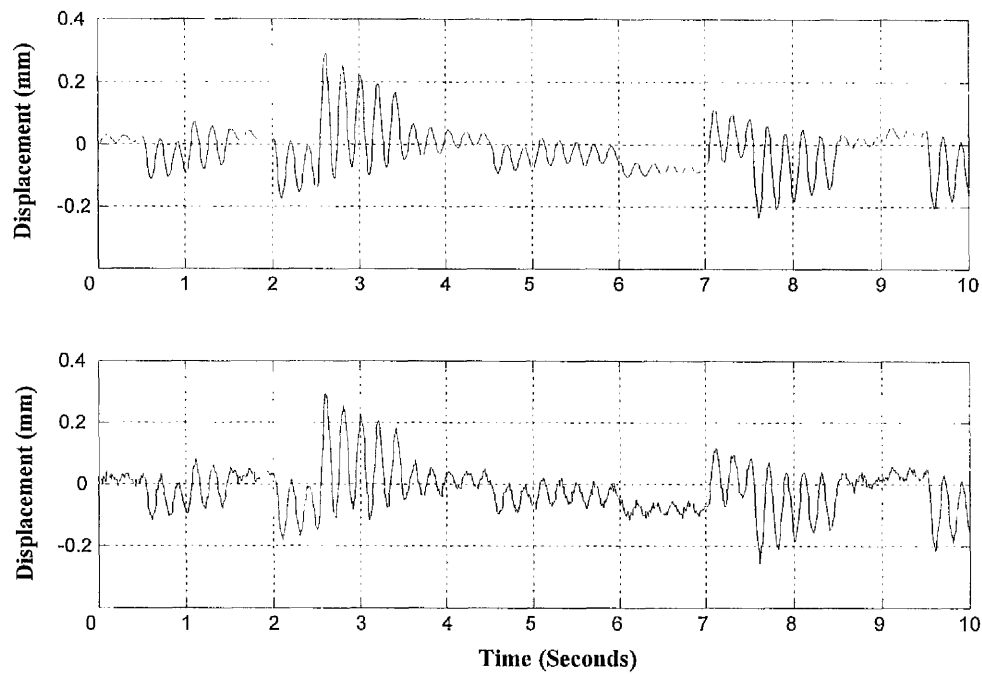


Figure 4.4 Shows the deterministic system output, and the system output with noise added.

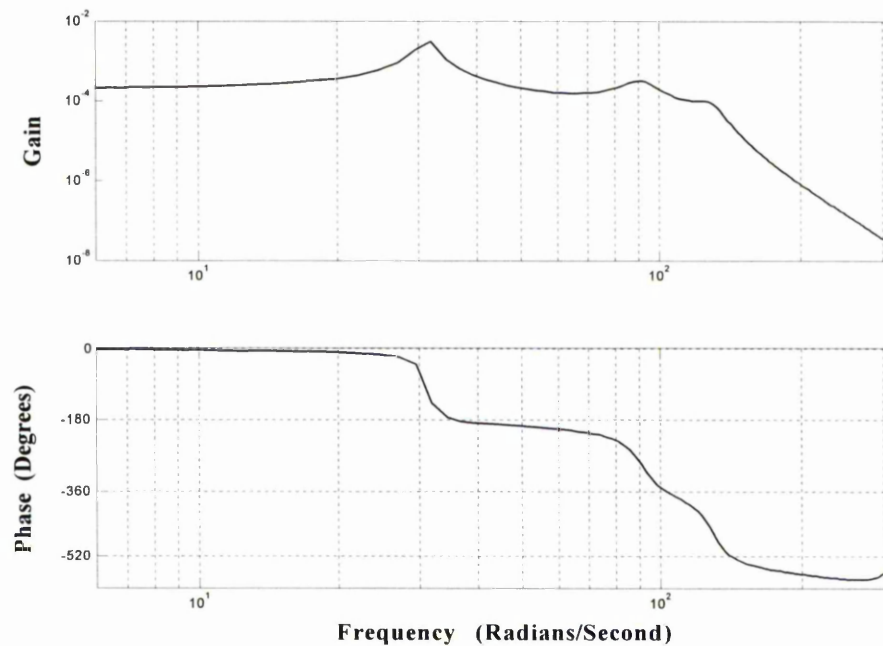


Figure 4.5 Bode plot showing the frequency response of the system from 1-50 Hertz.

n	B	$mse (\times 100)$		n	B	$mse (\times 100)$	
		$M_1(\theta)$	$M_2(\theta)$			$M_1(\theta)$	$M_2(\theta)$
1	0	122.20	123.47	2	0	155.52	107.86
1	1	86.431	83.580	2	1	2.1500	78.158
1	2	109.31	221.42	2	2	4.4107	26.061
1	3	120.00	127.29	2	3	3.0144	62.216
1	4	118.04	99.561	2	4	1.4562	120.84
1	5	110.61	96.861	2	5	1.1241	19.836
1	6	102.18	80.360	2	6	1.0668	1.7534
1	7	93.748	76.047	2	7	1.0467	1.0513
1	8	86.066	74.750	2	8	1.0474	2.3736
1	9	74.832	77.471	2	9	1.0264	4.3772
1	10	76.635	82.337	2	10	1.0267	1.1466
1	11	76.176	87.521	2	11	1.0307	1.1194
1	12	75.472	90.760	2	12	1.0344	1.7639
1	13	74.774	91.749	2	13	1.0467	1.2806
1	14	74.161	89.491	2	14	1.0553	1.3015
1	15	73.702	85.329	2	15	1.0575	1.6762
1	16	73.362	83.656	2	16	1.0348	1.0331
1	17	73.087	79.614	2	17	1.0244	1.0211
1	18	72.747	79.718	2	18	1.0180	1.0204
1	19	72.263	81.631	2	19	1.0164	1.0266
1	20	72.518	84.267	2	20	1.0166	1.1994

Table 4.2 (Continued Over)

n	B	$mse (\times 100)$		n	B	$mse (\times 100)$	
		$M_1(\theta)$	$M_2(\theta)$			$M_1(\theta)$	$M_2(\theta)$
3	0	206.97	56.761	5	11	1.0253	0.9971
3	1	U	U	5	12	1.0213	1.0246
3	2	17.689	5.6206	5	13	1.0150	1.0032
3	3	5.6916	U	5	14	1.0069	1.0036
3	4	1.0729	1.0977	5	15	1.0007	0.9980
3	5	1.0813	1.0850	5	16	0.9993	0.9983
3	6	1.0243	1.0237	5	17	0.9988	0.9972
3	7	1.0389	1.0386	5	18	1.0011	1.0239
3	8	1.0361	1.0118	5	19	1.0034	1.0026
3	9	1.0206	1.0095	5	20	1.0034	1.0145
3	10	1.0174	1.0112	6	0	31.3803	60.6854
3	11	1.0159	1.0181	6	1	U	U
3	12	1.0224	1.0061	6	2	U	U
3	13	1.0295	1.0076	6	3	U	U
3	14	1.0270	1.0351	6	4	U	1.0177
3	15	1.0195	1.0141	6	5	U	1.0264
3	16	1.0196	1.0161	6	6	1.0143	1.0072
3	17	1.0167	1.0085	6	7	1.0131	1.0070
3	18	1.0135	1.0080	6	8	1.0131	1.0211
3	19	1.0111	1.0073	6	9	1.0157	0.9989
3	20	1.0136	1.0349	6	10	28.1260	1.0005
4	0	60.805	42.592	6	11	1.0258	1.0233
4	1	2.4667	2.8705	6	12	1.0140	1.0002
4	2	U	U	6	13	1.0099	1.0117
4	3	1.0305	1.1381	6	14	1.0019	0.9985
4	4	1.0057	1.0787	6	15	0.9992	0.9976
4	5	1.0108	1.0280	6	16	0.9975	0.9946
4	6	1.0082	1.0267	6	17	1.0002	1.0225
4	7	1.0162	1.0109	6	18	1.0043	1.0023
4	8	1.0103	1.0025	6	19	1.0022	1.0141
4	9	1.0093	1.0028	6	20	1.0044	1.0044
4	10	1.0086	1.0145	7	0	21.9229	10.985
4	11	1.0185	0.9976	7	1	U	U
4	12	1.0253	0.9982	7	2	U	U
4	13	1.0244	1.0269	7	3	U	U
4	14	1.0140	1.0051	7	4	U	U
4	15	1.0066	1.0058	7	5	1.0692	1.0096
4	16	1.0007	0.9978	7	6	1.0177	1.0057
4	17	0.9994	0.9983	7	7	1.0156	1.0103
4	18	0.9996	0.9974	7	8	1.0201	1.0018
4	19	1.0019	1.0231	7	9	1.0175	5262.5
5	0	615.541	51.203	7	10	1.0323	1.0161
5	1	1665.9	U	7	11	1.0217	1.0021
5	2	U	U	7	12	1.0128	1.0067
5	3	U	1.1151	7	13	1.0020	0.9981
5	4	U	U	7	14	0.9986	0.9979
5	5	1.0089	1.0299	7	15	0.9981	0.9941
5	6	1.0173	1.0094	7	16	1.0014	1.0208
5	7	1.0102	1.0024	7	17	1.0034	1.0019
5	8	1.0086	1.0029	7	18	1.0055	1.0140
5	9	1.0087	1.0129	7	19	1.0072	1.0043
5	10	1.0159	0.9972	7	20	1.0077	1.0068

Table 4.2 Validation results from **C1**. The shaded regions indicate minimum values of the mean squared prediction error for $n=1:7$, for models M_1 and M_2 . The number of block rows used in the Hankel matrices is $r = n + B$. U indicates that an unstable A matrix was identified.

Model order	$M_1(\theta)$		$M_2(\theta)$	
	<i>mse</i> *100	AIC	<i>mse</i> *100	AIC
1	72.2633	-0.3199	74.7504	-0.2860
2	1.0164	-4.5809	1.0204	-4.5770
3	1.0111	-4.5831	1.0061	-4.58804
4	0.9994	-4.5918	0.9974	-4.5937
5	0.9988	-4.5894	0.9971	-4.5911
6	0.9975	-4.5877	0.9946	-4.5906
7	0.9981	-4.5841	0.9941	-4.5881
8	0.9971	-4.5821	0.9945	-4.5847
9	1.0187	-4.5577	1.0049	-4.5713
10	1.0194	-4.5540	1.0017	-4.5715

Table 4.3 AIC values for M_1 and M_2 .

<i>n</i>	<i>mean mse</i> (x 100)		<i>Std Dev</i> (x 100)		<i>min. mse</i> (x 100)	
	$M_1(\theta)$	$M_2(\theta)$	$M_1(\theta)$	$M_2(\theta)$	$M_1(\theta)$	$M_2(\theta)$
1	71.707	74.358	0.4885	0.5943	70.752	72.973
2	1.0430	1.0454	0.0306	0.0331	0.9634	0.9614
3	1.0361	1.0307	0.0325	0.0314	0.9489	0.9483
4	1.0246	1.0211	0.0321	0.0316	0.9312	0.9314
5	1.0235	1.0211	0.0311	0.0309	0.9333	0.9330
6	1.1293	1.0210	0.7848	0.0316	0.9313	0.9328
7	1.0233	1.0211	0.0335	0.0324	0.9326	0.9315
8	1.0242	1.0210	0.0335	0.0316	0.9311	0.9321
9	1.0261	1.0227	0.0355	0.0333	0.9321	0.9324
10	1.0266	1.0227	0.0330	0.0324	0.9329	0.9332

Table 4.4 Monte Carlo (50), **C1**, the shaded regions indicate the minimum AIC values for M_1 and M_2 .

	<i>mean mse</i> (x 100)		<i>Std Dev</i> (x 100)		<i>min. mse</i> (x 100)	
	$M_1(\theta)$	$M_2(\theta)$	$M_1(\theta)$	$M_2(\theta)$	$M_1(\theta)$	$M_2(\theta)$
1	72.102	74.343	0.4651	0.4790	70.990	73.330
2	2.3390	2.2593	0.4015	0.3375	1.0356	1.0371
3	2.2615	2.2718	0.3268	0.3346	1.0153	1.0141
4	2.2894	2.2703	0.3525	0.3360	1.0143	1.0128
5	2.2907	2.2808	0.3574	0.3503	1.0135	1.0123
6	2.3946	2.3009	0.6831	0.3796	1.0124	1.0102
7	2.3071	2.3190	0.3680	0.3833	1.0110	1.0100
8	2.3051	2.2996	0.3716	0.3698	1.0098	1.0087
9	2.3053	2.2559	0.3600	0.3246	1.0140	1.0111
10	2.3092	2.2371	0.3626	0.3189	1.0136	1.0115

Table 4.5 Monte Carlo (50), **C2**, the shaded regions indicate the minimum AIC values for M_1 and M_2 .

	mean mse (x 100)			Std Dev (x 100)			min. mse (x 100)			AIC		
	$M_3(\theta)$	$M_4(\theta)$	$M_5(\theta)$	$M_3(\theta)$	$M_4(\theta)$	$M_5(\theta)$	$M_3(\theta)$	$M_4(\theta)$	$M_5(\theta)$	$M_3(\theta)$	$M_4(\theta)$	$M_5(\theta)$
2	1.1538	1.1538	1.0370	0.0456	0.0457	0.0278	1.0336	1.0335	1.0172 (12)	-4.541	-4.541	-4.550
3	1.0294	1.0296		0.0320	0.0319		0.9492	0.9492		-4.609	-4.609	
4	1.0193	1.0192		0.0321	0.0316		0.9308	0.9309		-4.613	-4.613	
5	1.0178	1.0178		0.0312	0.0314		0.9311	0.9311		-4.608	-4.610	
6	1.0170	1.0173		0.0316	0.0317		0.9297	0.9303		-4.605	-4.607	

Table 4.6 Monte Carlo (50), **C1**, the shaded regions indicate the minimum AIC values for M_3 , M_4 and M_5 .

	mean mse (x 100)			Std Dev (x 100)			min. mse (x 100)			AIC		
	$M_3(\theta)$	$M_4(\theta)$	$M_5(\theta)$	$M_3(\theta)$	$M_4(\theta)$	$M_5(\theta)$	$M_3(\theta)$	$M_4(\theta)$	$M_5(\theta)$	$M_3(\theta)$	$M_4(\theta)$	$M_5(\theta)$
2	2.4461	2.4449	2.2830	0.2340	0.2313	0.0923	2.1330	2.1349	2.0762 (8)	-4.447	-4.447	-3.849
3	2.3929	2.3943		0.1744	0.1785		2.1877	2.1878		-4.455	-4.455	
4	2.4388	2.4403		0.2340	0.2351		2.1563	2.1577		-4.460	-4.460	
5	2.4251	2.4266		0.2424	0.2415		2.1560	2.1589		-4.457	-4.457	
6	2.4281	2.4334		0.2391	0.2474		2.1531	2.1545		-4.455	-4.454	

Table 4.7 Monte Carlo (50), **C2**, validation data for M_3 , M_4 and M_5 . (ARX Orders 1:61)

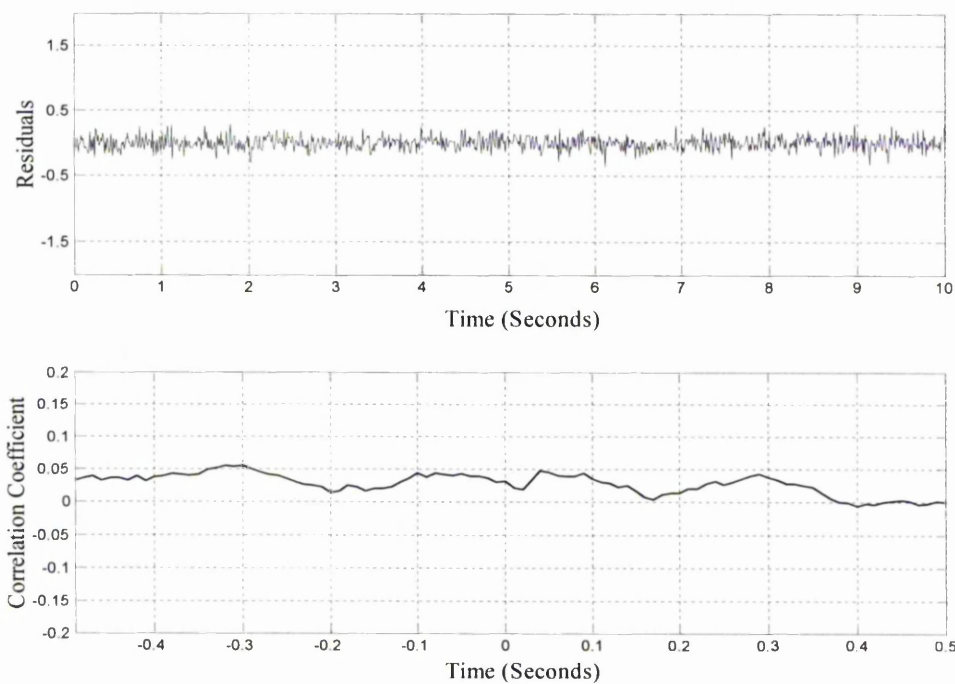


Figure 4.6 Residuals Analysis: 4th Order M_1 , data set **C1**, $mse = 0.0103$

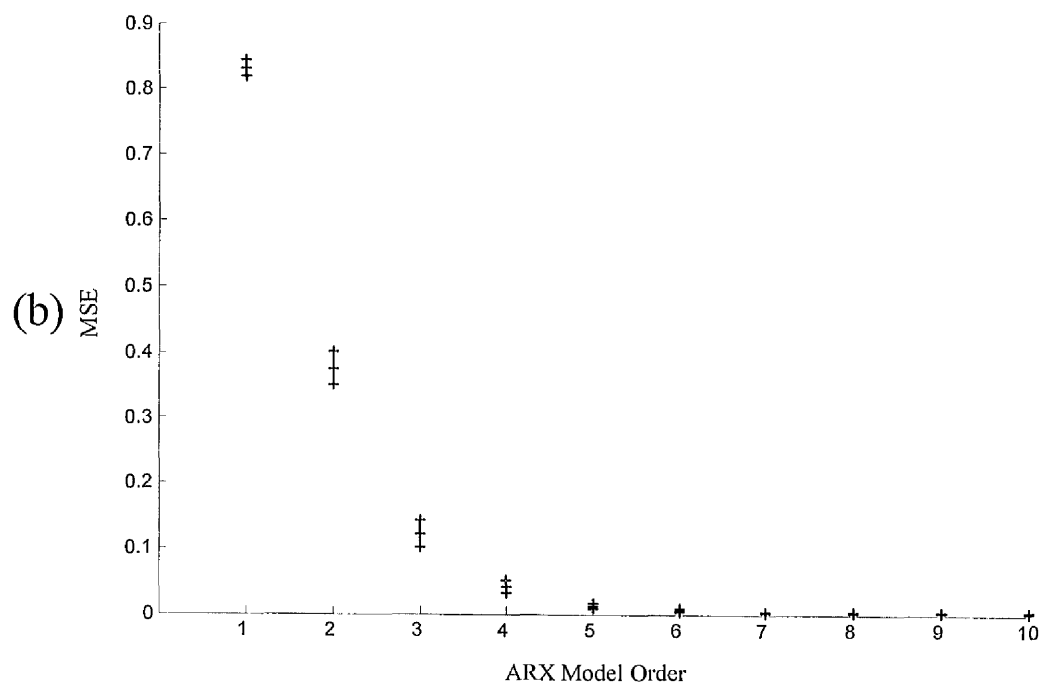
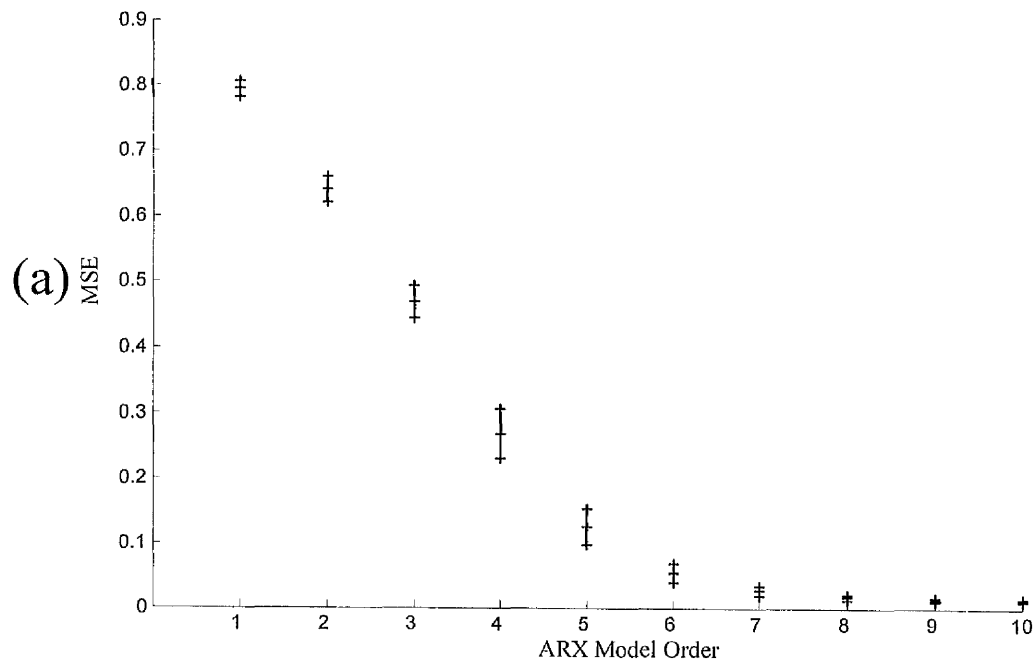


Figure 4.7(a) Results for Monte Carlo (50) **C1**. Mean MSPE ($\mu \pm 3\sigma$) as a function of ARX Model Order.

Figure 4.7(b) Results for Monte Carlo (50) **C5** (coloured noise). Mean MSPE ($\mu \pm 3\sigma$) as a function of ARX Model Order.

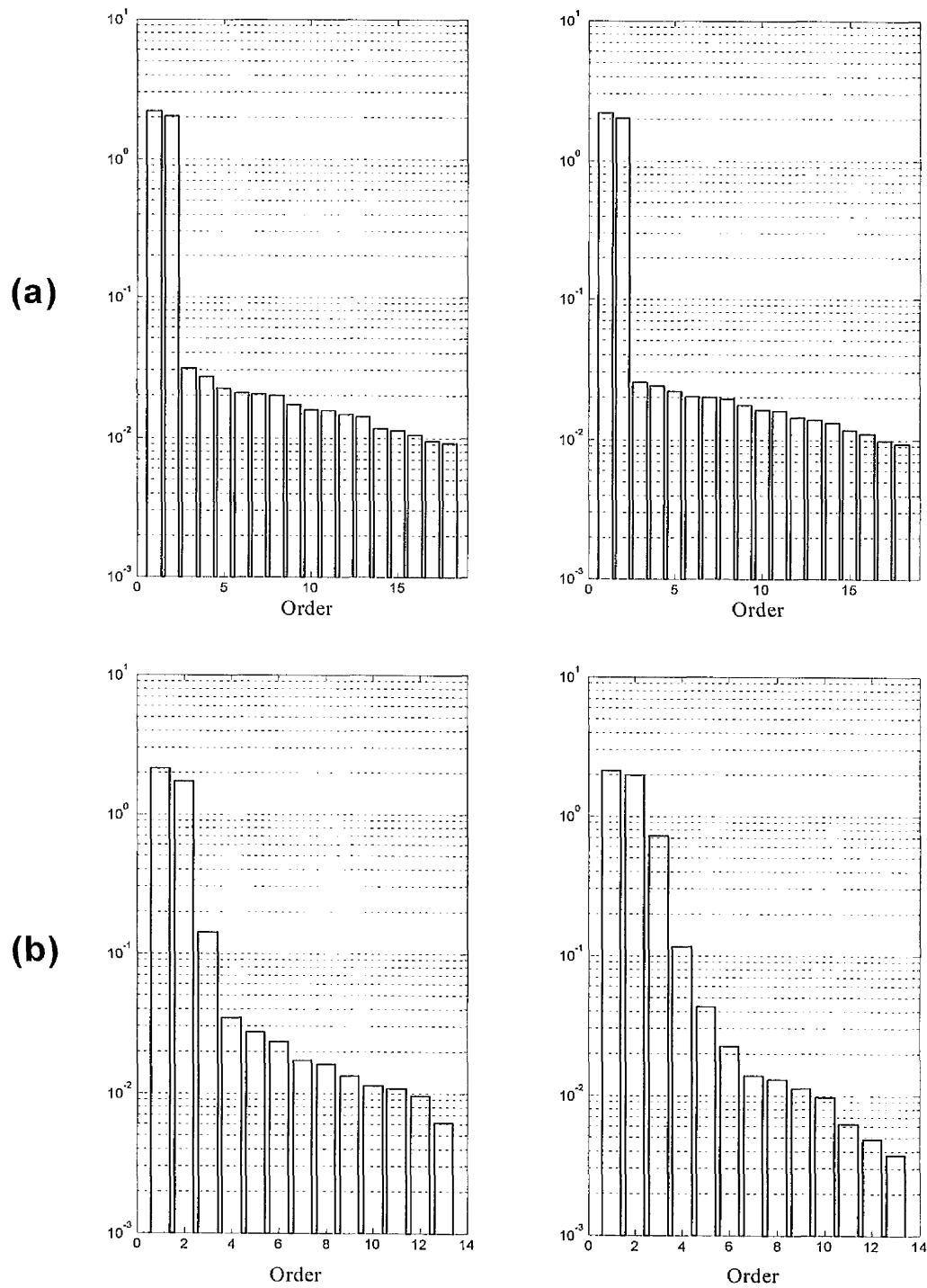


Figure 4.8(a) Singular values of Hankel matrix projection(18 block rows). Top Left: Data taken from **C1**. Top Right: Data taken from **C2**.

Figure 4.8(b) The effect of pure time delays on the singular values. Bottom Left: Data taken from **C1**. Bottom Right: Data taken from **C2**.

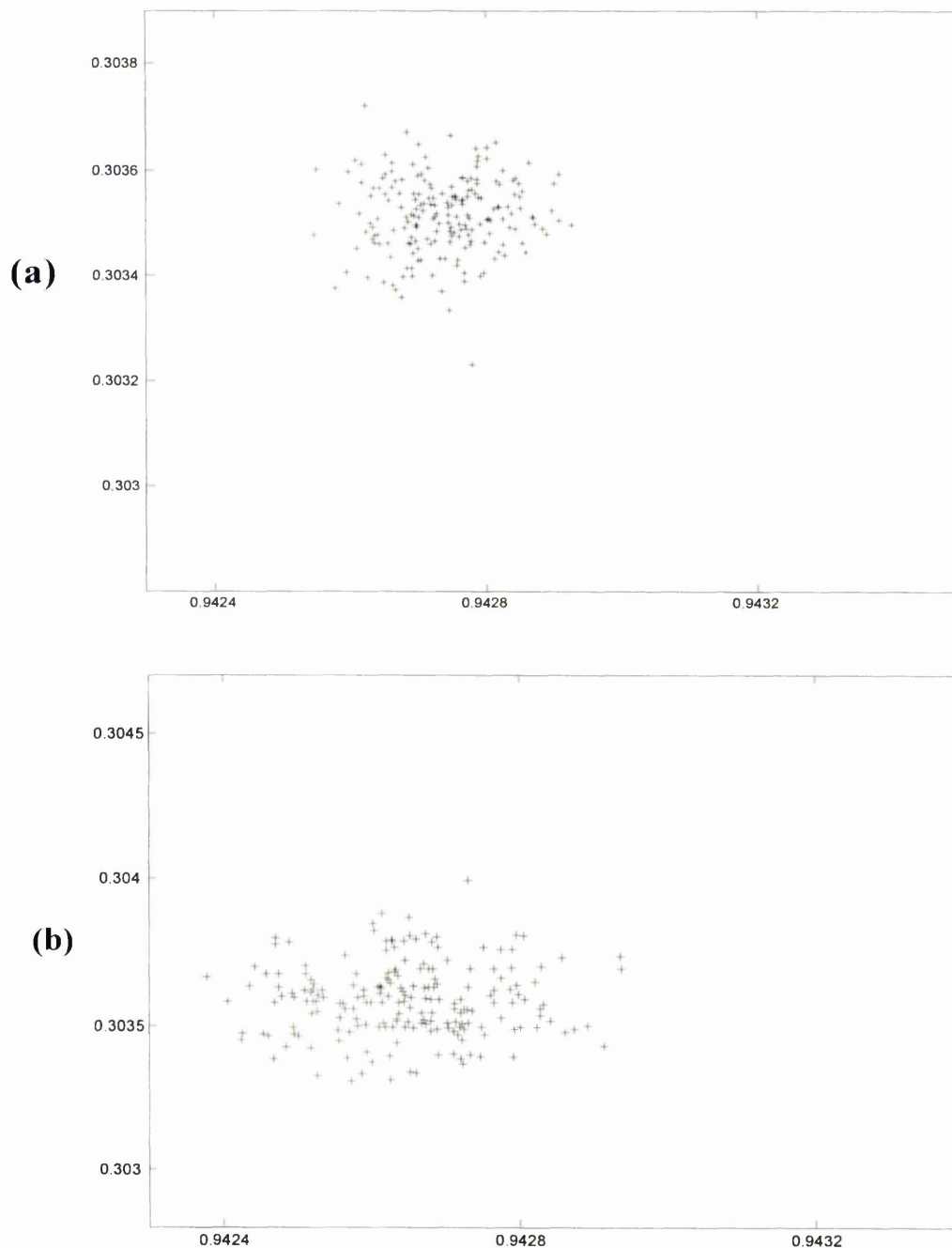
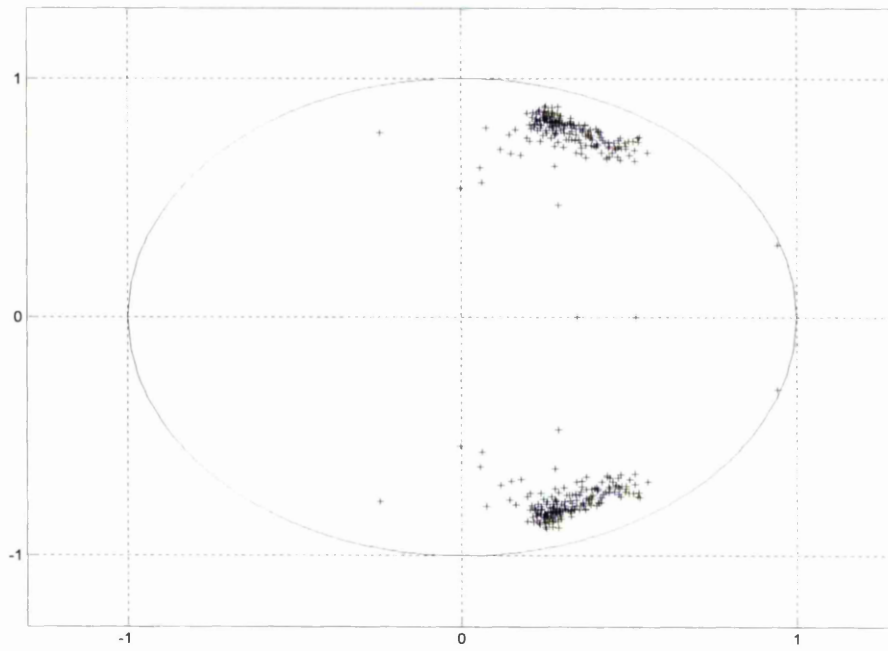


Figure 4.9(a) Monte Carlo (200) **C1**. Eigenvalue plot of the dynamical A matrix of 2nd order state space models identified using algorithm \mathcal{M}_1 ($r = 18$). All 200 points are centered in the regions $0.94 \pm 0.3i$.

Figure 4.9(b) Monte Carlo (200) **C5** (coloured noise). Eigenvalue plot of the dynamical A matrix of 2nd order state space models identified using algorithm \mathcal{M}_1 ($r = 18$). The spread of the estimates has increased slightly.

(a)



(b)

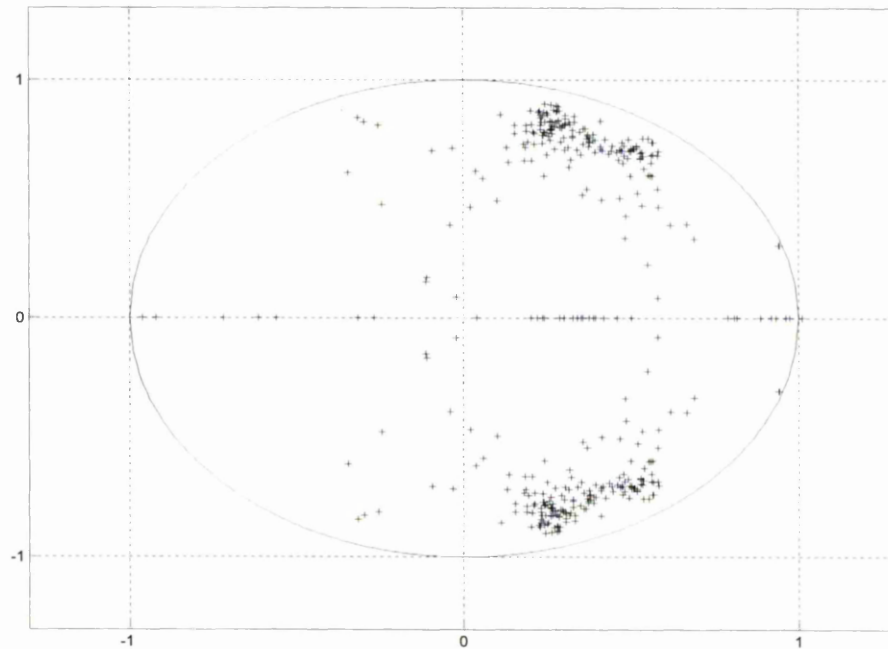
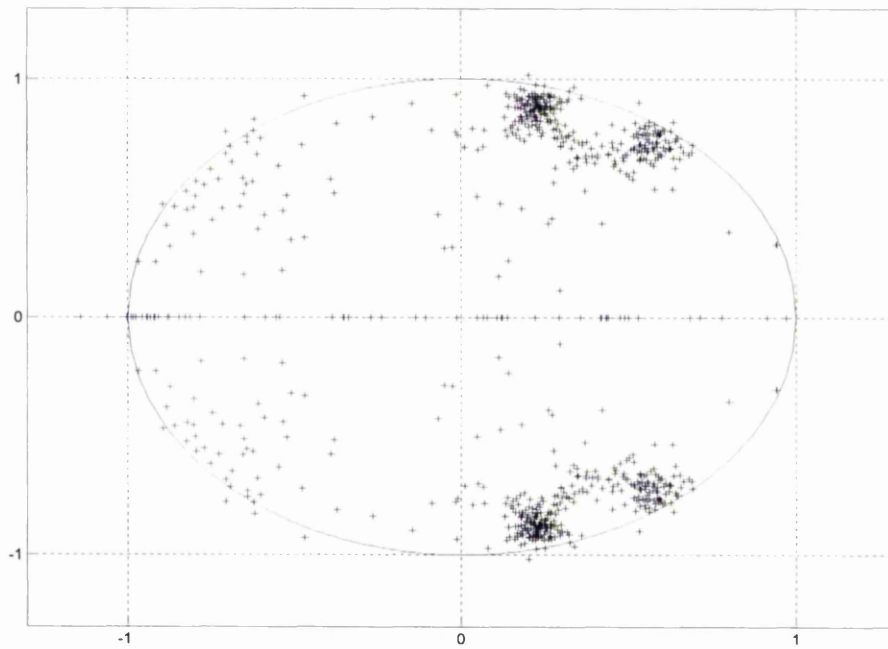


Figure 4.10(a) Monte Carlo (200) **C1** (White noise appended to the system outputs). Eigenvalue plot of the dynamical **A** matrix of 4th order state space models identified using algorithm \mathbf{M}_1 ($r = 18$). The 200 eigenvalues that correspond to the first mode of vibration are centred on $0.94 \pm 0.3i$.

Figure 4.10(b) Monte Carlo (200) **C2** (White noise appended to the system inputs and system outputs). Eigenvalue plot of the dynamical **A** matrix of 4th order state space models identified using algorithm \mathbf{M}_1 ($r = 18$).

(a)



(b)

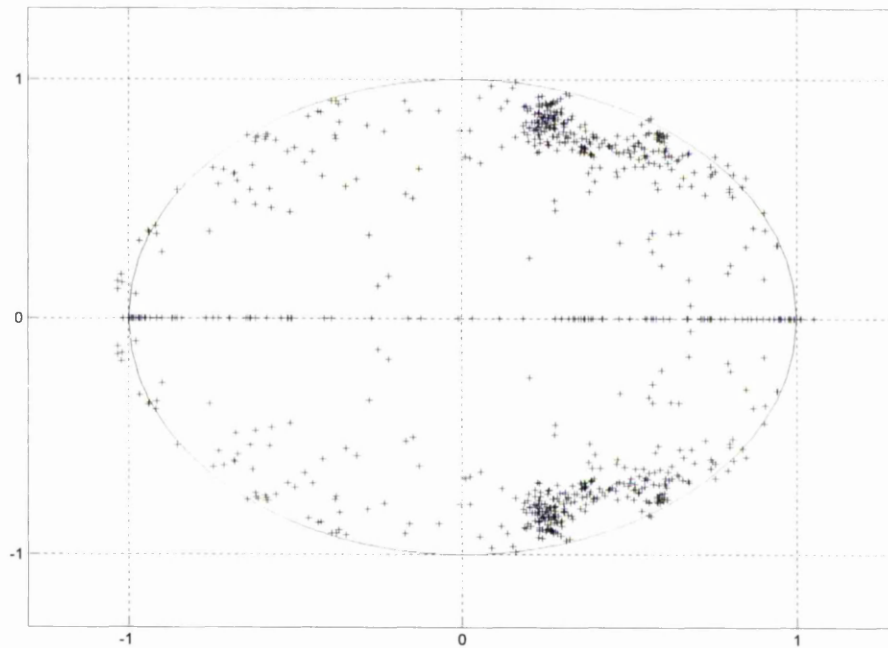
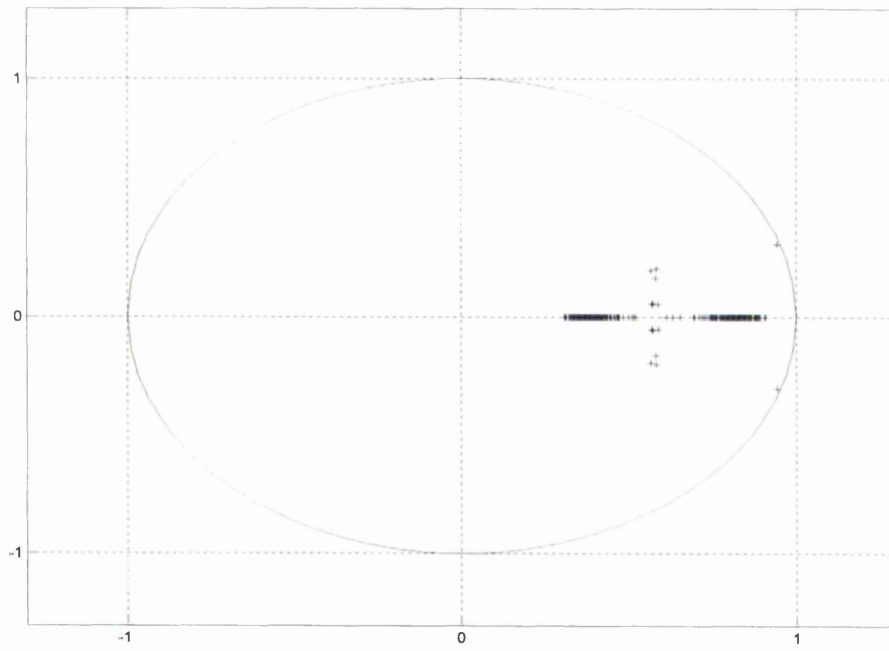


Figure 4.11(a) Monte Carlo (200) **C1** (White noise appended to the system outputs). Eigenvalue plot of the dynamical **A** matrix of 6th order state space models identified using algorithm \mathbf{M}_1 ($r = 18$).

Figure 4.11(b) Monte Carlo (200) **C2** (White noise appended to the system inputs and system outputs). Eigenvalue plot of the dynamical **A** matrix of 6th order state space models identified using algorithm \mathbf{M}_1 ($r = 18$).

(a)



(b)

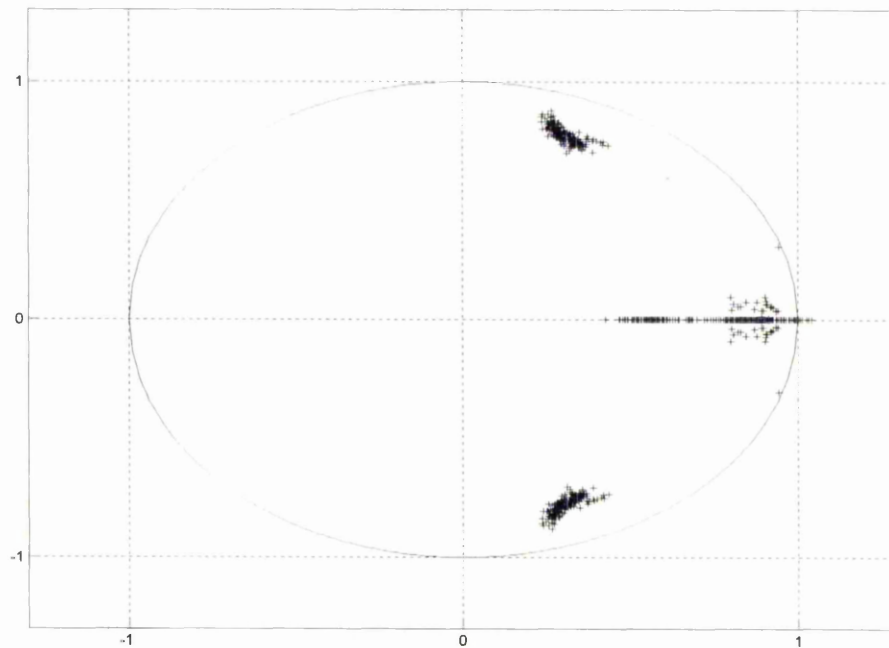


Figure 4.12(a) Monte Carlo (200) **C5** (Coloured noise appended to the system outputs). Eigenvalue plot of the dynamical **A** matrix of 4th order state space models identified using algorithm \mathbf{M}_1 ($r = 18$). The 200 eigenvalues that correspond to the first mode of vibration are centred on $0.94 \pm 0.3i$.

Figure 4.12(b) Monte Carlo (200) **C5** (Coloured noise appended to the system outputs). Eigenvalue plot of the dynamical **A** matrix of 6th order state space models identified using algorithm \mathbf{M}_1 ($r = 18$).

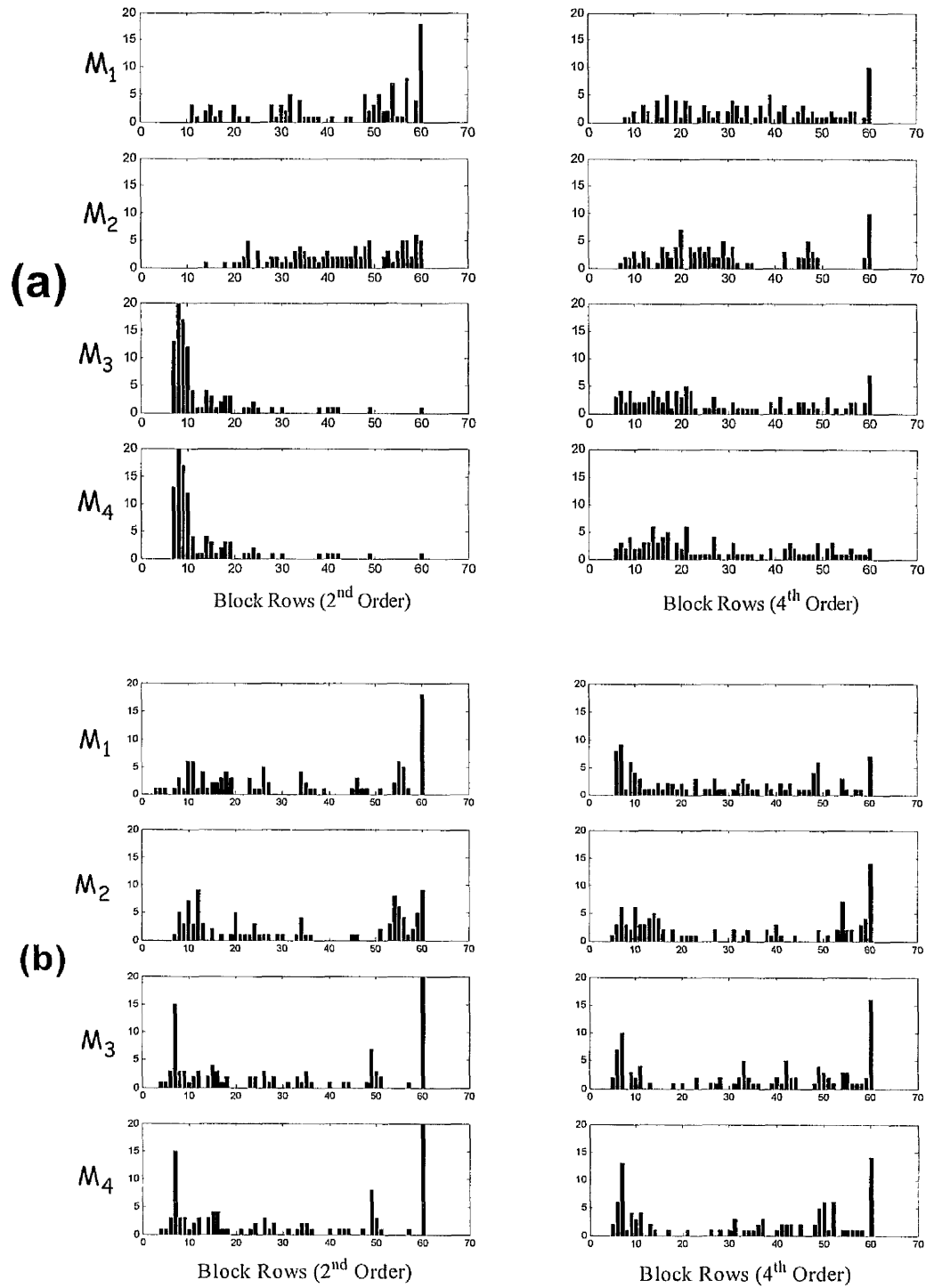


Figure 4.13 Frequency histograms showing r_{opt} for algorithms M_1 , M_2 , M_3 and M_4 , based on Monte Carlo (100) **C1** (Fig. 4.13a) and Monte Carlo (100) **C2** (Fig. 4.13b). The left column shows the results for 2nd order models. The right columns show the results for 4th order models.

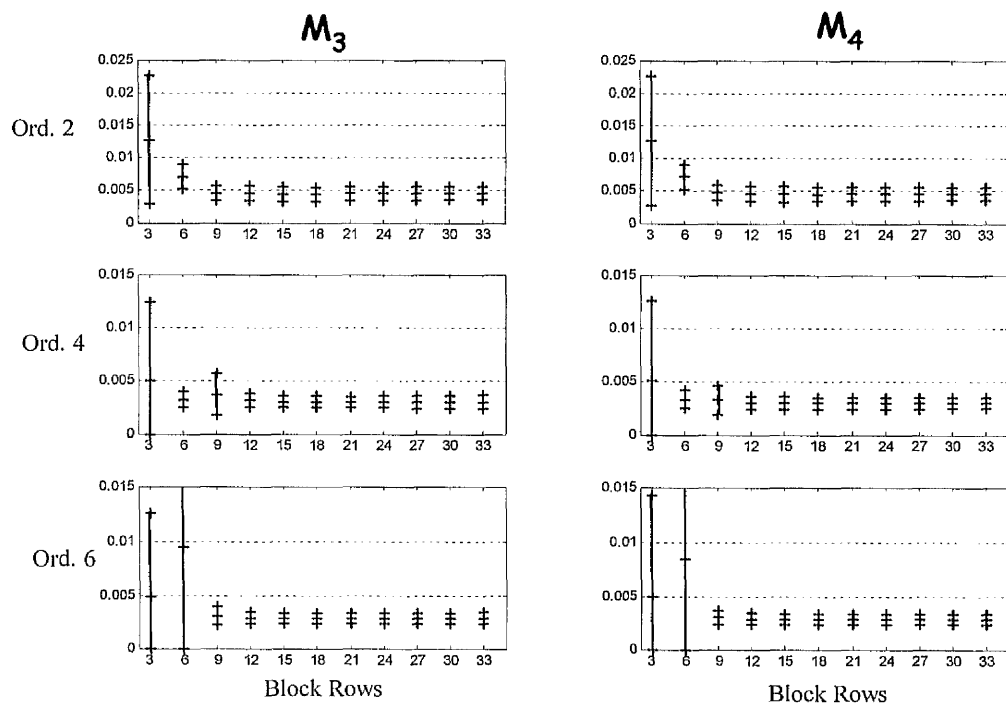


Figure 4.14 The effect of increasing the number of block rows of input-output data on the prediction accuracy of the resulting state space models (identified using algorithms M_3 and M_4). Monte Carlo (200) experiment using dataset **C5**. Mean MSPE ($\mu \pm 3\sigma$) as a function of r .

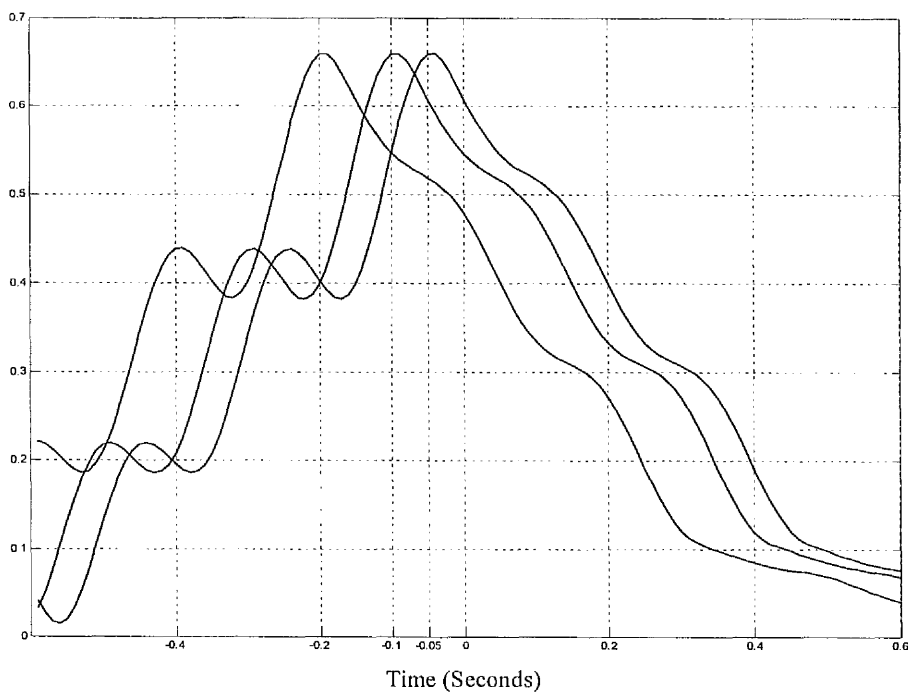


Figure 4.15 Cross-correlation between output and input for data sets **C1**, **C4** and **C5**.

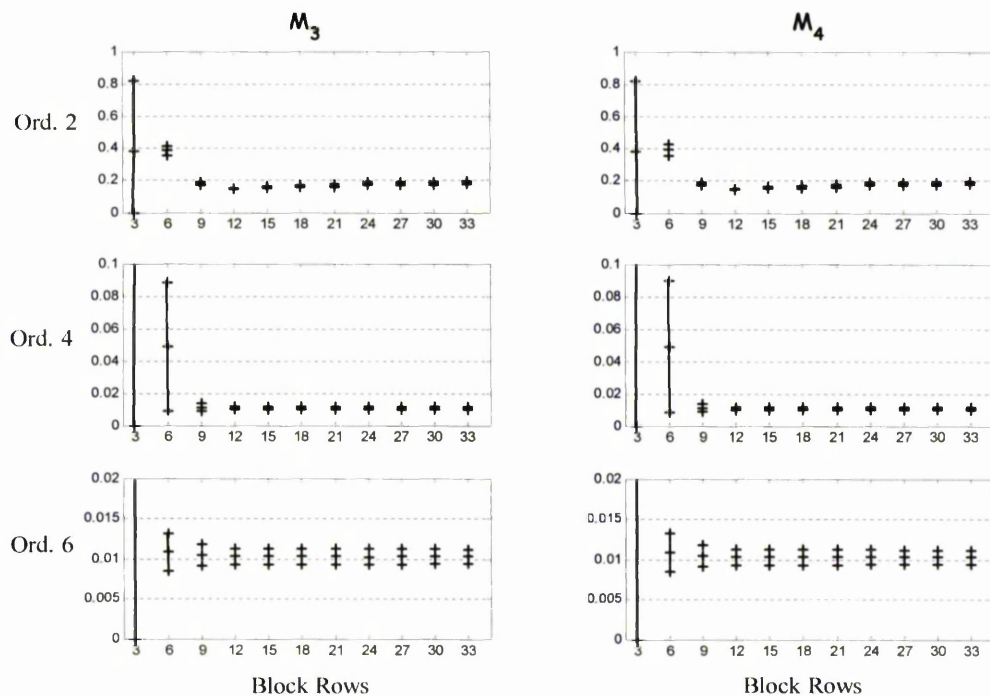


Figure 4.16 Shows the effect of a measurement time delay on the prediction accuracy of algorithms M_3 and M_4 . 50 Monte Carlo experiments were performed using dataset **C3**. Mean MSPE ($\mu \pm 3\sigma$) is plotted as a function of r .

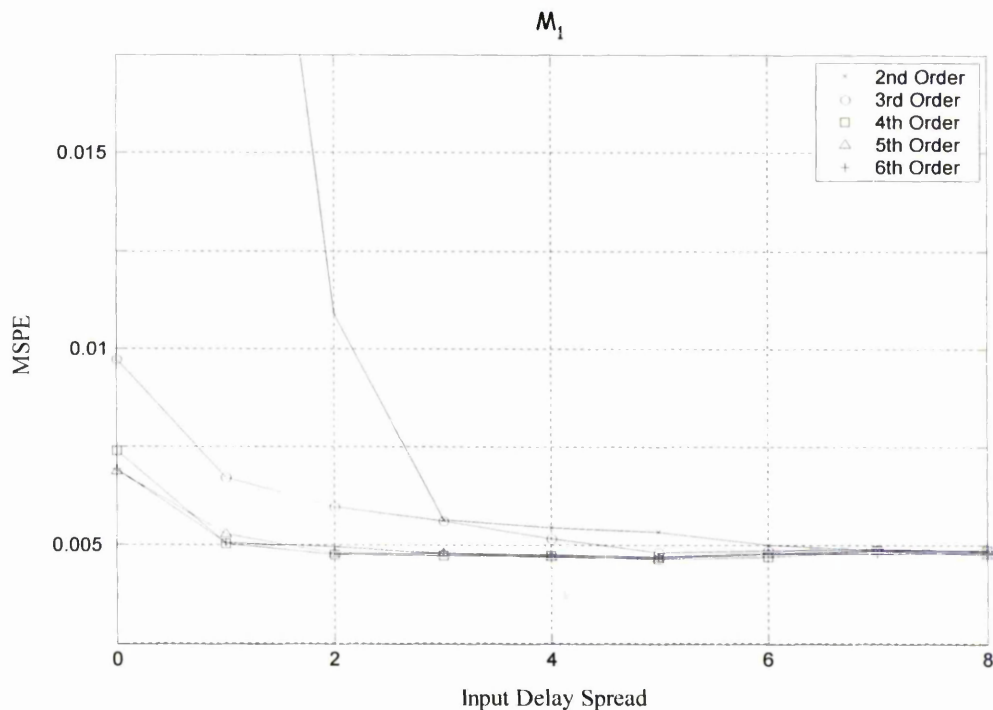


Figure 4.17 Shows the effect of including time-lagged inputs in the state space model structure. MSPE is plotted as a function of the size of the delay spread. Algorithm M_1 was applied to data taken from **C3** (measurement delay of 5 samples) to produce state space models with orders $n = 2 : 6$.

Figure 4.18(a) State Equation for M_1 2nd Order

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0.94 & 0.34 \\ -0.27 & 0.95 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} -31.9 & 0 & 0 & 0 & 0 & 0 & -6.63 \\ 1.92 & 0 & 0 & 0 & 0 & 0 & 29.0 \end{pmatrix} \begin{pmatrix} u_k \\ u_{k-1} \\ u_{k-2} \\ u_{k-3} \\ u_{k-4} \\ u_{k-5} \\ u_{k-6} \end{pmatrix}$$

Figure 4.18(b) State Equation for M_1 6th Order

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} = \begin{pmatrix} 0.95 & 0.32 & 0.00 & 0.00 & 0.00 & -0.00 \\ -0.28 & 0.94 & -0.02 & 0.01 & -0.01 & -0.00 \\ -0.01 & 0.01 & 0.11 & -0.89 & -0.12 & 0.04 \\ -0.00 & 0.00 & 0.89 & 0.25 & -0.24 & -0.09 \\ -0.00 & 0.00 & -0.14 & 0.11 & -0.80 & -0.39 \\ -0.00 & 0.00 & -0.03 & 0.11 & 0.36 & -0.54 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} + \begin{pmatrix} -33.6 & 0 & 0 & 0 & 0 & 0 & 4.39 \\ 14.2 & 0 & 0 & 0 & 0 & 0 & 35.5 \\ -0.35 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.21 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.10 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.06 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} u_k \\ u_{k-1} \\ u_{k-2} \\ u_{k-3} \\ u_{k-4} \\ u_{k-5} \\ u_{k-6} \end{pmatrix}$$

Figure 4.18(c) State Equation for M_2 2nd Order

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0.89 & 0.29 \\ -0.32 & 0.99 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + 0.01 * \begin{pmatrix} -19.9 & 0.60 & -0.30 & 0.33 & -0.02 & -0.15 & -6.63 \\ -33.0 & -0.03 & 0.01 & -0.04 & -0.00 & 0.00 & 29.0 \end{pmatrix} \begin{pmatrix} u_k \\ u_{k-1} \\ u_{k-2} \\ u_{k-3} \\ u_{k-4} \\ u_{k-5} \\ u_{k-6} \end{pmatrix}$$

Figure 4.18(d) State Equation for M_2 6th Order

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} = \begin{pmatrix} 0.91 & 0.26 & 0.02 & 0.01 & -0.00 & 0.00 \\ -0.36 & 0.97 & -0.00 & -0.01 & 0.01 & -0.00 \\ -0.13 & -0.02 & 0.00 & 0.44 & 0.12 & -0.05 \\ -0.14 & 0.02 & -1.18 & 0.26 & -0.47 & -0.16 \\ 0.09 & -0.01 & 0.66 & 0.43 & -0.61 & -0.12 \\ 0.00 & 0.00 & -0.01 & 0.29 & 0.45 & -0.52 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} + 0.01 * \begin{pmatrix} -15.1 & 0.53 & -0.26 & -0.26 & -0.09 & -0.14 & 0.10 \\ -38.8 & -0.18 & 0.09 & -0.09 & 0.04 & 0.06 & -0.05 \\ -24.4 & 0.03 & -0.02 & 0.10 & -0.20 & -0.33 & -0.40 \\ -10.5 & 0.89 & -0.07 & 0.37 & 1.42 & -0.13 & -0.79 \\ 2.42 & 0.87 & 0.22 & -0.54 & 0.82 & -0.62 & -0.22 \\ -13.6 & -1.74 & 0.47 & 0.13 & 0.12 & 1.31 & -0.57 \end{pmatrix} \begin{pmatrix} u_k \\ u_{k-1} \\ u_{k-2} \\ u_{k-3} \\ u_{k-4} \\ u_{k-5} \\ u_{k-6} \end{pmatrix}$$

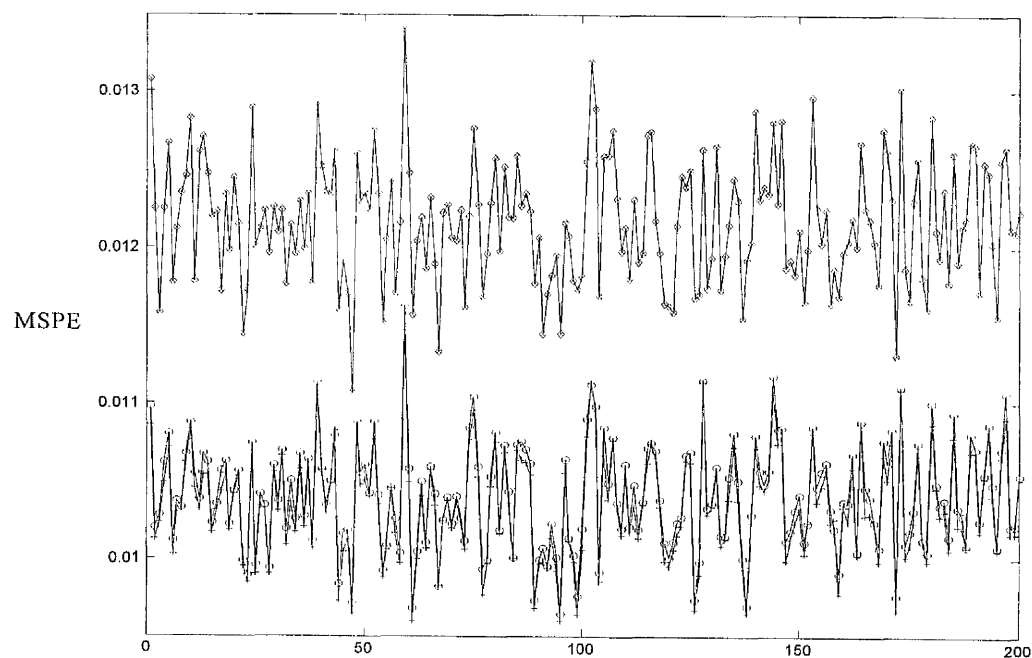


Figure 4.19 Validation Experiment, Monte Carlo (200) **C1**. MSPE for 2nd Order M_1 (+); M_2 (o); M_3 (x); M_4 (\diamond).

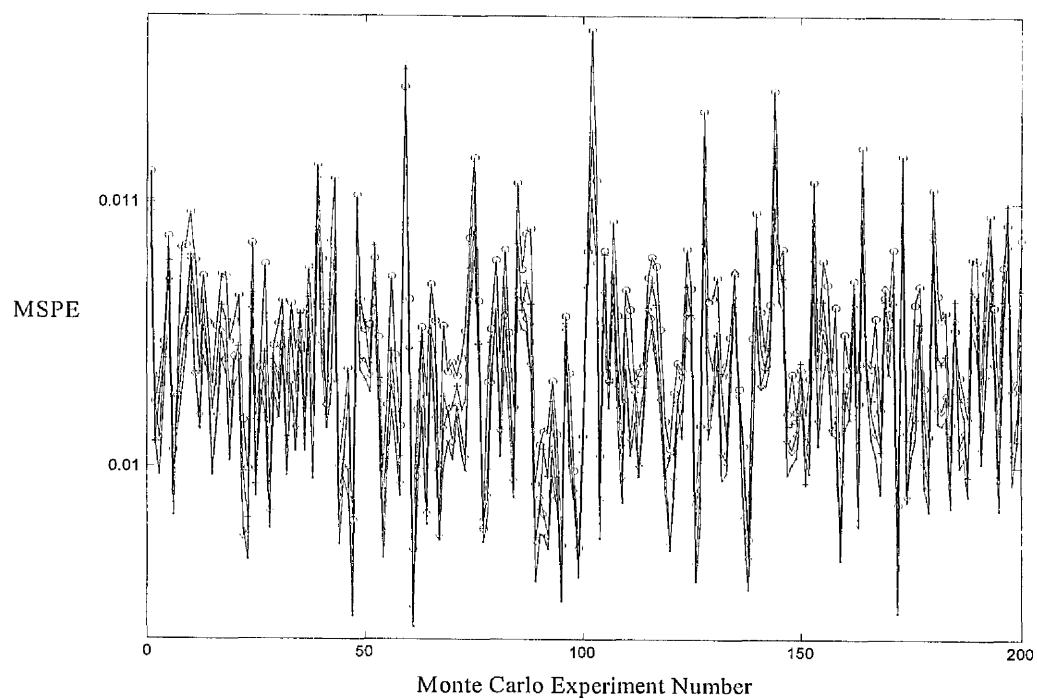


Figure 4.20 Validation Experiment, Monte Carlo (200) **C1**. MSPE for 4th Order M_1 (+); M_2 (o); M_3 (x); M_4 (\diamond).

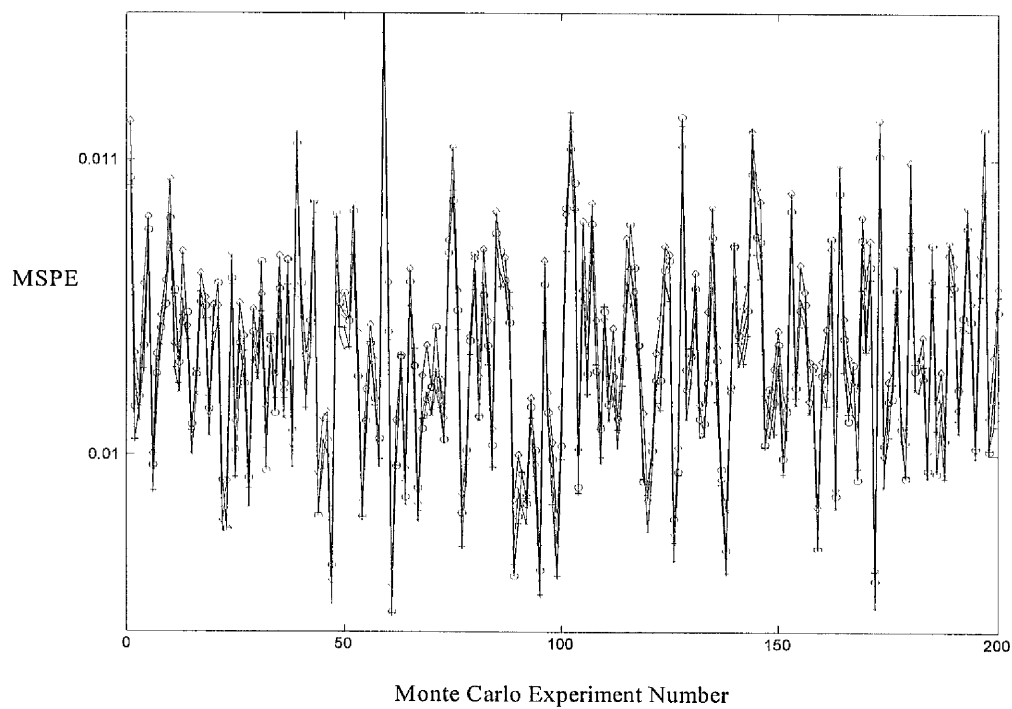


Figure 4.21 Validation Experiment, Monte Carlo (200) **C1**. MSPE for 6th Order M_1 (+); M_2 (o); M_3 (x); M_4 (\diamond).

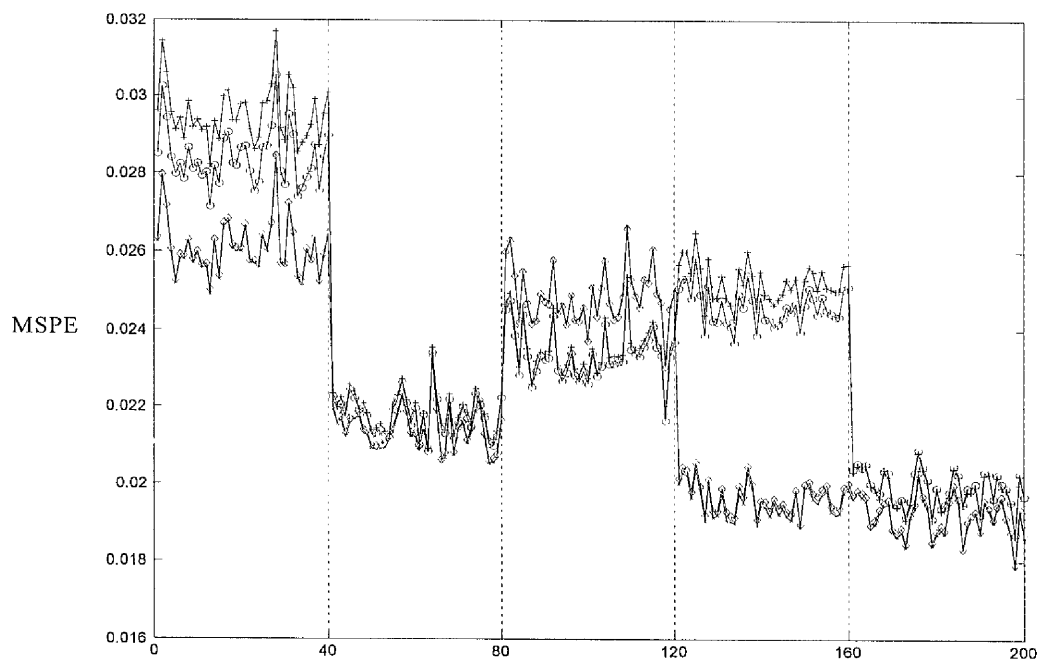


Figure 4.22 Validation Experiment, Monte Carlo (200) **C2**. MSPE for 2nd Order M_1 (+); M_2 (o); M_3 (x); M_4 (\diamond).

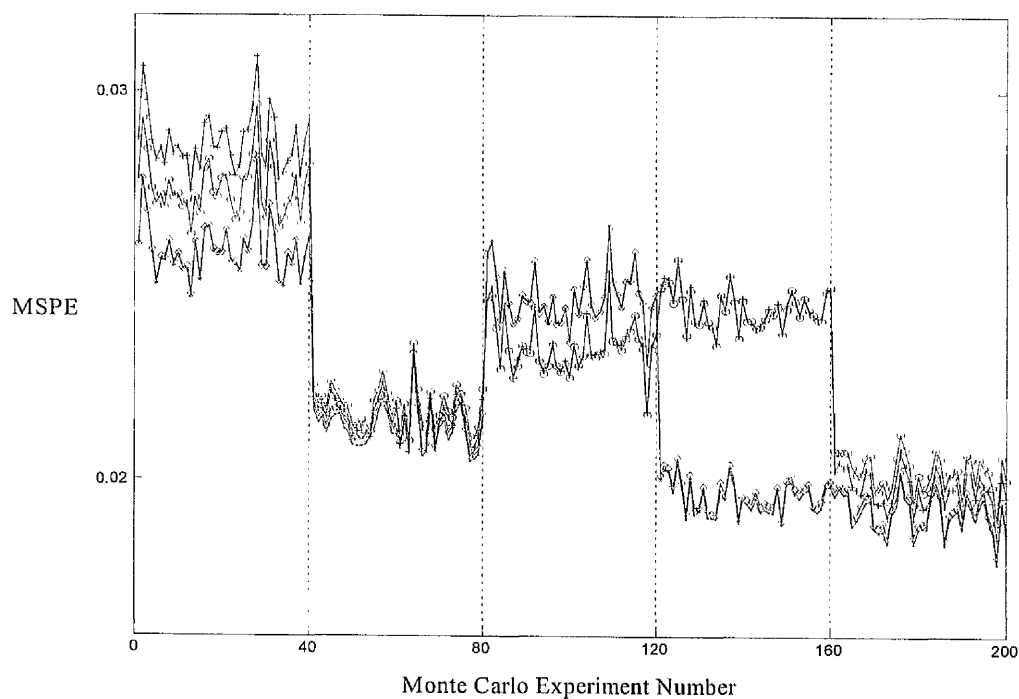


Figure 4.23 Validation Experiment, Monte Carlo (200) **C2**. MSPE for 4th Order M_1 (+); M_2 (o); M_3 (x); M_4 (\diamond).

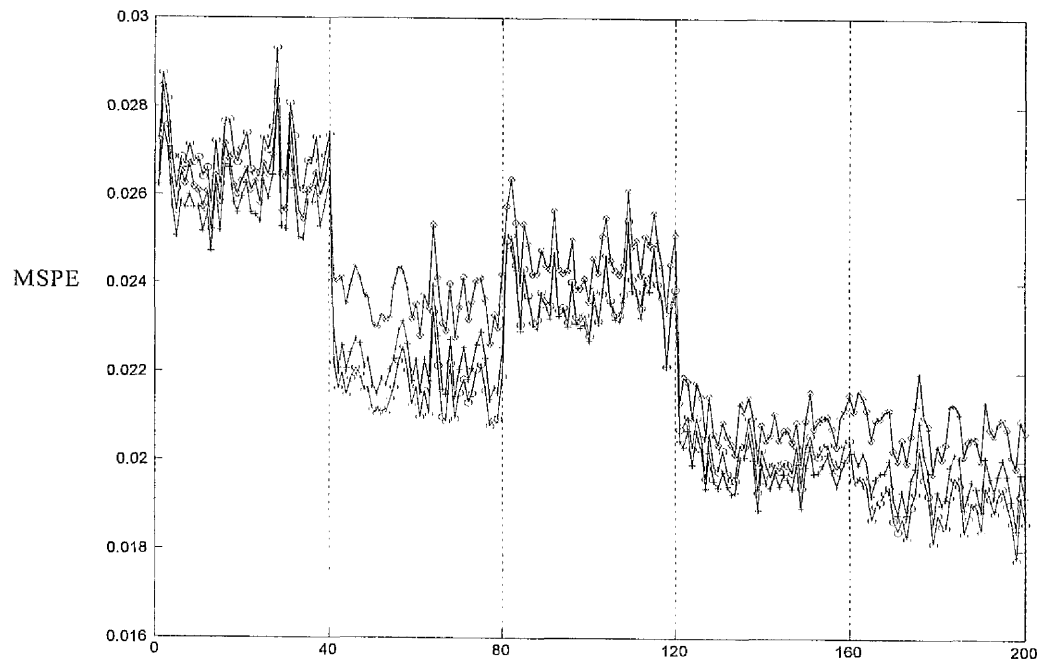


Figure 4.24 Validation Experiment, Monte Carlo (200) **C2**. MSPE for 6th Order M_1 (+); M_2 (o); M_3 (x); M_4 (\diamond).

Chapter 5

Identification of Industrial Plant

A simulation of complex industrial plant is used to demonstrate subspace system identification. The performance of subspace algorithms is compared with ARX and FIR models. User choices associated with the subspace algorithms and aspects of model order reduction are also considered.

5.1 Introduction

In this chapter, state space models obtained using subspace system identification are compared to ARX and FIR models of a complex industrial system.

Linear models of complex industrial processes are widely used, for example on distillation processes in the petrochemical industry, and on drying and evaporation processes in the food and beverage industry. It is expected that the performance of automated control software packages can be improved through the incorporation of improved system identification methods. An improved model of the process will help a model predictive controller to find a better control solution, and aid in the reduction of plant operating costs.

Considerable modelling challenges present themselves in the form of the fluid catalytic cracking units (FCCU) of petrochemical plant. These processes are highly interacting, multiple input multiple output (MIMO) systems. As shown in Figure 5.1, FCC units produce gasoline from low market value feed stocks, making them important to the overall economic performance of the oil refinery, and prime candidates for advanced control strategies such as model predictive control [70].

The size of the models, usually related to the number of process input and output variables, and the number of model parameters, has a significant influence upon the computational effort required to find an optimal control solution. Since the number of process variables is specific to the application, it is desired to keep the number of model parameters as small as possible. This generally contradicts with the requirement for providing a model that predicts the process output variables with sufficient accuracy, i.e. the fewer model parameters the less accurate the model predictions. It is therefore required to select the model structure and the number of parameters carefully.

The models under consideration in this chapter incorporate state space, ARX and FIR model structures. As described in previous chapters, the discrete, linear state space model is of the form

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{v}_k \\ \mathbf{y}_k &= \mathbf{C}\mathbf{x}_k + \mathbf{D}\mathbf{u}_k + \mathbf{w}_k \end{aligned} \quad (5.1)$$

where $\mathbf{A} \in \mathbb{R}^{r \times r}$, $\mathbf{B} \in \mathbb{R}^{r \times m}$, $\mathbf{C} \in \mathbb{R}^{l \times r}$, $\mathbf{D} \in \mathbb{R}^{l \times m}$ are the state-space matrices and $\mathbf{x}_k \in \mathbb{R}^r$, $\mathbf{u}_k \in \mathbb{R}^m$, $\mathbf{y}_k \in \mathbb{R}^l$ are the states, the input and the output vector at time instance k , respectively, \mathbf{v}_k and \mathbf{w}_k are white noise sequences, and the number of state, input and output variables are r , m , and l respectively. Usually, only the input and the output variables are measured, and hence, the state variables need to be estimated.

The ARX model structure is described as an equation error model structure [6]. A 2^{nd} order ARX model; for predicting the k^{th} instance of the j^{th} output prediction variable, $\hat{y}_k^{(j)}$, is given by

$$\hat{y}_k^{(j)} = a_1^{(j)} \hat{y}_{k-1}^{(j)} + a_2^{(j)} \hat{y}_{k-2}^{(j)} + \sum_{i=1}^N b_1^{(ji)} u_{k-1}^{(i)} + b_2^{(ji)} u_{k-2}^{(i)}, \quad (5.2)$$

where $\hat{y}_k^{(j)}$, $\hat{y}_{k-1}^{(j)}$ and $\hat{y}_{k-2}^{(j)}$ are the predicted values of the j^{th} output prediction variable at time instances k , $k-1$, and $k-2$, respectively. The model coefficients $a_1^{(j)}$ and $a_2^{(j)}$ are for the j^{th} output prediction variable, and the model coefficients for the i^{th} manipulated variable are $b_1^{(ij)}$ and $b_2^{(ij)}$. The $(k-1)^{th}$ and the $(k-2)^{th}$ instances of the i^{th} manipulated variable are $u_{k-1}^{(i)}$ and $u_{k-2}^{(i)}$, respectively, and N is the number of manipulated variables.

The Finite Impulse Response (FIR) model is described as an output error model structure [6], and may be regarded as a special case of the ARX model, but with the autoregressive terms omitted. Utilising a FIR model structure, the j^{th} output variable, $\hat{y}_k^{(j)}$ is predicted as

$$\hat{y}_k^{(j)} = \sum_{i=1}^n \sum_{m=1}^N a_i^{(jm)} u_{k-i}^{(m)}, \quad (5.3)$$

where $u_{k-i}^{(m)}$ is the value of the m^{th} input variable at time $k-i$ and n is the number of time lagged input values, and N is the number of manipulated variables.

In the FCCU study that follows, two subspace algorithms are compared with linear difference equation model structures (ARX and FIR). The FCCU study uses two data sets, **C6** and **C7**, obtained from simulation study. The comparison study will be used to demonstrate the effect of user choices on the accuracy of subspace system identification. There will also be consideration of the number of parameters required for each model structure, and advantages and disadvantages associated with subspace system identification will be considered.

5.2 The FCCU Simulation

Petrochemical refineries are used for the conversion of crude oil into a range of refined petroleum products: gasoline, diesel fuels, heating oils, aviation fuel oils, asphalt and feedstock for lubricants and petrochemicals. Conversion taking place in FCC units plays an important role in ensuring profitability in refineries and meeting market requirements [136]. The FCCU typically receives several different heavy feedstocks from other refinery units and cracks these streams to produce lighter, more valuable components,

that are eventually blended into gasoline and other products. A schematic of the FCCU simulation is presented in Figure 5.2. The principal feed to the unit is gas oil, but heavier diesel and wash oil streams also contribute to the total feed stream. Fresh feed is preheated and then passed to the riser, where it is mixed with hot, regenerated catalyst from the regenerator. The hot catalyst provides the heat necessary for the endothermic cracking reactions. The gaseous cracked products are passed to the main fractionator for separation. Wet gas at the top of the main fractionator is pressurised for downstream separation by the wet gas compressor. Separation of light components occurs in this downstream separation section.

As a result of the cracking process, coke is deposited on the surface of the catalyst, which rapidly lowers its activity. Spent catalyst is recycled to the regenerator where it is mixed with air in a fluidised bed for regeneration of its catalytic properties. The deposited coke is burnt off to produce carbon monoxide and carbon dioxide. Air is pumped to the regenerator by a high-capacity combustion air blower and a smaller lift air blower. In addition to contributing to the combustion process, air from the lift air blower assists with catalyst circulation. The performance of the reactor-regenerator section strongly affects the overall performance of the plant.

The implementation of model based controllers has proved highly advantageous, but depends on obtaining accurate models of the plant dynamics, based on input-output data, since a complete and accurate model from first principals is generally too complicated, expensive and time-consuming.

Complete details of the mechanistic simulation model for this particular model IV FCCU can be found in [71].

5.2.1 FCCU Experimental Design

The system identification procedure begins with allocation of the appropriate process measurements, needed in the model to be used by the model predictive controller. The FCCU simulator incorporates 10 manipulated variables, 3 disturbance variables and 36 measured output variables. However, the dynamic models of the FCCU system used in this study build a model based on the following 3 manipulated (or input) variables and 4 controlled (predicted or output) variables, as recommended in [72, 73].

The manipulated variables were selected to be:

- (1) the flow rate of the diesel fuel supply (A100),
- (2) the total fresh feed flow rate and (A102),
- (3) the supply of air to the process through a lift air blower (A108).

It is important to provide proper excitation for the process [5]. This was achieved by adding a pseudo random binary sequence (PRBS) to the constant signal of each of the manipulated variables.

The output prediction variables (Figures 5.3 and 5.4) were selected to be:

- (1) the temperature of the reactor (M308),
- (2) the temperature of the regenerator (M314),
- (3) the oxygen concentration in the stack gas (M317),
- (4) the position of the wet gas valve (M334).

Economic operation of the plant involves controlling the oxygen level in the stack gas to set-point, while maximising the feed flow rate for an acceptable level of conversion and may involve running the plant close to metallurgical constraints.

Two data sets (**C6** and **C7**), each corresponding to several days of operation (8000 data points) were created. The data set **C6** contains clipped data in the form of the oxygen concentration sensor (M317) that saturates at 3 mol/m³ oxygen. There is also a slow drift. This models the presence of an unmeasured disturbance, e.g. ambient temperature. **C7** is more linear, where the oxygen level doesn't breach the 3 mol/m³ saturation limit. For each data set, the wet gas valve is kept below its upper threshold saturation limit (100%).

The excitation signals for **C6** and **C7** are shown in Figure 5.5. Each has been obtained by applying pseudobinary input 12.8 ± 0.9 (A100), 125.1 ± 1.5 (A102) and 75.0 ± 1.0 (A108).

5.2.2 FCCU Model Identification

The data were mean centred and scaled to unit variance prior to the application of the identification algorithms. This helps guarantee the numerical accuracy of the calculations, and also scales the model residuals, so that a comparison of model accuracy from variable to variable is more meaningful.

The subspace models (see also Table 4.1) are as follows:

- $\mathbf{M}_1(\theta)$ is subid.m, as described in Chapter 3 and in [1].
- $\mathbf{M}_3(\theta)$ is n4sid.m from MSIT, with “weighting” = “moesp”.

The ARX and FIR models correspond to $\mathbf{M}_6(\theta)$ (arx.m), from MSIT. An iterative search was conducted to determine the appropriate model structures with which to model the system. The optimum model structures were found by splitting each of **C6** and **C7** into training and cross-validation data sets, then AIC and MSPE were calculated using MSIT. AIC was calculated as

$$AIC = \log(V(\theta)) + 2d / N, \quad (5.4)$$

$$V(\theta) = \det(\text{cov}(\hat{y}(\theta) - y)), \quad (5.5)$$

where d is the number of degrees of freedom in the model, N is the number of data points and $V(\theta)$ is a loss function calculated using the covariance of the prediction error matrix; \hat{y} are the model predictions and y are the process measurements.

5.2.3 Optimising the FCCU model structures

An iterative search for appropriate model structures involved investigating a range of state space model orders, and for each, iterating through a range of values for the number of block rows, r . ARX and FIR model structures were also investigated, where a range of model orders were considered, and for each, input delay spreads of values ranging from 2-100 time lags were tried.

Table 5.2 summarises the results for each of the model structures that minimised AIC for data sets **C6** and **C7**. In column two, the state space model structure is defined by

(a,b) , where a is the model order and b is the number of block rows used in the Hankel matrices. The ARX model structure is defined by (a,b,c) , a is the number of autoregressive terms, b is the delay spread for the manipulated variables, and c is the pure time delay incorporated into the model structure. The FIR model structure is defined by (a,b) ; a is the delay spread for the manipulated variables, and b is the pure time delay incorporated into the model structure. The values of MSPE represent the scaled prediction errors of the process variables, corresponding to scaling applied at the beginning of the identification procedure. These values contribute to AIC through the calculation of the loss function $V(\theta)$ (see Eq. 5.5).

Figures 5.6-5.13 show the predictions for the subspace models, \mathbf{M}_1 (o), \mathbf{M}_3 (x), the ARX model \mathbf{M}_6 (Δ) and FIR model \mathbf{M}_7 (∇), based on the model structures that minimise AIC (Table 5.2). Each of these linear models is applied to an approximately linear data set, and each of the models has performed reasonably well. Certainly it is not expected that one model would perform significantly better than the others, since linear systems theory allows linear transformations between linear model structures, for example between state space and ARX model structures. However, as found using the mass spring damper simulation in the previous chapter, and again here, AIC tends to identify “optimum” model structures, that are higher order models than may be necessary. For example, the minimum AIC was found for **C7** to be a 12th order model (\triangleright). However, as can be seen in Figures 5.19-5.23, a 4th order model (x) might be sufficient.

5.2.4 Choosing the number of block rows used in the subspace method and its effect on prediction accuracy

In this section, a novel way to present the results of subspace identification procedure is introduced. The mean squared prediction error vectors for the subspace algorithms are scaled to zero mean, unit variance so that the error for all four model predictions can be plotted on a single graph, across the entire range of models considered. The aim is to compare the effect of varying the number of block rows and the model order on each of the four prediction error vectors. Results are provided for both data sets **C6** and **C7**.

Figures 5.14 - 5.17 show the effect of increasing the number of block rows, on model accuracy, for algorithms \mathbf{M}_1 and \mathbf{M}_3 . These results have been uniformly scaled to zero mean and unit variance. The line $\circ \cdots \circ$ (in the form of a staircase) provides an indication of the model order, which increases step-wise from order 2 to order 12. The line (in the shape of saw teeth) $\Delta - - - \Delta$ indicates the (increasing) number of block rows B for each model order n ; $B = 1, 2, 3 \dots 20$, $r = n + B$ (see Figure 5.14). The effect of the number of block rows on prediction accuracy was evaluated by calculating the mean squared prediction error (MSPE) over an infinite horizon of 4000 data points, equivalent to almost 3 days of FCCU operation. In these figures, the specific process outputs (M308, M314, M317 and M334) are not differentiated, but are indicated by the four solid lines.

Figure 5.14 shows the MSPE for algorithm \mathbf{M}_1 , dataset **C6**, where it can be seen that increasing the number of block rows ($\Delta - - - \Delta$) leads to smaller error for model orders $n = 2, 4, 6, 8, 10$. The flat sections (e.g. in Figure 5.14, for model order $n = 12$ at $r = 21 - 27$ and $r = 30 - 31$) indicate that unstable models were identified. Note that for $\mathbf{M}_1, \mathbf{C6}$, the minimum AIC was found at $(n, r) = (12, 13)$, however 10^{th} order state space models generally performed the best.

Figure 5.15 shows the MSPE for algorithm \mathbf{M}_1 , dataset **C7**. In general, increasing the number of block rows ($\Delta - - - \Delta$) leads to smaller error for model orders $n = 2, 4, 6, 8, 10$. However, for model order $n = 12$, the performance of the algorithm becomes erratic as indicated by the prediction error of the output variables (the four solid lines) increasing with added block rows. Again the 10^{th} order state space models performed the best.

Figure 5.16 shows the results for \mathbf{M}_3 , dataset **C6**, where it can be seen that increasing the number of block rows ($\Delta - - - \Delta$) often leads to larger levels of prediction error, in particular for model orders $n = 4, 6, 8, 10, 12$. For model order $n = 10$, MSPE increased at first, then decreased. For model order $n = 12$, the accuracy fluctuated with increasing r . The most consistent results were obtained using model order $n = 10$.

Figure 5.17 shows the results for \mathbf{M}_3 , dataset **C7**. Data set **C7** reveals a similar pattern where the accuracy decreased as r increased for orders 2,4,6,8. For model order $n=10$, the number of block rows had less effect on model accuracy. The results were erratic for model order $n=12$. The most consistent results were obtained using model order $n=10$.

Some general trends are noted in Figures 5.14 – 5.17. For data set **C6**: on the basis of the sum MSPE across the four predicted variables, for models identified using algorithm $\mathbf{M}_1(\theta)$, increasing the number of block rows improved the results. In contrast, models identified using algorithm $\mathbf{M}_3(\theta)$ generally performed better using fewer block rows.

5.2.5 Using an eigenvalue plot to determine the model order

Figure 5.18 shows eigenvalue plots from the \mathbf{M}_1 (N4SID) subspace system identification procedure, that are used to estimate the system order. The 10th order model “cut-off” point is indicated by the shaded region on the bar chart. As can be seen, choosing the appropriate cut-off point is a non-trivial procedure. One approach is to choose a likely model order, then use cross-validation on a range of model orders in the region of the one chosen, then choose an appropriate order according to the results. For example, Figures 5.19-5.22 show cross-validation results for model orders $n=2,4,6,8,10$. It can be seen that even though the minimum AIC (Table 5.1) was found for a 12th order (\triangleright) model, that a 2nd order model (o) captures a significant proportion of the dynamics.

5.2.6 State space model order reduction

Figures 5.19-5.22 illustrate the performance of $\mathbf{M}_1(\theta)$, for model orders 2(o), 4(x), 6(Δ), 8(∇), 10(\square) and 12(\triangleright). The solid line shows the measured output of the process. Subspace methods deliver a balanced realisation. This means that the greatest part of the variation is captured by the first state sequence, then the second and so on. It is therefore easy to calculate a reduced order model, where the user bases the choice of the model order according to the relative magnitude of the singular values of the SVD (Figure 5.18).

Figures 5.19-5.22 show that even though the higher order models improve the prediction accuracy, that lower order models (e.g. a 4th order model) capture most of the process variation. In this respect, AIC can be regarded as a secondary measure of system order, when the probability density function of the process measurements is unknown, and because, in many cases, lower order models capture sufficient process dynamics for the application in hand. For example, a comparison of the 4th order (x) and 10th order (□) models in Figure 5.19, reveals that although the MSPE of the 10th order model is smaller, that the 4th order model has performed nearly as well.

5.2.7 A comparison between state space and ARX model structures

Figures 5.23-5.27 present a direct comparison of 4th order state space models (\mathbf{M}_1 and \mathbf{M}_3) and a 4th order ARX model. **C7** was used to generate infinite horizon predictions for \mathbf{M}_1 (▷), \mathbf{M}_3 (x) and \mathbf{M}_6 (Δ). The figures illustrate the power of subspace system identification, where the SVD is employed to find orthogonal state sequences that describe the system dynamics in fewer dimensions than the ARX model structure is able. Figure 5.27 shows the residuals from each of the 4th order model predictions. It is easy to see that the ARX model (\mathbf{M}_6) has not been able to catch all the important process dynamics. Table 5.3 gives the error for each of the 4th order models. The first four columns show the (scaled) MSPE for each of the output variables. The final column gives the value of the cost function $V(\theta)$. The fourth order \mathbf{M}_1 model has performed better than the fourth order \mathbf{M}_3 model, each doing better than \mathbf{M}_6 .

5.2.8 Relative speed of the algorithms

A simple test was applied using \mathbf{M}_1 and \mathbf{M}_3 : each algorithm was run with $(n,r)=(10,25)$. The user choice “cov” = “none” was applied for \mathbf{M}_3 , since the calculation of the covariance matrix takes up a considerable amount of the computational effort. It was found that \mathbf{M}_1 (subid.m) was 15-20% faster than \mathbf{M}_3 .

(n4sid.m) for the two data sets considered. The “best model” prediction error for \mathbf{M}_1 for **C6** and **C7** was also found to be smaller than that for \mathbf{M}_3 .

5.3 Conclusion

In this chapter, linear parametric models that incorporate state-space, ARX and FIR model structures have been used to model a complex industrial processes. As was found in the previous chapter, the data sets themselves have *the* major effect on the accuracy obtained. Generally speaking, linear methods deliver linear models, i.e. providing the system is “reasonably” linear, and that the correct model structure is used, then all the linear methods considered in this study can be expected to deliver reasonably accurate models. However, the model structure, the algorithms used and the number of parameters required for each of the methods may differ considerably. A summary of the findings and conclusions from the two studies in this chapter is as follows:

- The main advantage of using FIR models lies in their relative simplicity, however they often require a large number of parameters. For example the FIR model of **C7** required 84 input lags, meaning a model with 1032 parameters, and a large computational burden for a controller. In the case of large dimension MIMO systems, the number of parameters required for the FIR model structure soon becomes prodigious. In contrast, the fully paramatised state space models delivered by subspace system identification are able to provide a more parsimonious description of the system dynamics.
- The power of subspace models has been demonstrated on the basis of the prediction accuracy of low order state space models and equivalent order ARX models. ARX models use fewer parameters than FIR models, however subspace system identification methods are able to determine low order state space models that are generally more concise than ARX models, because they describe the system using powerful directions in state space, while the ARX model can only describe the system in terms of the vector space defined by the measured inputs and outputs. This makes subspace methods naturally suited to handling process data with correlated outputs. The subspace methods identified 4th order state space models for the FCCU that were far more accurate than equivalent 4th order ARX models.

- Section 5.2.4 and Figures 5.14 – 5.17 introduce a novel way to present results from subspace identification procedure. This method was found to give a very clear indication of the way forward when trying to determine the correct permutation of model order and the number of block rows of data to use in the algorithms.

- The subspace algorithms determine state sequences to describe the process variation, where the discarded state sequences are more affected by measurement noise. However, the FIR and the ARX models use only the process measurements, and do not have such a “filtering” mechanism.

- The performance of two subspace algorithms, `subid.m` (\mathbf{M}_1), and `n4sid.m` (\mathbf{M}_3) has been compared where some variation in the relative performance of each was found, depending on the system identified. Both algorithms were used to identify linear state space models for the FCCU simulation. On the basis of the MSPE for the FCCU validation data sets, low order (2^{nd} and 4^{th}) models of the system, identified by \mathbf{M}_1 performed well. In contrast, higher order (8^{th} , 10^{th} and 12^{th}) models identified by \mathbf{M}_3 were found to provide more accuracy. In theory [32], each of the subspace methods identifies a state space that is (up to within a similarity transformation) equivalent to the original process. However the inevitable presence of noise and disturbances in process data, means that under certain circumstances, one algorithm may work better than another. Therefore, the user might consider trying two or more subspace algorithms and compare results.

- It is very important to identify the appropriate state space model order, and sufficient resources should be allocated to it. The studies applied here have revealed that AIC tends to identify higher order state space models than may be required. The AIC does not necessarily provide the “optimum” solution, because it assumes that the probability distribution of the measurements is known. Because this can at best be estimated, the AIC may sometimes be better considered as a secondary measure of the likely system order.

A pragmatic approach to determining (n, r) is to employ a combination of the N4SID eigenvalue plot, cross-validation, and AIC. The eigenvalue plot (Figure 5.15) gives a

good indication of the likely model order, then this initial estimate can be further evaluated using cross-validation.

- The results from this study help substantiate the observations presented in Qin and Badgwell (2001) [4], who claim that the next generation linear MPC modelling techniques will feature the use of subspace methods to identify linear models in state space. A key advantage of state space models is that they deliver fully parameterised state space models, for which there exists a wealth of control theory.
- The subspace method is naturally suited to handling correlated inputs and/or outputs. This leads to a more parsimonious description of the data using powerful directions in subspace.
- An advantage of the subspace approach (but also a challenge), is the degree of flexibility that choosing the number of block rows, r , brings to the identification procedure. This user choice, although bringing flexibility, also introduces a complexity, because care and attention is required in choosing r . The procedure has been automated in MSIT, however specifying the user choice manually provides a greater scope (at the expense of time and effort) for model optimisation.

The subspace method employs a singular value decomposition to calculate the principal directions in the vector space defined by the input-output data set. Directions of maximum correlation are described, where the first state contributes most to the process dynamics and the second state makes the second most significant contribution and so on. It has already been pointed out that this enables a more accurate description of the system, using fewer state variables than for an ARX model. In the next chapter, this ability to find a low dimension latent state space will be used to develop a process condition monitoring model that provides an alternative to currently employed dynamic models in monitorMV.

5.4 Figures and Tables

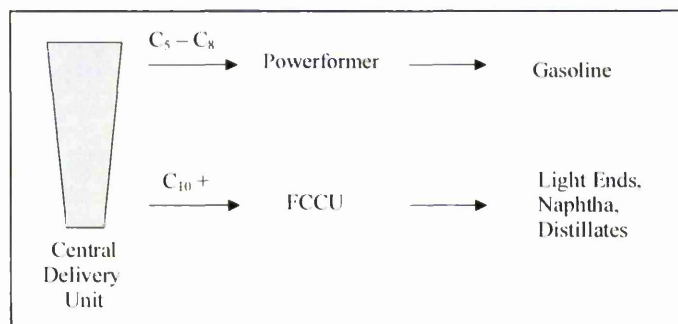


Figure 5.1 Crude oil first passes to the CDU where it is separated into light, medium and heavy cuts. The heavier grades go to the FCCU for catalytic cracking.

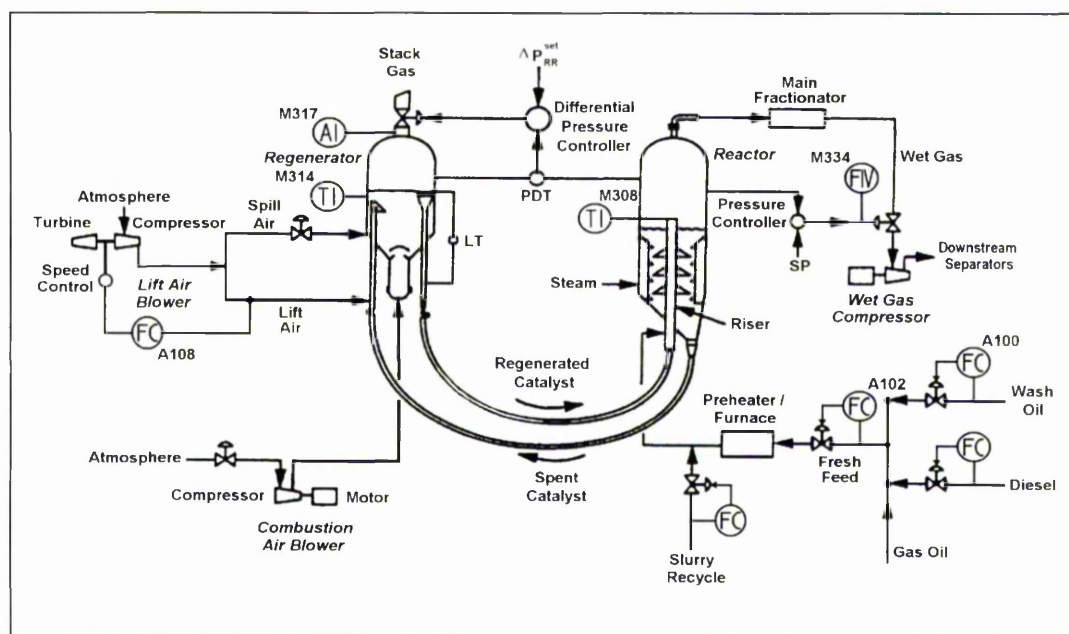


Figure 5.2 Schematic of the fluid catalytic cracking unit simulation [71].

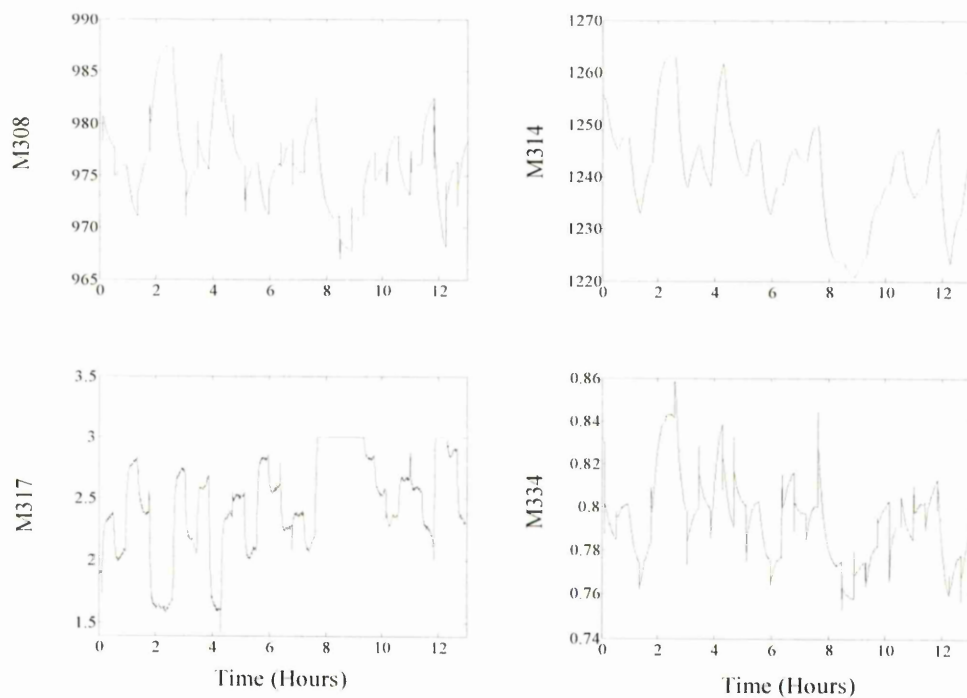


Figure 5.3 The FCCU simulator outputs (C6).

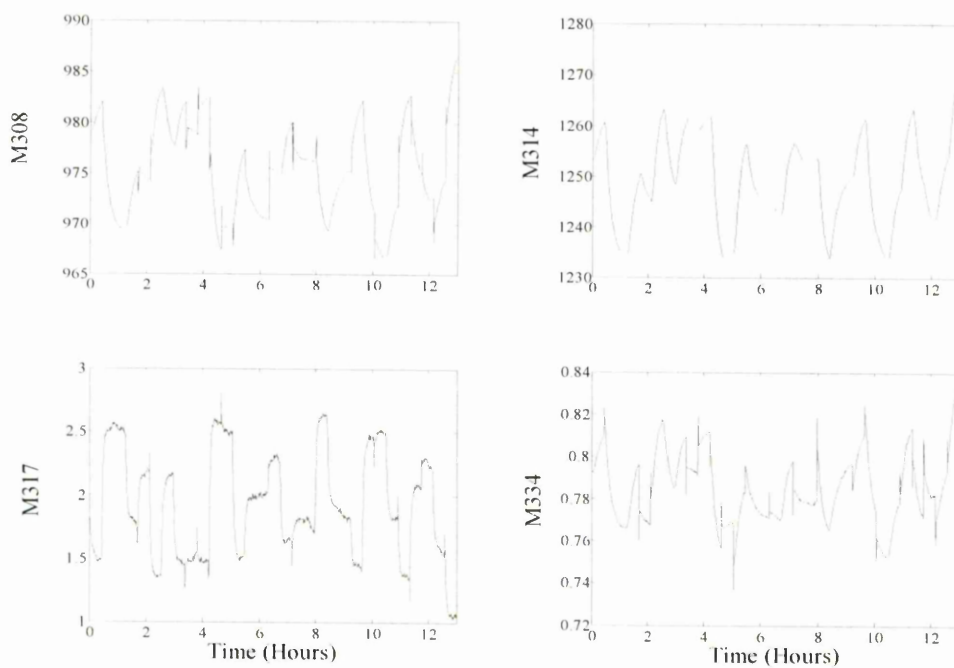


Figure 5.4 The FCCU simulator outputs (C7).

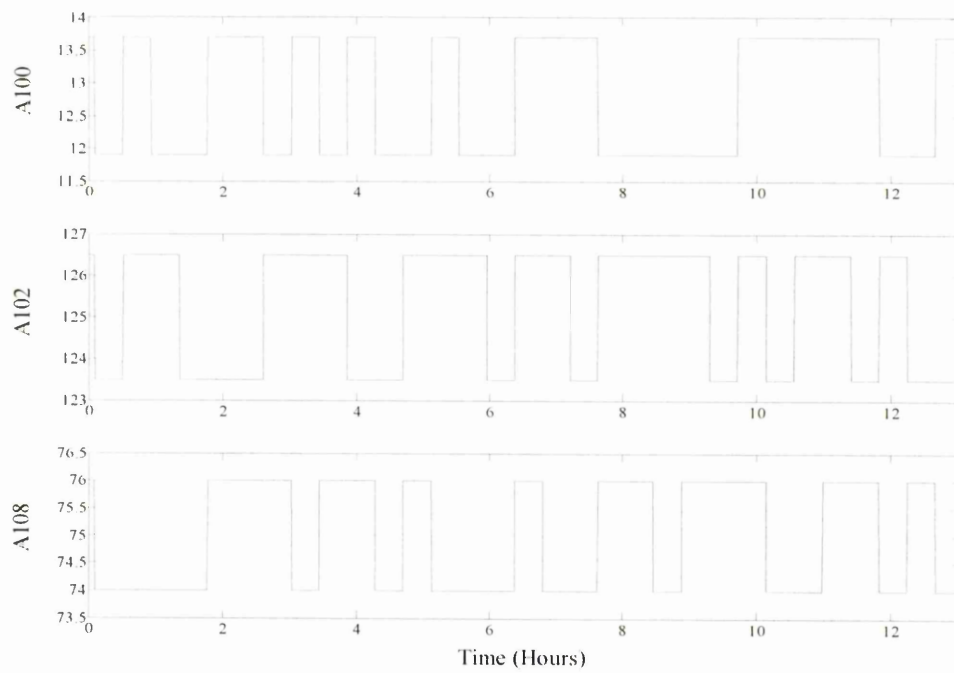


Figure 5.5 Pseudo binary excitation signals (**C6** and **C7**).

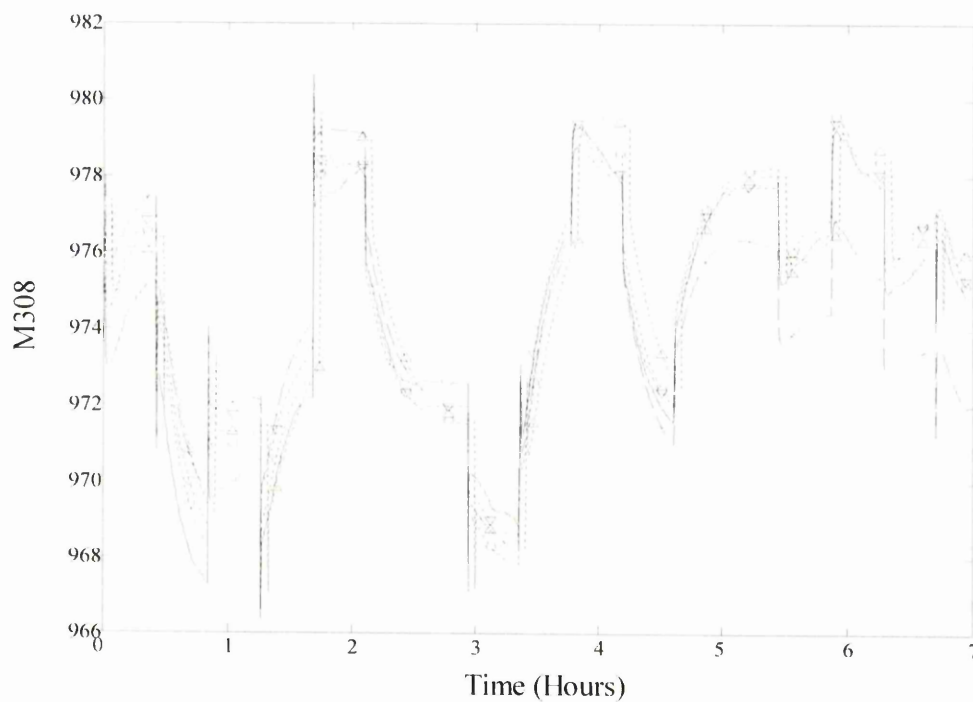


Figure 5.6 **C6** riser temperature predictions. M_1 (o), M_3 (x), M_6 (Δ) and M_7 (∇).

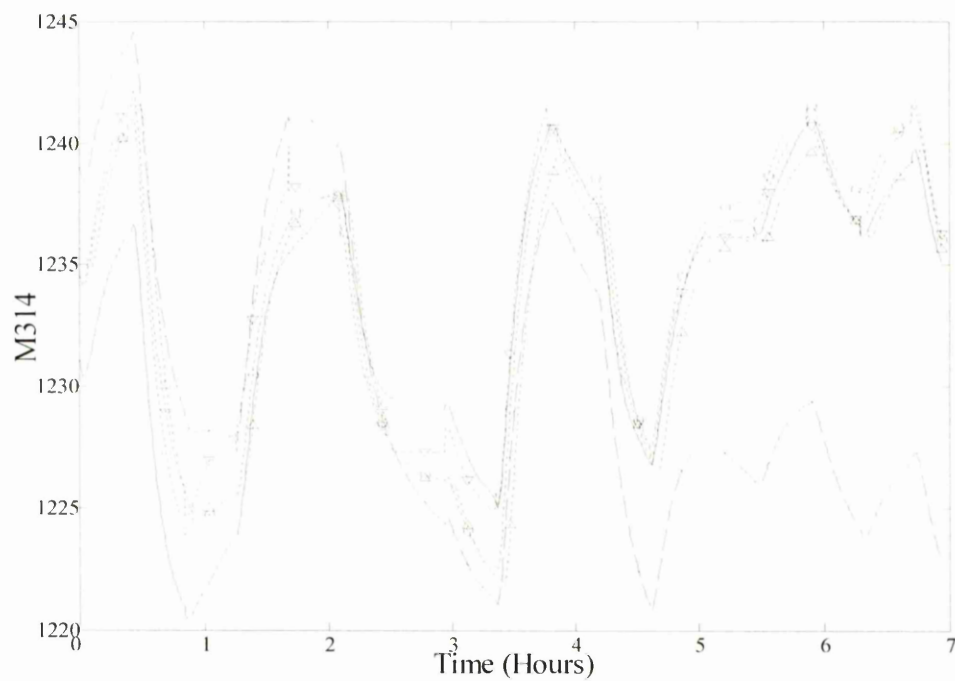


Figure 5.7 C6 regenerator temperature predictions. $M_1(o)$, $M_3(x)$, $M_6(\Delta)$ and $M_7(\nabla)$.

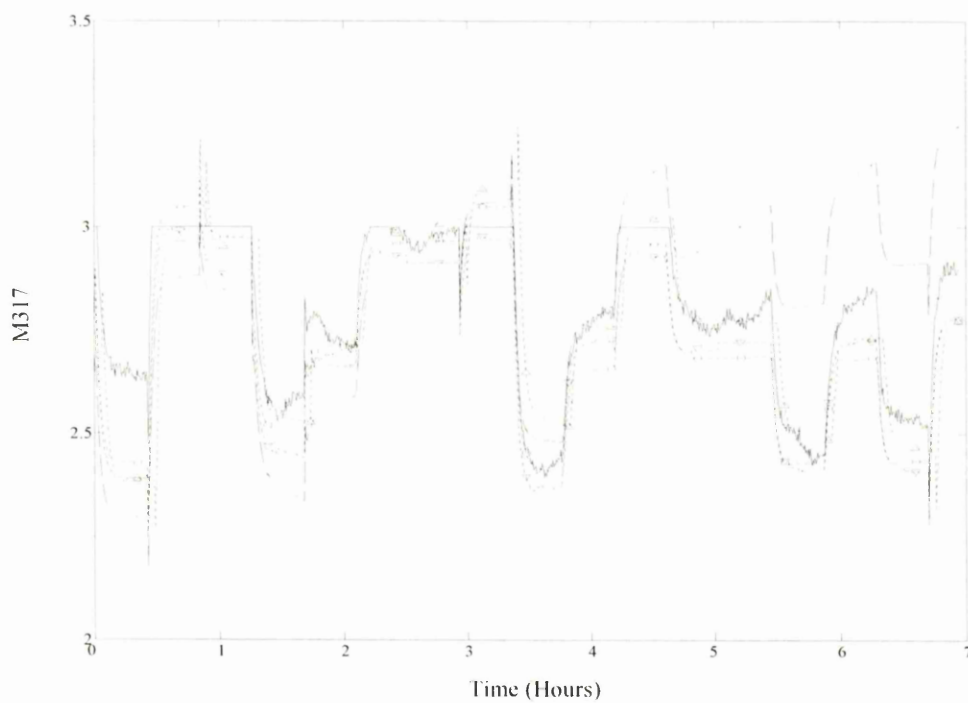


Figure 5.8 C6 oxygen concentration predictions. $M_1(o)$, $M_3(x)$, $M_6(\Delta)$ and $M_7(\nabla)$.

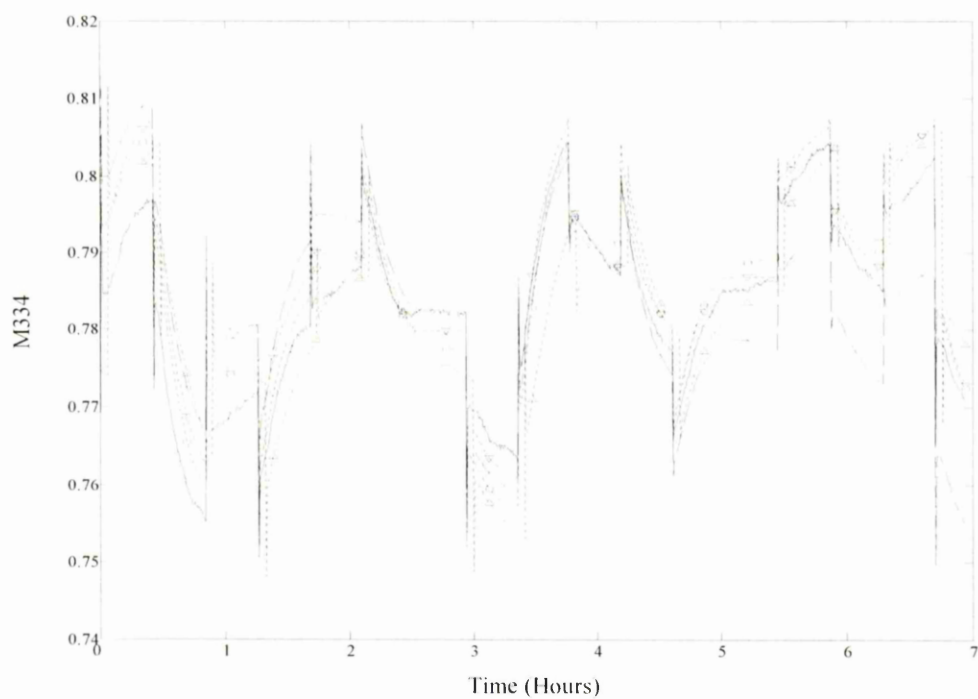


Figure 5.9 C6 wet gas valve position predictions. $M_1(o)$, $M_3(x)$, $M_6(\Delta)$ and $M_7(\nabla)$.

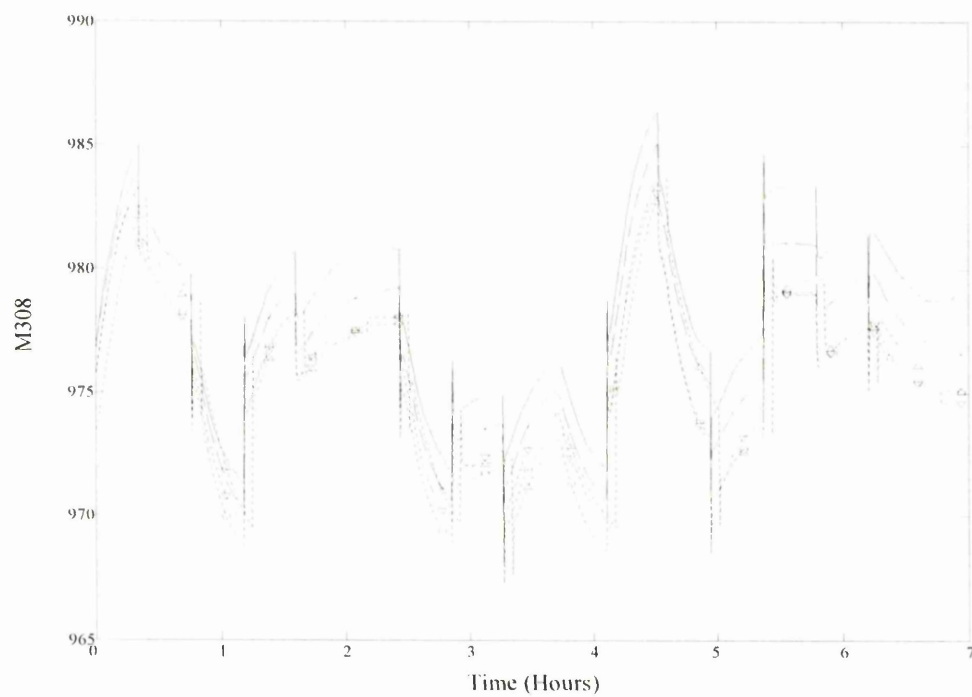


Figure 5.10 C7 riser temperature predictions. $M_1(o)$, $M_3(x)$, $M_6(\Delta)$ and $M_7(\nabla)$.

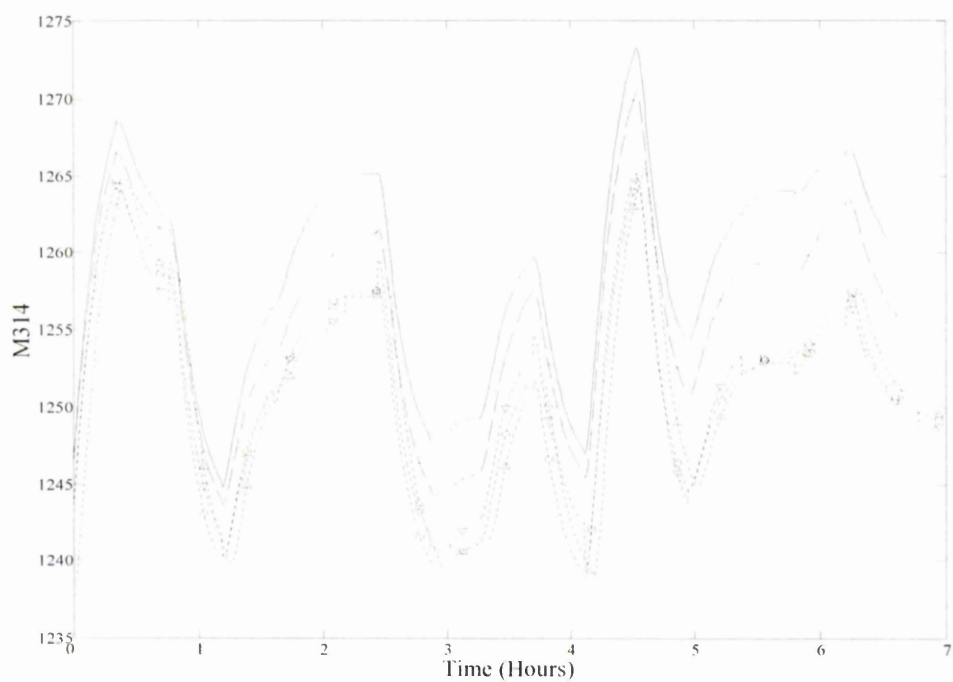


Figure 5.11 C7 regenerator temperature predictions. $M_1(o)$, $M_3(x)$, $M_6(\Delta)$ and $M_7(\nabla)$.

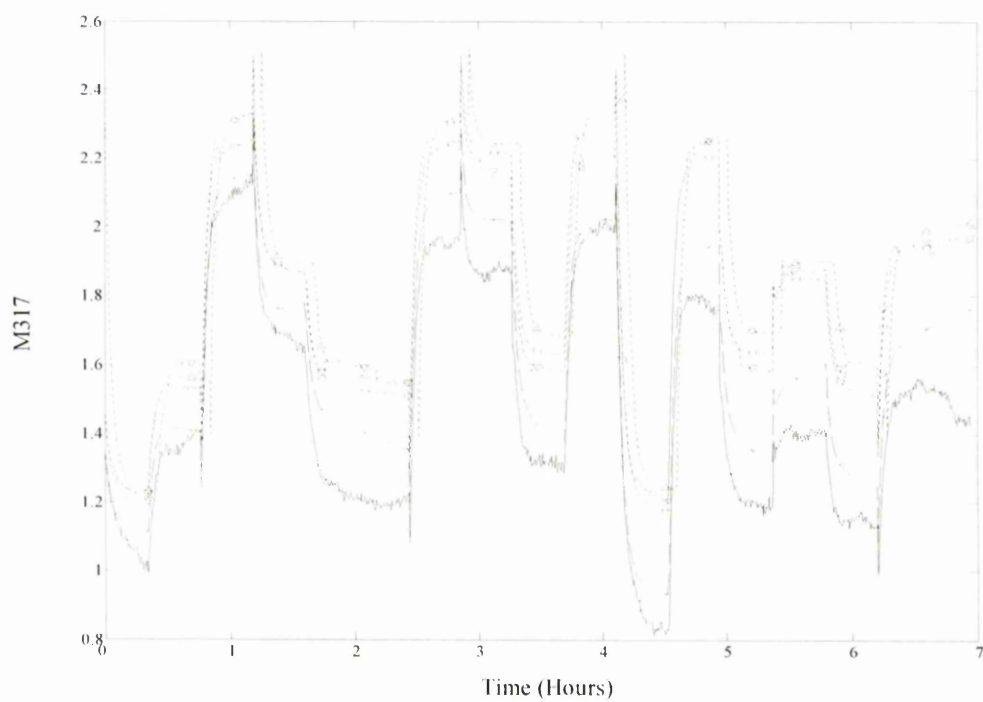


Figure 5.12 C7 oxygen concentration predictions. $M_1(o)$, $M_3(x)$, $M_6(\Delta)$ and $M_7(\nabla)$.

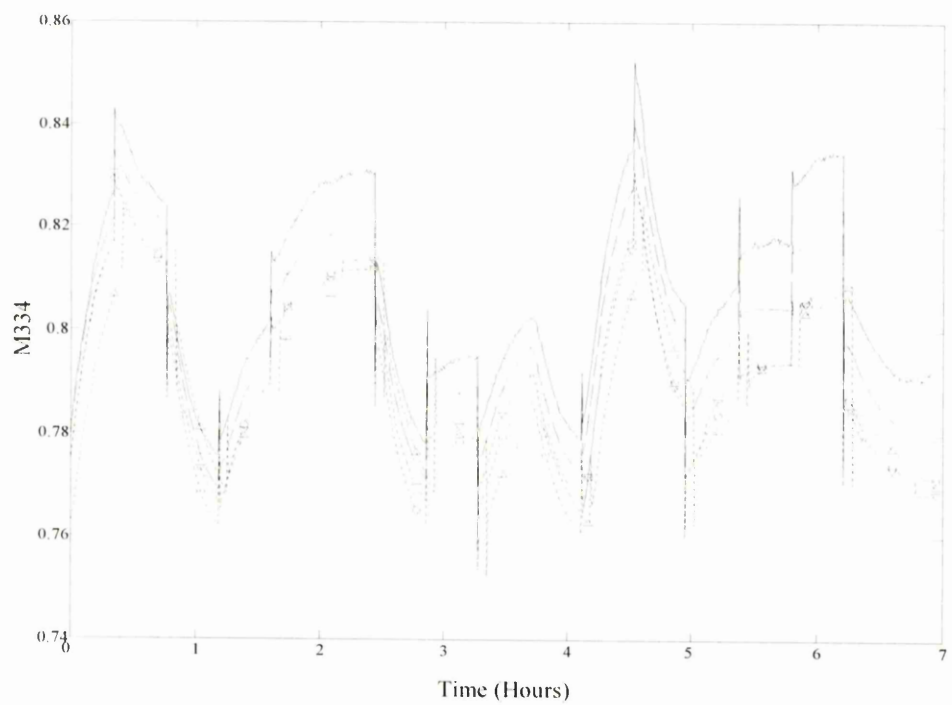


Figure 5.13 C7 wet gas valve position predictions. $M_1(o)$, $M_3(x)$, $M_6(\Delta)$ and $M_7(\nabla)$.

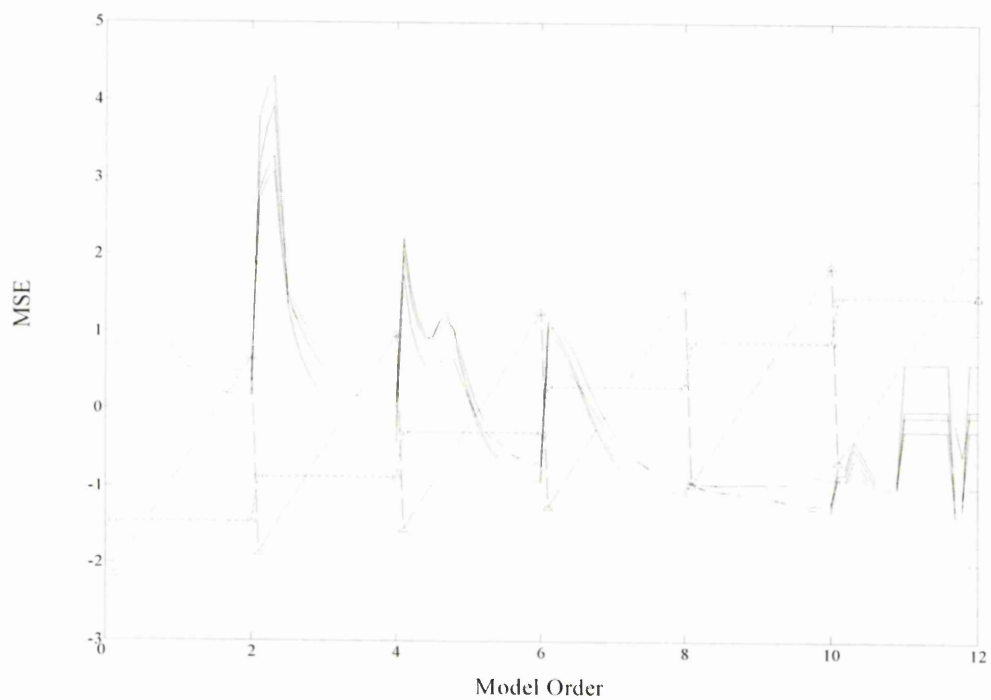


Figure 5.14 MSPE (C6). $M_1(\theta)$

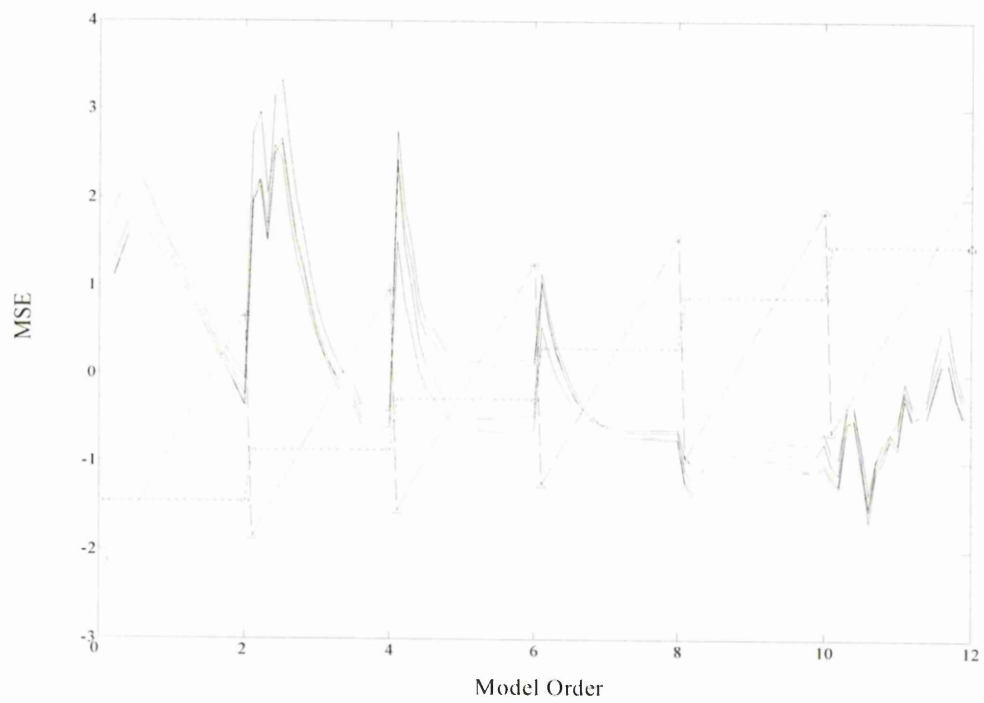


Figure 5.15 MSPE (C7). $\mathbf{M}_1(\theta)$

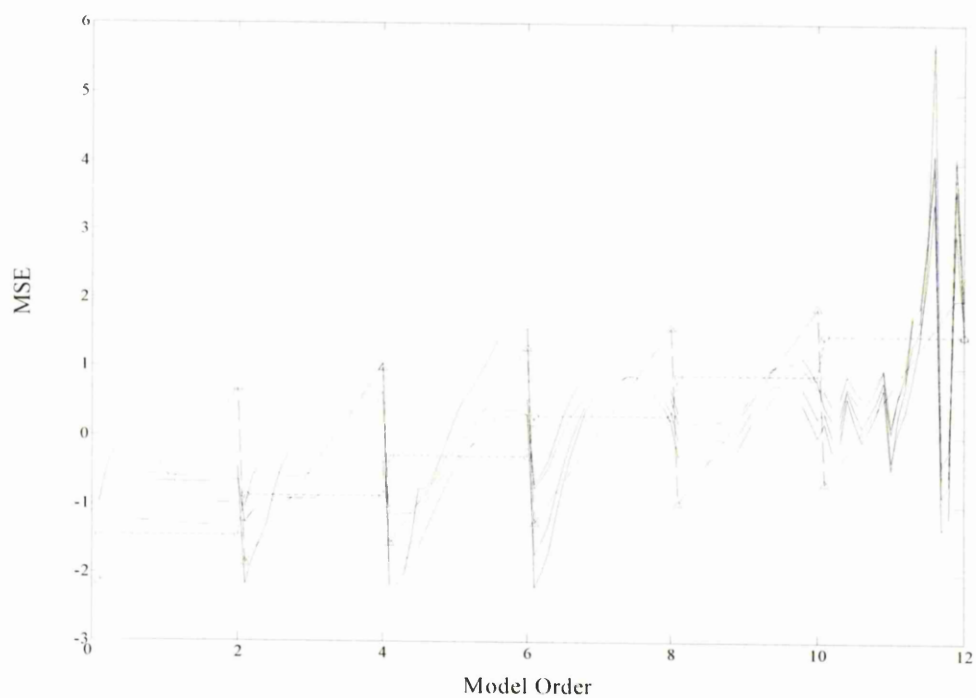


Figure 5.16 MSPE (C6). $\mathbf{M}_3(\theta)$

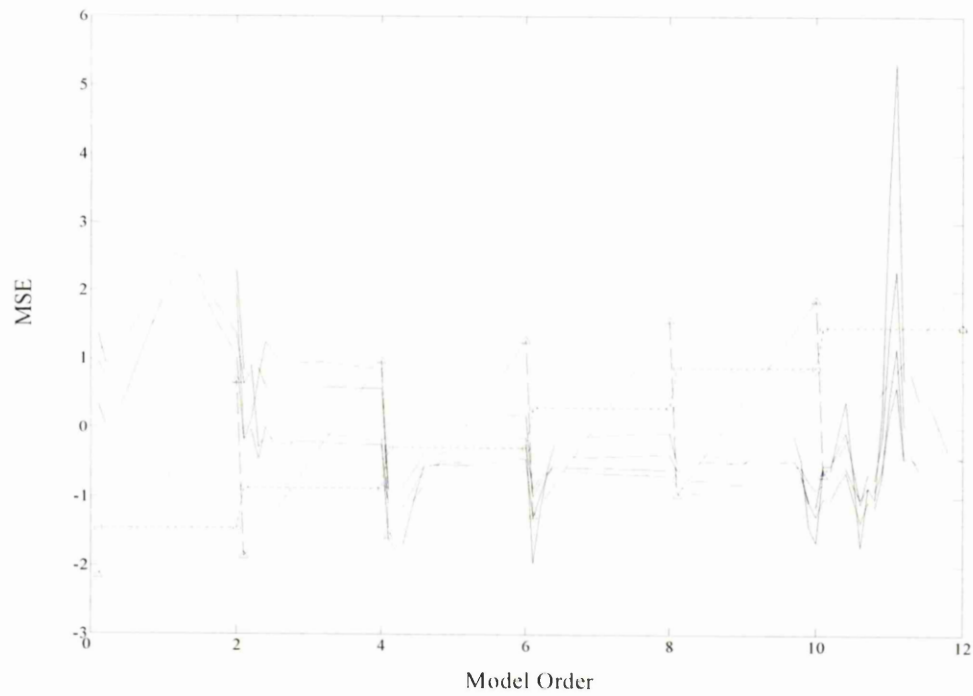


Figure 5.17 MSPE (**C7**). $M_3(\theta)$

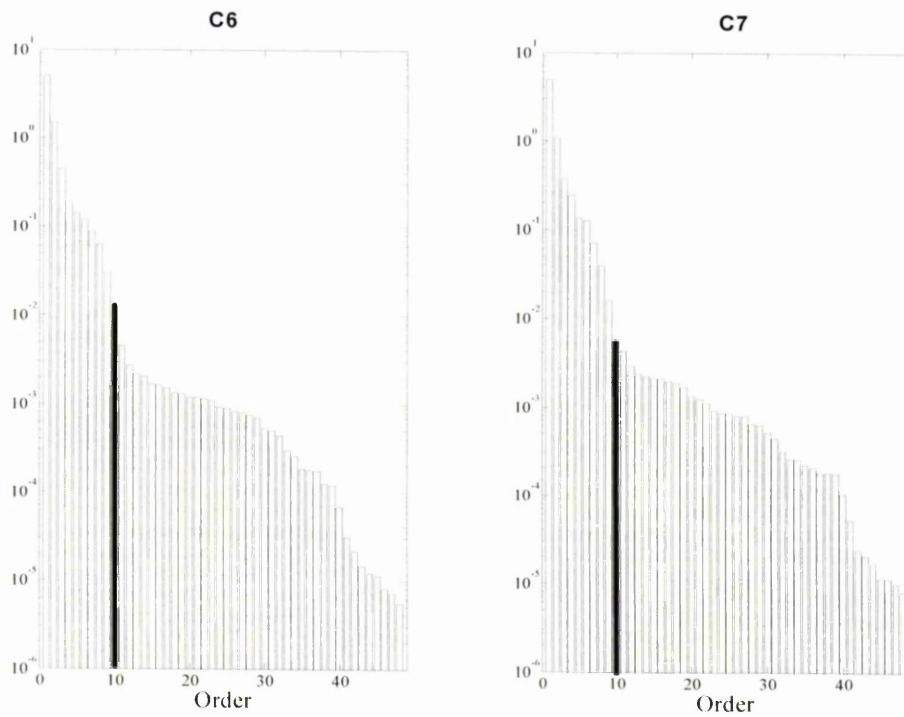


Figure 5.18 $M_1(\theta)$ eigenvalue plots. For data sets **C6** and **C7**. The bold value is the selected state space model order (10).

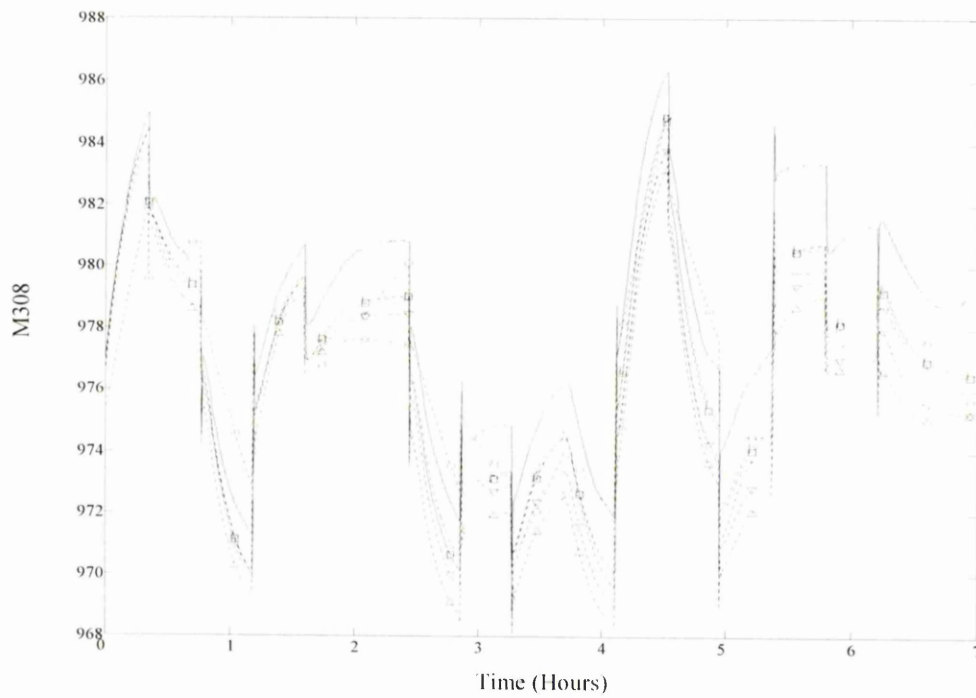


Figure 5.19 C7 riser temperature predictions. $\mathbf{M}_1(\theta)$ Model orders model orders 2(\circ), 4(\times), 6(Δ), 8(∇), 10(\square) and 12(\triangleright).

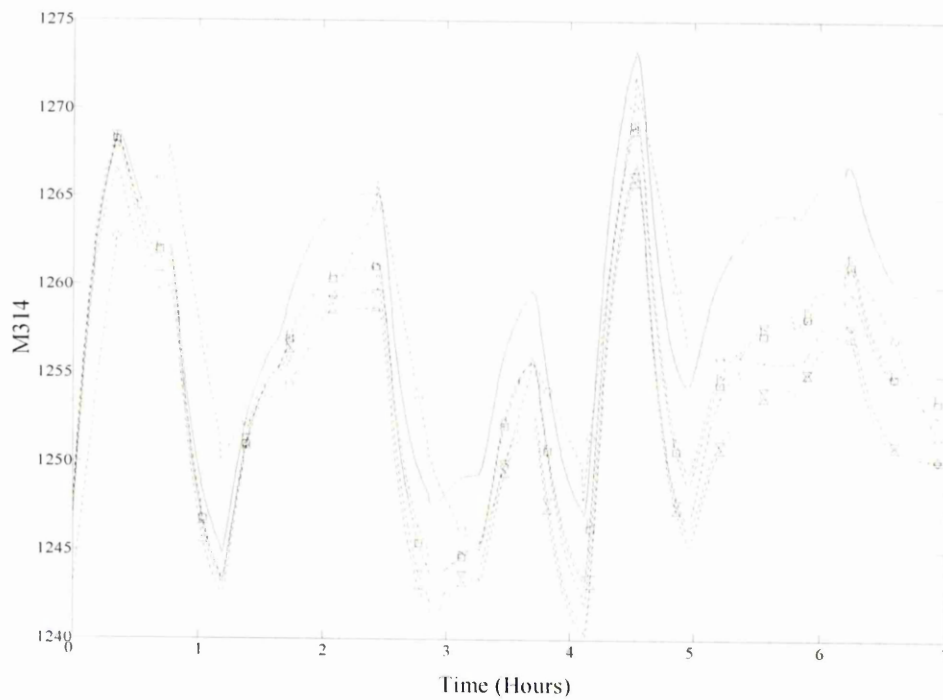


Figure 5.20 C7 regenerator temperature predictions. $\mathbf{M}_1(\theta)$ Model orders model orders 2(\circ), 4(\times), 6(Δ), 8(∇), 10(\square) and 12(\triangleright).

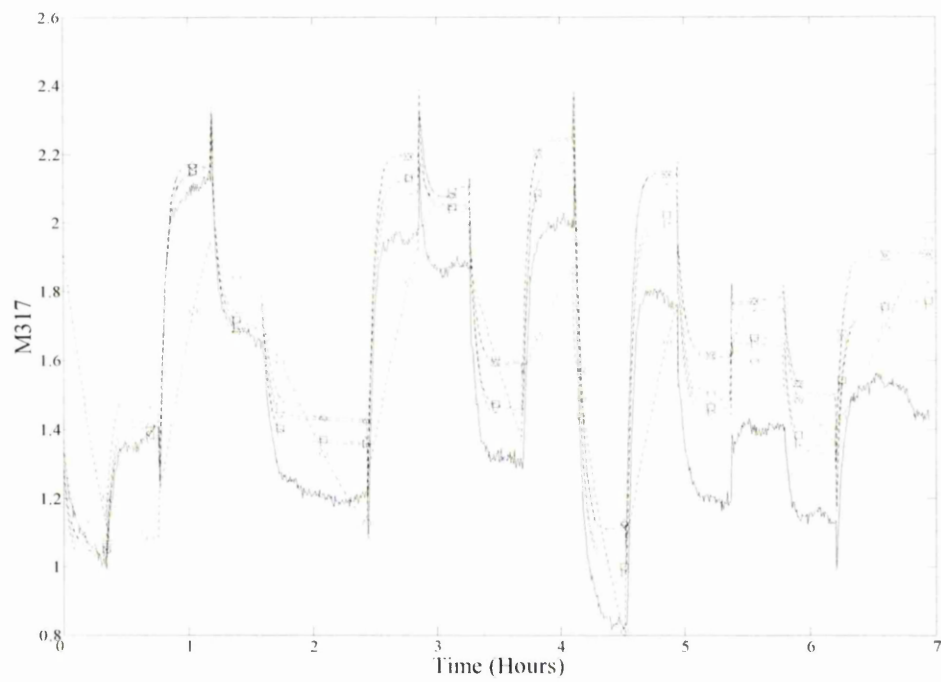


Figure 5.21 C7 oxygen concentration predictions. $\mathbf{M}_1(\theta)$ Model orders 2(\circ), 4(x), 6(Δ), 8(∇), 10(\square) and 12(\triangleright).

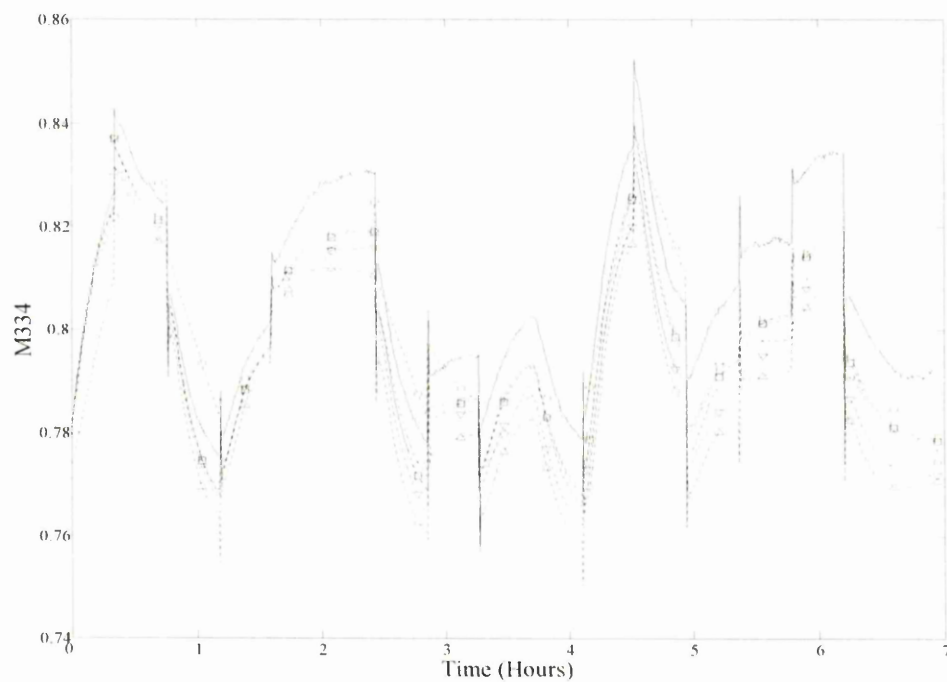


Figure 5.22 C7 wet gas valve position predictions. $\mathbf{M}_1(\theta)$ Model orders 2(\circ), 4(x), 6(Δ), 8(∇), 10(\square) and 12(\triangleright).

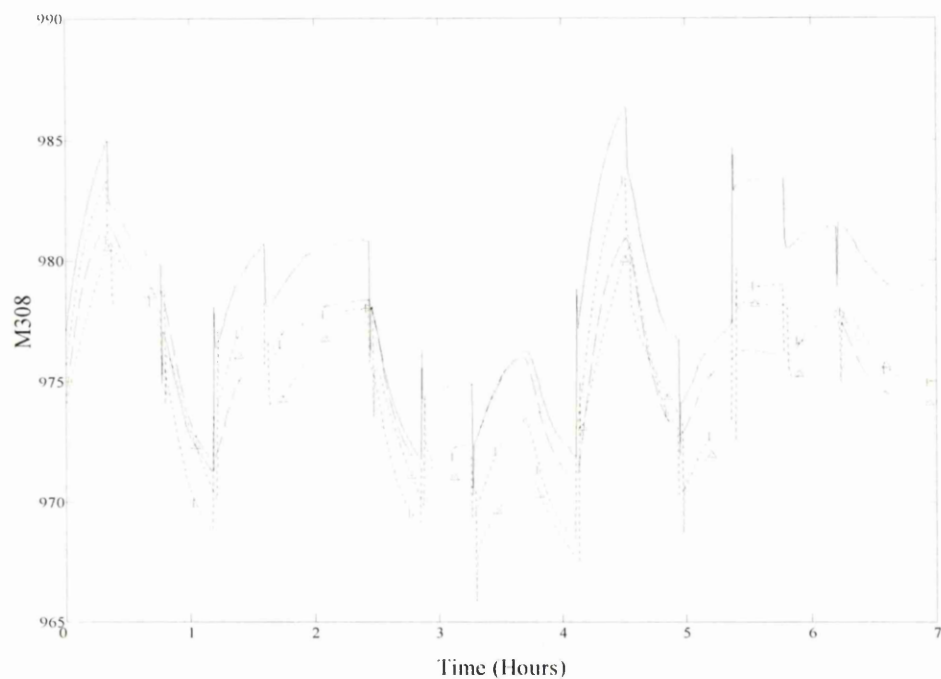


Figure 5.23 C7 riser temperature predictions. 4th Order models, M_1 (\triangleright), M_3 (x) and M_6 (Δ).

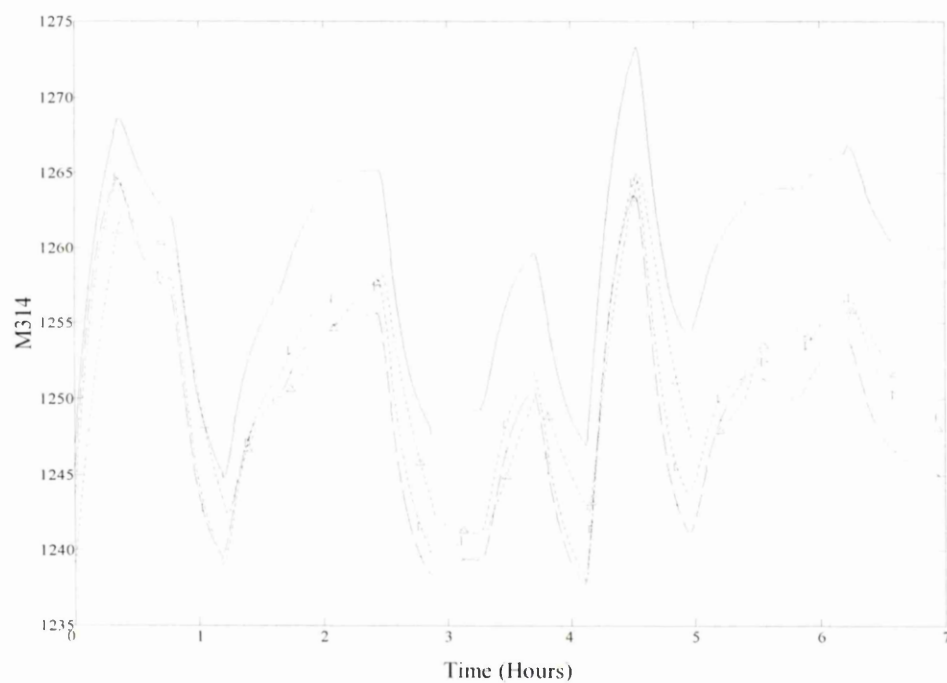


Figure 5.24 C7 regenerator temperature predictions. 4th Order models, M_1 (\triangleright), M_3 (x) and M_6 (Δ).

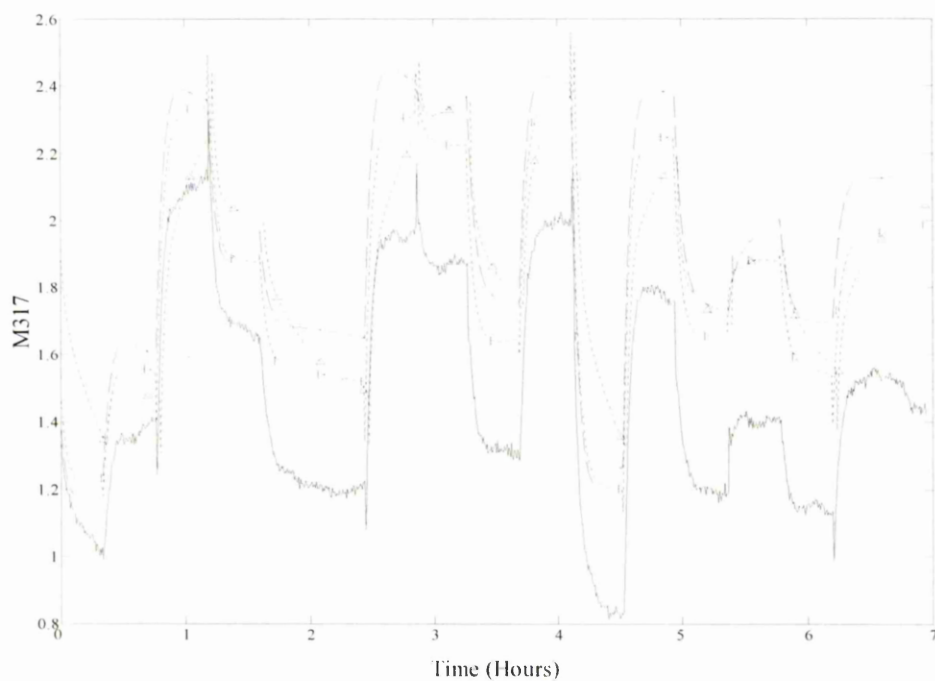


Figure 5.25 C7 oxygen concentration predictions. 4th Order models, M_1 (\triangleright), M_3 (x) and M_6 (Δ).

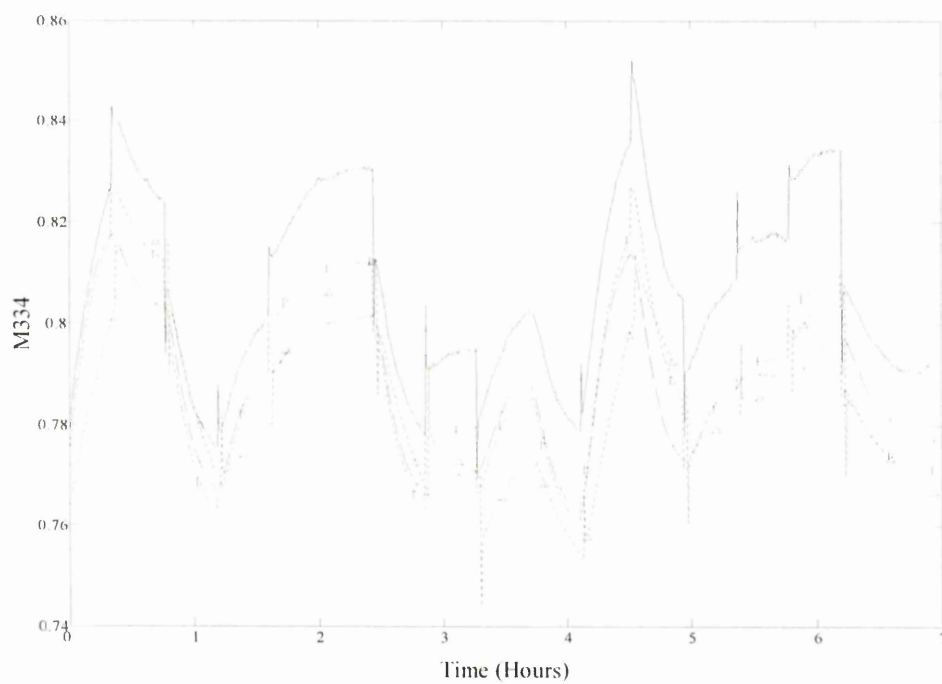


Figure 5.26 C7 wet gas valve position predictions. 4th Order models, M_1 (\triangleright), M_3 (x) and M_6 (Δ).

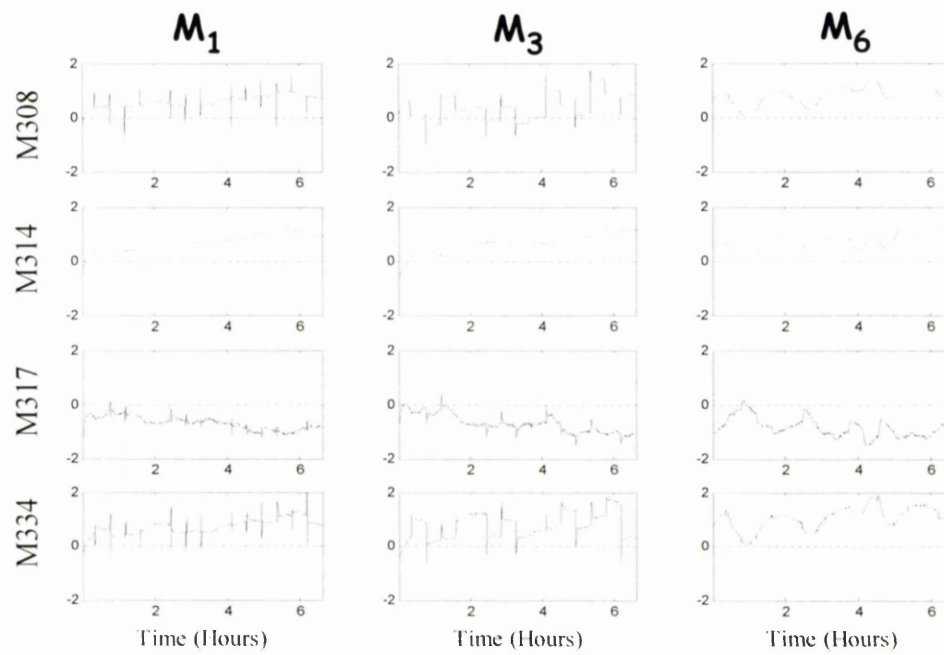


Figure 5.27 C7 model residuals, 4th order models.

	Data set C6			Data set C7		
	Model Structure	MSPE	$V(\theta)$	Model Structure	MSPE	$V(\theta)$
$M_1(\theta)$ N4SID	(12,13)	0.1731 0.1674 0.1705 0.2431	1.59×10^{-7}	(12,12)	0.1802 0.2728 0.2446 0.2534	7.01×10^{-9}
$M_3(\theta)$ MOESP	(10,14)	0.4025 0.8622 0.7705 0.5671	3.10×10^{-6}	(12,16)	0.1751 0.2768 0.2614 0.2557	2.44×10^{-8}
$M_6(\theta)$ ARX	(8,10,0)	0.2023 0.2362 0.2191 0.3515	1.47×10^{-5}	(10,11,0)	0.3581 0.5775 0.5485 0.6598	1.80×10^{-7}
$M_7(\theta)$ FIR	(60,0)	0.1867 0.1872 0.1995 0.2415	1.62×10^{-7}	(86,0)	0.3823 0.5768 0.5295 0.5874	1.03×10^{-8}

Table 5.1 Model structures corresponding to minimum AIC for **C6** and **C7**.

	MSPE				$V(\theta)$
	M308	M314	M317	M334	
$\mathbf{M}_1(\theta)$	0.2769	0.5059	0.5104	0.4520	6.42×10^{-7}
$\mathbf{M}_3(\theta)$	0.4334	0.6815	0.7370	0.7742	8.08×10^{-5}
$\mathbf{M}_6(\theta)$	0.5555	0.8053	0.8951	0.8722	4.77×10^{-6}

Table 5.2 MSPE **C7** . 4th order models.

Chapter 6

A subspace method for monitorMV

The principles of subspace system identification are used to develop a dynamic model for multivariate statistical process control. Hotellings T^2 and Q statistics are defined. The novel method uses a state space model structure to monitor continuous processes. It is shown to correspond to a principal components analysis of an augmented matrix containing the process measurements and the associated state sequences of the process. A procedure is outlined for optimising user choices concerning the structure of the model. Finally, an online condition monitoring scheme for the subspace method is outlined.

6.1 Introduction

The requirement for profit and efficiency in the chemical, manufacturing and other process industries imposes a constant need for attention to condition monitoring strategies for industrial plant. In addition, environmental issues are of primary importance in the management of process operation, where concerns about global climate change are bringing increasing pressure to reduce carbon dioxide and other toxic emissions. Elsewhere, recent catastrophic failures in chemical plants (e.g. the Union Carbide Plant in Bhopal, India) underline the importance of condition monitoring

strategies not only for plant operation but for the security of the communities that surround them.

The profitability of continuous industrial processes depends on the avoidance of expensive shutdowns and the ability to maintain the process within specification. Effective maintenance strategy includes early warning of impending failure and the ability to detect and diagnose process faults. The latest in process monitoring technology is therefore highly desirable to help meet the demands of product and operation specifications, and for safe, reliable and slick operation of industrial processes.

MonitorMV provides an important part of the solution for economic operation and safer, more reliable plant. It combines a group of methodologies that address the need for detecting and diagnosing abnormal process behaviour. In Part II of this dissertation, a new dynamic modelling capability involving a subspace approach is developed and assessed. The theory for the subspace method is expected to broaden the range of possible applications for monitorMV.

Given that the process variables of industrial plant are often highly correlated, it is possible to identify low order state space models using subspace system identification. This was demonstrated in Chapter 5 on the basis of an FCCU simulation. In this Chapter, it will be shown that further dimension reduction is possible leading to a novel subspace method for process condition monitoring. This is compared in the next chapters with alternative approaches, including dynamic principal component analysis (DPCA). The DPCA method is currently available in the monitorMV package and provides a benchmark for the performance of the novel subspace method.

The subspace method is cast into a statistical framework in such a way that it may be considered to belong to the group of technologies that come under the banner of multivariate statistical process control (MSPC). In addition, contribution charts have been developed for the subspace monitoring approach, to be used for diagnosing anomalous process behaviour.

MSPC is a multivariate extension to statistical process control (SPC) methods which employ univariate charts to assess the quality of processes. Univariate SPC charts plot key product variables as a function of time in order to detect the occurrence of special

events that usually have assignable causes [76]. These fault signatures can be used to obtain improvements in process and product quality by eliminating causes or by improving process operation [76]. In general, SPC methods chart only a few key output variables - in most cases the product quality variables which are examined one at a time. The application of SPC has been demonstrated to be a valuable aid to process engineering however these procedures are mostly inadequate for modern processes mainly because they treat each of the quality variables as being independent, when industrial data actually consists of highly correlated process variables [77-79]. The size and multivariate nature of modern process data therefore requires a multivariate extension to SPC techniques for identifying and isolating the cause of abnormal process behaviour [80].

For these reasons, the very important task of detecting and diagnosing abnormal process behaviour has led to the evolution of a range of condition monitoring strategies, collectively referred to as multivariate statistical process control which have been the subject of considerable research interest and a large number of publications over the past decade e.g. [76, 81, 82]. Multivariate analysis provides the means for dealing with large numbers of highly correlated process data [83]. This is done by defining a reduced set of statistically uncorrelated variables [76, 81, 84], thereby removing the high degree of redundancy in the data.

The most well known MSPC techniques are PCA and PLS, which use orthogonal projections that exploit cross-correlation among the process variables [77, 84-87]. The reduced variable set is then used to determine univariate statistics for on-line process monitoring, i.e. the Hotelling's T^2 and Q statistics. However, the PCA/PLS analysis relies on a static model that assumes the process operates at a predefined steady-state condition, which is often not the case. For example, the process under study may contain non-stationary and/or time-varying behaviour [88,89]. In such circumstances, the steady-state MSPC approach with constant confidence limits gives rise to frequent false alarms. One way to address the problem of false alarms is to apply a recursive adaptation of the MSPC models, and an adaptation of the confidence limits to accommodate time-varying and non-stationary process behaviour [89, 90].

Other problems may arise with throughput changes, which result in dynamic transients. Also controller feedback causes process variables to be auto-correlated [80]. This will

invalidate the theory when the assumption imposed on the statistically based monitoring approach is that the process variables are independent, normally distributed sequences. The presence of auto-correlation and dynamics in process data can provide a considerable challenge for the static modelling methods. One option available in monitorMV is to use dynamic PCA [91] to capture dynamic process behaviour, where time-shifted process variables are added to an augmented data matrix, which is usually based on an ARX structure. However it is found that analysis of MSPC contribution plots, for models based on the ARX model structure, is sometimes cumbersome due to the number of process measurements used to build the model.

6.1.1 State space models for process monitoring

An alternative to using the ARX structure for process monitoring is to use state-space models which are identified using subspace identification. Published results concerning condition monitoring using subspace models include those of Abdelghani [92], who used a simulation of a rocket in flight to assess subspace-based monitoring algorithms, and Basseville [93], who reported that changes in the eigenstructure of linear dynamic systems could be used for structural vibration monitoring. An application study in which a state space model was used to detect process faults in the Tennessee Eastman process simulation was given in [94]. Other approaches have centred on the canonical variate analysis of Larimore [65]. Norvalis [95] presented a case study using a canonical variates analysis together with a knowledge based system to detect disturbances of a simulation of polymerization in a continuous stirred tank reactor. Simoglou [96] compared a CVA based procedure to PCA and PLS on the basis of a spray drying process and found similar fault detection rates for all three methods.

Shi & MacGregor [97] compared latent variable and subspace identification methods for modelling dynamic systems. They concluded that models obtained by subspace identification methods [19, 24, 25], were inferior to PCA and PLS as condition monitors because they focus only on predicting future values of the process outputs and provide little information concerning the covariance structure of the process inputs. One way to provide a more balanced model, where the emphasis is spread across both the inputs and the outputs, is to apply total least squares (TLS) approach [98, 99] to the system identification problem. The TLS approach developed in this chapter is closely related to

an error-in-variables (EIV) approach to the subspace system identification problem, and is aimed at overcoming the deficiencies described by Shi & MacGregor [97].

Yoon [100] provided a framework in which statistical and causal (dynamic) model-based approaches were compared with respect to fault detection and isolation. In this work, a parity relation approach based on the work of Gertler [101] was applied, to generate process residuals which were then used to detect and diagnose abnormal process behaviour. The parity relations can be derived from dynamic models in state space or ARX form. Secondary parity relations can also be created, so that the model residuals give clearer identification as to the origin of process faults. A CSTR simulation was used to show how causal and statistical models (e.g. PCA) have complementary strengths and weaknesses.

The application of dynamic models for condition monitoring relies on the identification of accurate process models, and has generally been considered unsuited for application to continuous processes with many process variables that are difficult to model accurately [100]. However subspace methods [19, 24, 25] have been shown to provide a means for the easy identification of state space models of processes with large numbers of process variables. For example a dynamic model of an industrial simulation of a Fluid Catalytic Cracking Unit with over 30 measured variables was successfully identified using subspace methods [102].

6.2 A subspace method for process monitoring

The first step to subspace monitoring involves the calculation of state sequences of the process, as described in Chapter 3. This is followed by the construction of a data matrix \mathbf{Z} which contains the process measurements and the state sequences. This is followed by the application of PCA to \mathbf{Z} , corresponding to an TLS solution of the subspace system identification problem. Finally a cross-validation procedure can be applied to determine the number of principal components, and the Hotelling's T^2 and Q statistics for the process.

In this section, a method for obtaining the state sequences based on the N4SID subspace system identification algorithm is presented. This leads on to the development of a subspace method for on-line process monitoring.

Subspace system identification determines the Kalman state sequences directly from the process measurements, as opposed to the classical modelling approach, where the state-space model is required in order to calculate the Kalman states [1]. The states are low dimension “principal directions” of the process. These are the key underlying dynamics that, either alone or in combinations, are driving a far greater number of the external variables. They possess the important property of being orthogonal to each other, so that each of the states can be monitored independently [95, 103]. Several algorithms have been proposed to (i) determine the state-sequences, and (ii) identify the state-space matrices, of which the CVA algorithm [18, 104], the MOESP algorithm [24] and the N4SID algorithm [25] are well referenced.

6.2.1 Calculating the state sequences

The procedure begins with auto-scaling of the process variables to zero mean and unit variance, then the measured predictor and response variables are arranged to form Hankel matrices \mathbf{Y}_f , \mathbf{Y}_p , \mathbf{U}_f , and \mathbf{U}_p as described in Eq. 3.40 and Eq. 3.41. Note that the subscripts “ f ” for future and “ p ” for past are used intuitively to describe the way the Hankel matrices are divided. The Hankel matrices are arranged to form a linear least squares regression equation, where the rows of \mathbf{Y}_f are regressed on the matrix $(\mathbf{Y}_p^T \quad \mathbf{U}_p^T \quad \mathbf{U}_f^T)^T$.

This leads to

$$\tilde{\mathbf{Y}}_f = (\mathbf{R}_{Y_p} \quad \mathbf{R}_{U_p} \quad \mathbf{R}_{U_f}) \begin{pmatrix} \mathbf{Y}_p \\ \mathbf{U}_p \\ \mathbf{U}_f \end{pmatrix}, \quad (6.1)$$

from which the matrix, $\hat{\mathbf{Y}}_f$ is calculated using the following regression equation:

$$\hat{\mathbf{Y}}_f = (\mathbf{R}_{Y_p} \quad \mathbf{R}_{U_p}) \begin{pmatrix} \mathbf{Y}_p \\ \mathbf{U}_p \end{pmatrix}, \quad (6.2)$$

where \mathbf{R}_{Y_p} and \mathbf{R}_{U_p} are regression components. The above matrices are used to form state-space equations giving rise to:

$$\begin{aligned}\mathbf{X}_{f+1} &= \mathbf{A}\mathbf{X}_f + \mathbf{B}\mathbf{U}_f + \mathbf{G}, \\ \mathbf{Y}_f &= \mathbf{\Gamma}\mathbf{X}_f + \mathbf{R}_{U_f}\mathbf{U}_f + \mathbf{E}\end{aligned}\tag{6.3}$$

where \mathbf{A} and \mathbf{B} are the state-space matrices, $\mathbf{\Gamma}$ is the extended observability matrix, \mathbf{R}_{U_f} is part of the regression coefficient, and \mathbf{X}_{f+1} and \mathbf{X}_f are the state sequences and \mathbf{G} and \mathbf{E} are residual matrices of the appropriate dimensions.

Comparing Eq. 6.2 and Eq. 6.3, the matrix product $\mathbf{\Gamma}\mathbf{X}_f$ is equivalent to $\hat{\mathbf{Y}}_f$, where the state sequence \mathbf{X}_f can be obtained by a singular value decomposition (SVD) as was demonstrated in equations Eq. 3.60 to Eq. 3.64, where the number of state-variables can be selected on the basis of the singular values. Note that the CVA algorithm [18] uses Akaike's Information Criterion to select the number of states, however, as the results in Chapter 4 confirm, the N4SID approach is better served by choosing the model order based on the relative magnitude of the singular values, then checking the results using cross-validation.

The determination of \mathbf{X}_f is followed by shifting the rows in the Hankel matrices where an extra time-step (block row) is included in \mathbf{Y}_p and \mathbf{U}_p ; the dimensions of \mathbf{Y}_p and \mathbf{U}_p are increased by one block row, and the dimensions of \mathbf{Y}_f and \mathbf{U}_f are reduced by one block row. The regression calculations Eq. 6.1 and Eq. 6.2 are then repeated to determine \mathbf{X}_{f+1} .

Online determination of the states involves shifting the rows in the Hankel matrices forward a time-step. This is done by augmenting the Hankel matrices with extra columns as described below.

6.2.1.1 Online updating of the state sequence

The updating of the state sequence is achieved by capturing a new data point, then augmenting the block Hankel matrices \mathbf{U}_p , \mathbf{U}_f , \mathbf{Y}_p , and \mathbf{Y}_f by adding a new column. An adaptive model can be calculated by also deleting the first column and recalculating the model using Eq. 3.55 and Eq. 3.64

$$\mathbf{M}_{X_k} = \mathbf{\Gamma}_r^t \mathbf{Y}_f (\mathbf{P}^T \mathbf{U}_f^T) \left(\begin{array}{cc} \mathbf{P}\mathbf{P}^T & \mathbf{P}\mathbf{U}_f^T \\ \mathbf{U}_f \mathbf{P}^T & \mathbf{U}_f \mathbf{U}_f^T \end{array} \right)^{-1}_{\text{first } (lr+mr) \text{ rows}} \quad (6.4)$$

The state sequence is calculated as

$$\mathbf{X}_k = (\mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_N) = \mathbf{M}_{X_k} \begin{pmatrix} \mathbf{U}_p \\ \mathbf{Y}_p \end{pmatrix} \quad (6.5)$$

i.e.

$$\mathbf{x}_N = \mathbf{M}_{X_k} \begin{pmatrix} \mathbf{u}_{N-r} \\ \vdots \\ \mathbf{u}_{N-1} \\ \mathbf{y}_{N-r} \\ \vdots \\ \mathbf{y}_{N-1} \end{pmatrix} \quad (6.6)$$

and the new state is calculated as

$$\mathbf{x}_{N+1} = \mathbf{M}_{X_k} \begin{pmatrix} \mathbf{u}_{N-r+1} \\ \vdots \\ \mathbf{u}_N \\ \mathbf{y}_{N-r+1} \\ \vdots \\ \mathbf{y}_N \end{pmatrix} \quad (6.7)$$

Based on the matrices \mathbf{X}_{f+1} , \mathbf{X}_f , \mathbf{Y}_f and \mathbf{U}_f the following least squares equation can be established:

$$\begin{pmatrix} \mathbf{X}_{f+1} \\ \mathbf{Y}_f \end{pmatrix} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix} \begin{pmatrix} \mathbf{X}_f \\ \mathbf{U}_f \end{pmatrix} + \begin{pmatrix} \mathbf{G} \\ \mathbf{E} \end{pmatrix} \quad (6.8)$$

In previous applications of subspace methods for process monitoring, such as in the application of Negiz and Cinar (1997) and Norvalis et al (2000), the state variables have been used directly to define a Hotelling's T^2 statistic [94, 95, 103]. However, because the state-variables are determined to best fit the response variables, the variation of the predictor variables may not be adequately represented in this statistic, [97].

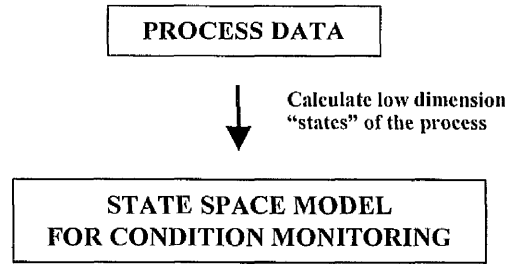


Figure 6.1 Summary of the state space modelling approach of Negiz & Cinar (1997) and Norvilas et al. (2000) where the state sequences are used to calculate Hotelling's T^2 and Q statistics which are then used to monitor the process.

6.3 The Subspace method

The subspace method uses PCA to develop a model for process condition monitoring. The state sequences in Eq. 6.8 are first scaled to unit variance; this does not affect the input-output properties of the model, however it ensures that the data is in the proper form for the PCA analysis that follows. PCA is then applied, where the important variation of the predictor, the response and the state variables is measured by first determining the principal components of the process, and then calculating the Hotelling's T^2 statistic. It is demonstrated below that the subspace method is equivalent to the total least squares (TLS) solution to the subspace system identification problem, and is also closely related to using an error-in-variables (EIV) approach as discussed by [38, 105, 106].

For application of subspace methods to process monitoring, the state variables are usually employed to define a Hotelling's T^2 statistic [94, 95, 103]. However, Shi & MacGregor [97] indicated that since the state variables are determined to fit the response variables, the variation of the predictor variables may not be adequately represented in this statistic. The TLS approach [98, 99] provides a more balanced model, with emphasis on both the predicted and predictor variables. Based on the matrices \mathbf{X}_{k+1} , \mathbf{X}_k , \mathbf{Y}_k and \mathbf{U}_k , the following least squares equation is established:

$$\begin{pmatrix} \mathbf{X}_{k+1} \\ \mathbf{Y}_k \end{pmatrix} = \begin{pmatrix} \mathbf{A}_{tls} & \mathbf{B}_{tls} \\ \mathbf{C}_{tls} & \mathbf{D}_{tls} \end{pmatrix} \begin{pmatrix} \mathbf{X}_k \\ \mathbf{U}_k \end{pmatrix} + \begin{pmatrix} \mathbf{G} \\ \mathbf{W} \end{pmatrix}. \quad (6.9)$$

The predicted variables $[\mathbf{X}_{k+1}^T \quad \mathbf{Y}_k^T]$ and predictor variables $[\mathbf{X}_k^T \quad \mathbf{U}_k^T]$ are used to construct the data matrix $\mathbf{Z} = [\mathbf{X}_{k+1}^T \quad \mathbf{Y}_k^T \quad \mathbf{X}_k^T \quad \mathbf{U}_k^T]^T$. The TLS solution is found by applying PCA to calculate $\mathbf{Z} = \mathbf{T}_M \mathbf{\Phi}_M^T + \mathbf{E}$, where $M \leq r + m$ is the number of principal components retained in the model, and $\mathbf{\Phi}_M$ forms the basis for the condition monitoring model and \mathbf{E} measures the distance between the process measurements and the process model.

An important property of state space models is that the state basis can be changed by any non-singular matrix $\mathbf{T} \in \mathbb{R}^{n \times n}$, without affecting the input and output measurements of the system [55]. Therefore, by applying the transformation

$$\bar{\mathbf{X}}_k = \mathbf{T} \mathbf{X}_k \quad (6.10)$$

the same input-output relationship is obtained with $\bar{\mathbf{A}}_{TLS} = \mathbf{T} \mathbf{A}_{TLS} \mathbf{T}^{-1}$; $\bar{\mathbf{B}}_{TLS} = \mathbf{T} \mathbf{B}_{TLS}$; $\bar{\mathbf{C}}_{TLS} = \mathbf{C}_{TLS} \mathbf{T}^{-1}$ and $\bar{\mathbf{D}}_{TLS} = \mathbf{D}_{TLS}$.

The following transformation scales the states to zero mean, unit variance:

$$\mathbf{T} = \begin{pmatrix} \lambda_1^{-1} & 0 & 0 & \dots & 0 \\ 0 & \lambda_2^{-1} & 0 & \dots & \vdots \\ 0 & 0 & \ddots & 0 & \vdots \\ \vdots & \vdots & 0 & \ddots & \vdots \\ 0 & \dots & \dots & 0 & \lambda_n^{-1} \end{pmatrix}, \quad (6.11)$$

where λ_i is the normal operation variance of the i^{th} state sequence. Since the states are zero mean due to the initial auto-scaling of the process data, the transformation

$\bar{\mathbf{X}}_k = \mathbf{T} \mathbf{X}_k$ corresponds to an autoscaling where

$$\bar{x}_{i,k} = \frac{1}{\lambda_i} (x_{i,k}), \quad (6.12)$$

and where

$$\lambda_i = \frac{1}{N} \sum_{k=1}^N (x_{i,k})^2. \quad (6.13)$$

For a stationary process with long data sequences ($N \gg 100$), the variance λ_i of $\mathbf{X}_{i,k}$ and $\mathbf{X}_{i,k+1}$ are approximately equal, i.e. $\lambda_{X_{i,k}} \cong \lambda_{X_{i,k+1}}$. Therefore the same transformation matrix \mathbf{T} can be applied to \mathbf{X}_k and \mathbf{X}_{k+1} leading to $\mathbf{Z} = [\bar{\mathbf{X}}_{k+1}^T \quad \mathbf{Y}_k^T \quad \bar{\mathbf{X}}_k^T \quad \mathbf{U}_k^T]^T$ where the columns of \mathbf{Z} are zero mean and unit variance (since the process data is pre-scaled before identification begins).

The scaling of the state sequences leads to a much larger contribution from the states in the SVD calculation of the principal directions. Note also that the vectors contained in each state sequence are orthogonal, meaning that each contributes to the calculation of the principal directions independently.

The important variation of the predictor, the response and the state variables is measured by calculating the Hotelling's T^2 statistic. Combining Eq. 6.9 and Eq. 6.10, the state-space matrices are obtained as follows

$$\begin{pmatrix} \bar{\mathbf{X}}_{k+1} \\ \mathbf{Y}_k \\ \bar{\mathbf{X}}_k \\ \mathbf{U}_k \end{pmatrix} - \begin{pmatrix} \bar{\mathbf{X}}_{k+1}^I \\ \mathbf{Y}_k^I \\ \bar{\mathbf{X}}_k^I \\ \mathbf{U}_k^I \end{pmatrix} = \begin{pmatrix} \bar{\mathbf{G}} - \bar{\mathbf{B}}_{TLS} \mathbf{F} + \bar{\mathbf{X}}_{k+1}^{II} \\ \mathbf{W} - \bar{\mathbf{D}}_{TLS} \mathbf{F} + \mathbf{Y}_k^{II} \\ \bar{\mathbf{X}}_k^{II} \\ \mathbf{U}_k^{II} - \mathbf{F} \end{pmatrix} \quad (6.14)$$

where the matrices $\bar{\mathbf{X}}_{k+1}^I$, \mathbf{Y}_k^I , $\bar{\mathbf{X}}_k^I$, \mathbf{U}_k^I are the linear model of the significant variation of the process, and the matrices $\bar{\mathbf{X}}_{k+1}^{II}$, \mathbf{Y}_k^{II} , $\bar{\mathbf{X}}_k^{II}$, \mathbf{U}_k^{II} represent the insignificant variation (or model mismatch) of $\bar{\mathbf{X}}_{k+1}$, \mathbf{Y}_k , $\bar{\mathbf{X}}_k$, \mathbf{U}_k .

Defining the following:

$$\mathbf{Z} = \begin{pmatrix} \bar{\mathbf{X}}_{k+1} \\ \mathbf{Y}_k \\ \bar{\mathbf{X}}_k \\ \mathbf{U}_k \end{pmatrix}^T, \quad \hat{\mathbf{Z}} = \begin{pmatrix} \bar{\mathbf{X}}_{k+1}^I \\ \mathbf{Y}_k^I \\ \bar{\mathbf{X}}_k^I \\ \mathbf{U}_k^I \end{pmatrix}^T, \quad \Phi = \begin{pmatrix} \Phi_{\bar{\mathbf{X}}_{k+1}} \\ \Phi_{\mathbf{Y}} \\ \Phi_{\bar{\mathbf{X}}_k} \\ \Phi_{\mathbf{U}} \end{pmatrix} \quad \text{and} \quad \mathbb{Z} = \begin{pmatrix} \mathbf{G} - \mathbf{B}\mathbf{F} + \bar{\mathbf{X}}_{k+1}^{II} \\ \mathbf{E} - \mathbf{D}\mathbf{F} + \mathbf{Y}_k^{II} \\ \bar{\mathbf{X}}_k^{II} \\ \mathbf{U}_k^{II} - \mathbf{F} \end{pmatrix}^T.$$

then,

$$\mathbf{Z} = \hat{\mathbf{Z}} + \mathbb{Z} = \mathbf{T}\Phi^T + \mathbb{Z} \quad (6.15)$$

with $\mathbf{T} = \mathbf{Z}\Phi$ and $\hat{\mathbf{Z}} = \mathbf{Z}\Phi\Phi^T$ and where $\Phi_{\bar{\mathbf{x}}_{k+1}}$, $\Phi_{\mathbf{y}}$, $\Phi_{\bar{\mathbf{x}}_k}$ and $\Phi_{\mathbf{u}}$ are equivalent to PCA loading matrices.

Eq. 6.15 corresponds to a PCA analysis of an augmented matrix containing the process inputs and outputs, and the scaled states of the process, i.e.

$$\mathbf{Z} = (\mathbf{Z}_{PCA} \quad \bar{\mathbf{x}}_k \quad \bar{\mathbf{x}}_{k+1}) \quad (6.16)$$

6.3.1 Multivariate Statistics for the Subspace method

$\mathbf{Z} = \mathbf{T}\Phi^T + \mathbb{Z}$ has been shown to correspond to the TLS solution to the subspace system identification problem, where the first M principal components are used to calculate the statistical model and the dynamics of the system. PCA cross-validation (PCA_{CV}) is applied to estimate the number of principal components (M) to retain in the analysis, leading to

$$(\bar{\mathbf{x}}_{k+1}^T \quad \mathbf{y}_k^T \quad \bar{\mathbf{x}}_k^T \quad \mathbf{u}_k^T) = \mathbf{T}_M \Phi_M^T + \mathbb{Z} \quad (6.17)$$

where the M columns of the the score matrix \mathbf{T}_M are orthogonal and the M rows of the loading matrix Φ_M^T are orthonormal. $\hat{\mathbf{Z}}_M = \mathbf{T}_M \Phi_M^T$ defines normal operation and \mathbb{Z} is the orthogonal residual space

$$\mathbb{Z} = \mathbf{T}_{res.} \Phi_{res.}^T = \sum_{i=M+1}^{2n+l+m} \mathbf{T}_i \Phi_i^T, \quad (6.18)$$

where n is the order of the state space model and l and m are the number of process outputs and inputs respectively.

The subspace method model is first trained on reference data, where the common-cause variation lying in the M -dimensional space is defined as $\hat{\mathbf{Z}} = \mathbf{T}_M \Phi_M^T$. The co-ordinates of the k^{th} sample point of the monitored process are then determined as:

$$\mathbf{T}_k = \Phi_M^T \zeta_k \quad (6.19)$$

where T_k is a measure of the variability that is incorporated in the sample point ζ_k . ζ_k is projected onto the principal directions defined by Φ_M , the loading matrix, and T_k contains the coordinates that measure the projection. ζ_k is a column vector in \mathbf{Z}^T in which the values of the k^{th} process measurements and associated state sequences are stored, i.e.

$$\zeta_k = (\bar{x}_{k+1}^T \quad y_k^T \quad \bar{x}_k^T \quad u_k^T)^T \quad (6.20)$$

The residuals of the model are used to calculate the squared prediction error (SPE) which is the mismatch between the measured and reconstructed values of the process variables:

$$\varepsilon_k = \zeta_k - T_k \Phi_M^T = \zeta_k (\mathbf{I}_\zeta - \Phi_M \Phi_M^T) \quad (6.21)$$

where M denotes the number of latent variables included in the subspace method analysis and \mathbf{I}_ζ is an identity matrix of dimension $\zeta = 2n + l + m$. The model residual vector, ε_k , is used to provide a measure of the likelihood that the process has strayed beyond the bounds of normal and acceptable operation. If the value of ε_k goes beyond predefined confidence limits, the process is deemed “out of control”, then a fault or other cause can be investigated.

6.3.2 Calculation of Hotelling's T^2 Statistic

Given that $\mathbf{Z}_{SUBSPACE}$ is a matrix containing the estimated state sequences, the predictor and the predicted variables, it is possible to create a Hotelling's T^2 statistic that is calculated as

$$SM - T_k^2 = T_k^T \Lambda_M T_k \quad (6.22)$$

where T_k is a column vector containing the T scores of the k^{th} sample point, determined using Eq. 6.19, and Λ_M is a diagonal matrix containing the normal operation variance of each of the M columns of \mathbf{T} .

The contribution of individual variables to the SM- T^2 statistic can be determined by separating the contribution of each of the individual variables, i.e. the state variables \mathbf{X}_{k+1} and \mathbf{X}_k , the response variables Y_f and the predictor variables U_f .

$$_{T^2} \tilde{\mathbf{C}}_Y = \Phi_Y \Lambda_{M_Y} \Phi_Y^T Y, \quad (6.23)$$

$$_{T^2} \tilde{\mathbf{C}}_U = \Phi_U \Lambda_{M_U} \Phi_U^T U, \quad (6.24)$$

$$_{T^2} \tilde{\mathbf{C}}_{\bar{X}_{k+1}} = \Phi_{\bar{X}_{k+1}} \Lambda_{M_{X_{k+1}}} \Phi_{\bar{X}_{k+1}}^T \bar{X}_{k+1}, \quad (6.25)$$

$$_{T^2} \tilde{\mathbf{C}}_{\bar{X}_k} = \Phi_{\bar{X}_k} \Lambda_{M_{X_k}} \Phi_{\bar{X}_k}^T \bar{X}_k, \quad (6.26)$$

or a single vector of contributions can be calculated,

$$_{T^2} \tilde{\mathbf{C}}_\zeta = \Phi_M \Lambda_M \Phi_M^T \zeta_k. \quad (6.27)$$

Λ_M is a diagonal matrix where the M diagonal entries of Λ_M contain the inverse values of the variance of the M principal components calculated from the normal operation data.

Contribution plots [107, 108] provide an aid for assessing the contribution of individual variables to the statistics that describe a special event. The contribution plots for the simulation studies combine the individual contributions into a single plot, i.e. the contributions of the response, predictor and state variables are read from a single chart, for example, as in Figure 7.7. Note that a similar approach is applied for the DPCA analysis, however, the number of contributions from the time-series model structures employed in DPCA is significantly higher. This is further demonstrated on the basis of the simulation study in Chapter 7.

The confidence limits for the T^2 statistics can be obtained as discussed by Jackson [109]

$$T_C^2(M, K, \alpha) = \frac{M(K-1)}{K-M} F_{M, K-M, \alpha}, \quad (6.28)$$

where M is the number of degrees of freedom, K is the number of recorded samples of the reference data set, α is the confidence limit, (typically 95% or 99%), and $F_{M,K-M,\alpha}$ is the value representing the confidence limit of an F-distribution.

6.3.3 Calculation of the Q Statistics

The diagnosis of abnormal behaviour may involve the determination of the contribution of individual process variables to the violation of Q statistic confidence limits. The stronger the contribution of a particular variable, the greater the likelihood that this variable is affected by an anomalous event. Given that $\mathbf{Z}_{SUBSPACE}$ contains the estimated state sequences and the predicted and predictor variables, it is possible to create three univariate statistics based on the Q statistic.

The $Q^{(Y)}$ statistic for the response variables, also known as the squared prediction error (SPE) statistic measures the distance from the model to the measured outputs, and a $Q^{(U)}$ statistic is a measure of the Euclidean distance from the measured to the predicted values for the input space:

$$SM - Q_k^{(Y)} = \varepsilon_{k,Y}^T \varepsilon_{k,Y}, \quad (6.29)$$

$$SM - Q_i^{(U)} = \varepsilon_{k,U}^T \varepsilon_{k,U}. \quad (6.30)$$

A third $Q^{(X)}$ statistic can be established by measuring the distance between the state space model state sequences, $X^{(m)}$ and condition monitor states $X^{(p)}$.

$$SM - Q_i^{(X)} = \varepsilon_{k,X}^T \varepsilon_{k,X}, \quad (6.31)$$

where

$$\varepsilon_{x_k} = X_k^{(m)} - X_k^{(p)}, \quad (6.32)$$

$$X_k^{(m)} = \mathbf{T} \mathbf{M}_{X_k} \mathbf{P}, \quad (6.33)$$

$$X_k^{(p)} = \Phi_{X_k} \Lambda_{M_{X_k}} \Phi_{X_k}^T X_k^{(m)}. \quad (6.34)$$

The same relations apply for $X_{k+1}^{(m)}$, where

$$X_{k+1}^{(m)}(k) = \mathbf{M}_{X_{k+1}} \mathbf{P}^{(+)} , \quad (6.35)$$

$$X_{k+1}^{(p)} = \mathbf{\Phi}_{X_{k+1}} \mathbf{\Lambda}_{M_{X_{k+1}}} \mathbf{\Phi}_{X_{k+1}}^T X_{k+1}^{(m)} , \quad (6.36)$$

where $\mathbf{P} = \begin{pmatrix} \mathbf{U}_p^T & \mathbf{Y}_p^T \end{pmatrix}^T$ as in Eq. 3.51. and $\mathbf{P}^{(+)}$ is constructed as in Eq. 3.62.

A reduction in the number of Q statistic charts can be achieved by combining $Q^{(U)}$, $Q^{(X)}$ and $Q^{(Y)}$ as follows

$$SM - Q_i^{\zeta} = \mathbf{\varepsilon}_k \mathbf{\varepsilon}_k^T , \quad (6.37)$$

where $\mathbf{\varepsilon}_k$ is calculated as in Eq. 6.21.

The vector of the k^{th} contributions to the Q statistic for the state, predictor and predicted variables is calculated as

$${}_{(Q)}\tilde{\mathbf{C}}_{\zeta,k} = \mathbf{\varepsilon}_k . \quad (6.38)$$

The confidence limit for the Q statistic can be calculated using Jackson and Mudholkar [110], alternatively the the approach by Box [111] can be applied. Given a statistic, $\varphi \geq 0$, which represents a sum of squared variables, along with its mean value, μ_{φ} and its variance, σ_{φ} , the confidence limit, φ_{α} , representing a confidence of α is as follows:

$$\varphi_{\alpha} = \rho \chi^2(\alpha, h) , \quad (6.39)$$

where $\rho = \frac{\sigma_{\varphi}}{2\mu_{\varphi}}$, $h = \frac{2\mu_{\varphi}^2}{\sigma_{\varphi}}$ and χ^2 denotes the Chi-Squared density function.

6.3.4 Determining the number of principal components

Several methods have been suggested for determining the number of principal components to be retained when applying PCA analysis [109]. One method, assuming the use of autoscaled data, is to retain components with corresponding eigenvalues

greater than one [109], or alternatively parallel analysis can be applied [91], however due to the idiosyncratic nature of industrial data sets, for example where non-linear operating regions may be encountered, it is expected that no one method is foolproof, therefore several methods might be applied in an attempt to reach consensus [91].

In this study PCA cross-validation (PCA_{CV}) is applied to estimate the number of principal components M , and hence the subspace method triplet (n, r, M) , where n is the system order, r is the number of block rows and M is the number of latent variables used to monitor the process. Various user choices are available for the Matlab PCA cross-validation algorithm; two methods were considered:

- (1) PCA_{CV1} : First divide the data into a finite number of blocks and then calculate the prediction residual error sum of squares (PRESS) statistic for each of data splits.
- (2) PCA_{CV2} : Leave-one-out cross-validation based on the entire training data set. For a given N data points this involves choosing $N-1$ points (hence leave-one-out) and then calculating the prediction error for the final point. The procedure is repeated N times and the results averaged before the optimum model is decided upon.

6.3.5 Determination of the Subspace Model Structure

The state matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ are not required explicitly in the subspace condition monitor, however they may be useful for the determination of the triplet (n, r, M) . For example, the use of a TLS approach to subspace system identification often leads to an unstable model being identified, however if a stable model were required, different values for r could be tried and the eigenvalues of \mathbf{A} could be calculated to check for stability.

The model order is determined by the singular values of the 4SID projection, then cross-validation is used to determine r by calculating the mean squared prediction error (MSPE) for a given model order. Finally PCA_{CV} is applied to $\mathbf{Z}_{SUBSPACE}$ to determine the number of latent variables in the model. The subspace method is therefore comprised of two dimension reduction steps, as shown in Figure 6.2.

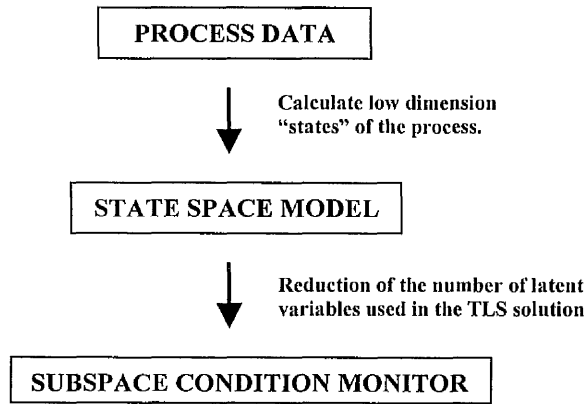


Figure 6.2 Summary of the subspace method in which two dimension reduction steps are used to develop the condition monitoring model.

6.3.6 Subspace Condition Monitoring Procedure

The subspace method begins with determination of the input-output structure to be used in the state space model. This may involve applying N4SID to a variety of input/output configurations to gauge the cause-effect relationships in the data. It is likely that some process variables will not be suitable for inclusion due to observability/controllability considerations, or process non-linearities. The next stage involves preprocessing the data, removal of outliers and scaling the measurements to be included in the analysis, to unit variance and zero mean. This is followed by a determination of the model structure i.e. the triplet (n, r, M) , where n is the state space model order, r is the number of block rows included in the N4SID calculations and M is the number of principal components used by the condition monitor. Consider first the eigenvalue plot which gives a good indication of the likely model order. Alternatively apply a subspace system identification algorithm such as N4SID [25] and determine the order of dynamics based on the infinite horizon predictions, as was outlined in Chapters 4 and 5. Then, for a given model order, iterate through values of r to determine an appropriate number of block rows to use when calculating the state sequences. As an example, this may involve the calculation of the eigenvalues of the resulting A matrix, if a stable model is required, or could be evaluated on the basis of the MSPE of the resulting model. With (n, r) decided upon, apply PCA_{CV} to determine the number of

principal components, M_{opt} , to retain in the model. Once the triplet (n, r, M) has been identified, an online implementation proceeds as explained in Table 6.1 below.

STEP	DESCRIPTION	EQUATION	Eq.
1	k^{th} observation becomes available	$(y_k \quad u_k)$	
2	Determine the state variables	$X_k^{(m)} = \mathbf{T} \mathbf{M}_{X_k} \mathbf{P}$	6.33
		$X_{k+1}^{(m)} = \mathbf{T} \mathbf{M}_{X_{k+1}} \mathbf{P}^{(+)}$	6.35
3	Scale the state variable	$\bar{\mathbf{X}}_k = \mathbf{T} \mathbf{X}_k$	6.10
4	Construct data vector	$\mathbf{Z} = (\bar{\mathbf{X}}_{k+1}^T \quad \mathbf{Y}_k^T \quad \bar{\mathbf{X}}_k^T \quad \mathbf{U}_k^T)$	6.16
5	Compute values of the score variables	$T_k = \Phi_M^T \zeta_k$	6.19
6	Calculate variable residuals	$\varepsilon_k = \zeta_k - T_k \Phi_M^T$	6.21
7	Determine T^2 and Q statistics	$T_k^2 = T_k^T \Lambda_M T_k$	6.22
		$Q_k = \varepsilon_k \varepsilon_k^T$	6.37
8	Compute contributions	$_{T^2} \tilde{\zeta}_\zeta = \Phi_M \Lambda_M \Phi_M^T \zeta_k$	6.27
		$_{(Q)} \tilde{\zeta}_{\zeta,k} = \varepsilon_k$	6.38
9	Update Monitoring Chart		
10	Go to StepOne	$k = k + 1$	

Table 6.1 On Line implementation of the Subspace Method.

The above table outlines the online condition monitoring procedure, where the models of normal operation, $\mathbf{T} \mathbf{M}_{X_k}$, $\mathbf{T} \mathbf{M}_{X_{k+1}}$, Λ_M , and Φ_M are calculated using a reference data set.

6.4 Conclusion

In this chapter, a new dynamic method for process condition monitoring has been outlined. The model uses the state space model structure, with state sequences identified using subspace methods. The states are then scaled to unit variance, then incorporated into a PCA analysis alongside the process variables. This has been shown to be equivalent to a PCA analysis of an augmented process data matrix that contains information about the dynamics of the process. The method has also been shown to correspond to the error-in-variables (EIV) solution to the subspace system identification problem.

An explicit procedure has been suggested for optimising the structure of the subspace monitor on the basis of the user choices (n, r, M) . Finally, a procedure for online implementation of the subspace method has been outlined.

The EIV approach to the subspace system identification problem involves the application of TLS to solving the state equations. The subsequent scaling of the states is followed by the application of PCA, which implies that the resulting projections give rise to score variables that contain variations of both, the predictor and response variables. The score variables form the basis for a T^2 statistic, whereas the residuals of the TLS model have been utilised to construct a Q statistic.

The Q statistic is used to measure the likelihood that the process has strayed beyond the bounds of normal and acceptable operation. The model residuals can also be used to generate a contributions analysis, that is used to diagnose the possible cause of abnormal operation.

The subspace method brings a dynamic aspect to the data, without the need for incorporating large numbers of time-shifted process variables into the contributions analysis (as with DPCA and DPLS). It will be demonstrated in the next chapters that this leads to a more simplified analysis involving fewer contributions to the monitoring statistics.

Chapter 7

Dynamic models for MSPC

The subspace method is compared with other linear dynamic modelling methods on the basis of a continuous stirred tank reactor simulation, where its performance is found to compare favourably. PCA, DPCA and PLS modelling procedures are outlined. The connection between the subspace method and DPCA is made, where both are shown to correspond to a total least squares (TLS) solution for linear systems. The Subspace Method is found to provide a condition monitor using fewer dimensions than DPCA.

7.1 Introduction

In Chapter 6, a linear dynamic model with a state space model structure was developed using subspace methods. The subspace method was cast into a MSPC framework where Hotelling's T^2 and Q statistics were defined. In this chapter, two further dynamic models are developed, using PCA and PLS. These models are based on an ARX model structure and form a basis for a comparison with the subspace method. A continuous stirred tank reactor (CSTR) process simulation [100] is used to create three process faults so that the subspace method is compared and contrasted with (i) a dynamic PCA approach and (ii) a dynamic PLS approach to dynamic modelling for multivariate statistical process control. The dynamic PCA (DPCA) method is currently installed in monitorMV and therefore provides a yardstick for the utility of the subspace method.

The chapter begins with a description of PCA and PLS modelling techniques and then the development of dynamic counterparts for each.

The objective of using dynamic models is to accommodate dynamic process behaviour such as throughput changes; and auto-correlation of the process variables caused by controller feedback. The following model structures are used to identify faults in the CSTR process:

- A state space approach using the subspace method developed in Chapter 6.
- The DPCA approach based on an ARX model structure (a technique currently available in the monitorMV software package).
- A PLS approach based on an ARX model structure.

In the last part of the chapter, a simple open-loop system, driven by autocorrelated inputs, is simulated to demonstrate an advantage enjoyed by dynamic models over (static) PCA. The behaviour of each of the models is considered for when an unmeasured disturbance enters the system, where it is suggested that the PCA model wrongly diagnoses a process fault. In contrast, it is suggested that the subspace and DPCA methods correctly indicate that there is no fault in the system.

7.2 Multivariate Statistical Process Control

In the next sections, a brief summary of the subspace method is followed by an overview of PCA, PLS and their dynamic counterparts, including the definition of statistics and contributions charts. This is followed by the CSTR application study.

7.2.1 Subspace Method

As outlined in Chapter 6, the subspace method overcomes the drawbacks of previous subspace monitoring approaches [97], by using TLS to deliver an error-in-variable (EIV) approach [112, 113] to the subspace system identification problem. TLS is employed to calculate a state space model, after the state sequences have been identified using an N4SID approach. The state sequences are then scaled to unit variance, so that the problem reduces to a PCA analysis, where inclusion of the states means that both

the dynamic and static information from the system is used to build the model. The TLS solution corresponds to the application of PCA to obtain a reduced dimension model of the important variance in the data, and the associated process condition monitoring statistics.

7.2.2 Principal Components Analysis

The use of PCA to build low dimensional models for analysis and monitoring of process operations is well documented, e.g. [76, 80, 81, 114-118]. PCA analysis begins with the process data being collected into a single data matrix \mathbf{Z} . The highly correlated nature of process data means that a rank reduction step via SVD yields a low dimension model that describes nearly all of the variation in the original data set. PCA is scale dependent [76] so that a scaling needs to be applied; typically each vector of measurements is scaled to zero mean and unit variance. It is important that a zero mean scaling is used, because PCA models the covariance relationships in the data. The matrix \mathbf{Z} is then decomposed as follows

$$\mathbf{Z} = \mathbf{T}\mathbf{\Phi}^T, \quad (7.1)$$

where \mathbf{T} (the score matrix) is orthogonal and $\mathbf{\Phi}$ (the loading matrix) is orthonormal, and the rows of \mathbf{T} and $\mathbf{\Phi}$ are the eigenvectors of $\mathbf{Z}^T\mathbf{Z}$ and $\mathbf{Z}\mathbf{Z}^T$ respectively [107, 119].

A cross-validation stopping criterion can be used to determine the number of principal components, N , that capture most of the relevant variability of the process:

$$\mathbf{Z} = \sum_{i=1}^N \mathbf{T}_i \mathbf{\Phi}_i^T + \mathbf{E}, \quad (7.2)$$

where $\hat{\mathbf{Z}}_N = \mathbf{T}_N \mathbf{\Phi}_N^T$ represents the common-cause variations lying in the N -dimensional subspace spanned by the first N principal components and \mathbf{E} is the orthogonal space of the residuals. The residual space is expressed as

$$\mathbf{E} = \mathbf{T}_r \mathbf{\Phi}_r^T = \sum_{i=N+1}^{n_z} \mathbf{T}_i \mathbf{\Phi}_i^T, \quad (7.3)$$

where n_ζ is the number of process variables in the analysis, and \mathbf{T}_r and $\mathbf{\Phi}_r$ are a matrix pair that define the residual space. It is assumed that the significant process variation is described by $\hat{\mathbf{Z}}_N$, so the base vectors of this subspace are considered to be the principal directions of the process. The model of the process, i.e. the matrix containing the base vectors, $\mathbf{\Phi}_N$, is determined using a reference data set which is drawn from historical data [76, 79]. Using this model, the co-ordinates of the k^{th} sample point of the online monitor are determined as

$$\tau_k = \zeta_k \mathbf{\Phi}_N, \quad (7.4)$$

where τ_k is a row vector storing the co-ordinates of the score which measures the magnitude of the projection of the k^{th} sample point in the direction of $\mathbf{\Phi}_N$. $\mathbf{\Phi}_N$ is also referred to as the loading matrix, in which the base vectors of the subspace are stored as column vectors; and ζ_k is a row vector in which the values of the process variables at the k^{th} sample point are stored.

The dimension of the subspace, N , is typically determined using cross-validation [120, 121]. However, a variety of alternative approaches are also available, for example, Jackson [109] and Valle [122].

The latent variables model is based only on the “significant” process variation and consequently, the process variables cannot be reconstructed using this variable set. The mismatch between the measured and reconstructed values of the process variables, ε_k , is then:

$$\varepsilon_k = \zeta_k - \tau_k \mathbf{\Phi}_N^T = \zeta_k \left(\mathbf{I}_{n_\zeta} - \mathbf{\Phi}_N \mathbf{\Phi}_N^T \right), \quad (7.5)$$

where n_ζ denotes the number of process variables included in the PCA analysis and \mathbf{I}_{n_ζ} is an $n_\zeta \times n_\zeta$ identity matrix. A more detailed analysis of PCA may be found in Jolliffe [123] or Wold [120].

7.2.2.1 PCA Univariate Statistics

Given the above discussion, two univariate statistics can be established for on-line process monitoring, the Hotelling's T^2 statistic and the Q statistic:

$$PCA-T_k^2 = \tau_k \Lambda_N^{-1} \tau_k^T, \quad (7.6)$$

$$PCA-Q_k = \varepsilon_k \varepsilon_k^T, \quad (7.7)$$

where Λ_N is a diagonal matrix storing the variance of the score variables of the reference data set. Λ_N rescales the normal operation principal components to have unit variance as suggested by Jackson [117], which is done by dividing the eigenvectors by the square roots of their corresponding eigenvalues.

7.2.2.2 PCA contribution calculations

The vector ε_k , calculated in Eq. 7.5, is a measure of the distance of the process measurements from the model of “normal operation”. If any of its values stray beyond the predefined confidence limit for normal operation, the process could be “out of control”, and a fault or other cause can be investigated. The T^2 statistic and Q statistic contribution calculations for PCA can be summarised as follows:

$$_{PCA} \mathbb{C}_{T^2} = \mathbf{Z} \Phi_N \Lambda_N^{-1} \Phi_N^T \quad (7.8)$$

$$_{PCA} \mathbb{C}_Q = \mathbf{Z} \left(\mathbf{I}_{n_z} - \Phi_N \Phi_N^T \right) \quad (7.9)$$

The residuals of the PCA model can be plotted in a bar chart to represent the contribution of individual variables to the PCA- Q statistic [81]. Alternatively they can be plotted against time as has been done for the CSTR simulation results, see for example Figures 7.7 and 7.9.

The contribution plots show the contribution of individual variables to the T^2 and Q statistics that describe a special event. The contribution of the j^{th} process variable to the i^{th} T score is:

$$T^2 C_{i,j}^{(k)} = \tilde{\zeta}_j^{(k)} \phi_{i,j} \frac{\tau_i^{(k)}}{\lambda_i}, \quad (7.10)$$

where $\tau_i^{(k)}$ and λ_i represent the k^{th} value and the normal operation variance of the i^{th} score variable respectively, $\phi_{i,j}$ is the element in the i^{th} row and the j^{th} column of the matrix Φ_N , and $\tilde{\zeta}_j^{(k)}$ is the k^{th} value of the scaled j^{th} process variable. Eq. 7.10 shows how the contribution of individual process variables can be determined for a particular score variable, where the overall contribution to $\tau_i^{(k)}$ is

$$\tau_i^{(k)} = \sum_{j=1}^{n_\zeta} T^2 C_{i,j}^{(k)}, \quad (7.11)$$

where n_ζ is the total number of process variables to be included in the analysis.

Each of the univariate statistics can be plotted with a time base and confidence limits. The confidence limits for the T^2 statistics can be obtained as discussed by Jackson [109]:

The confidence limits for the T^2 statistics can be obtained as discussed by Jackson [109]:

$$T_{n_{PC1}, \beta, \alpha}^2 = \frac{\dots}{\beta - n_{PCA}} F_{n_{PC1}, \beta - n_{PC1}, \alpha}, \quad (7.12)$$

where $T_{n_{PC1}, \beta, \alpha}^2$ is the value representing the confidence limit, β is the number of recorded samples of the reference data set, α is the confidence limit, typically 95% or 99%, and $F_{n_{PC1}, \beta - n_{PC1}, \alpha}$ is the value representing the confidence limit of an F-distribution [135]. The confidence limit for the Q statistic can be calculated according to the approach by Box [111]: Given the sum of squared variables $\varphi \geq 0$, along with its mean value, μ_φ and its variance, σ_φ , the confidence limit, φ_α , representing a confidence of α is as follows:

$$\varphi_\alpha = \rho \chi^2(\alpha, h), \quad (7.13)$$

where $\rho = \frac{\sigma_\varphi}{2\mu_\varphi}$, $h = \frac{2\mu_\varphi^2}{\sigma_\varphi}$ and χ^2 denotes the Chi-Squared density function.

7.2.3 Partial Least Squares

Partial Least Squares (PLS) uses a reduced number of latent variables to describe the relations that exist among the process variables. The process data is first arranged into two blocks; the X-Block contains the “independent” variables, here called the predictor variables, where for dynamic models this block also includes time-lagged process variables as required, for example, by an ARX model structure. The Y-Block contains the “dependent” variables, here called the predicted variables. PLS establishes independent subspaces for each of these variable sets.

PLS is an iterative approach in which each iteration step results in the determination of a pair of score variables. Höskuldsson [124] showed that these τ and ν score variables are determined to maximise their covariance. In order to determine the score variables along with the PLS weight and loading vectors, a variety of algorithms have been proposed such as the NIPALS [125] and the SIMPLS [126] algorithms. A detailed analysis of PLS algorithms is presented in [124-126].

The model is identified using a reference data set drawn from the process under normal operating conditions. This defines normal operation against which the incoming process data is compared. This reference data set is used to produce the PLS model which comprises two matrices: \mathbf{R}_n and \mathbf{Q}_n are matrices in which the base vectors of the subspaces for the predictor and the predicted variables are stored as column vectors, and n is the dimension of both subspaces [84, 119, 127]. The prediction of the ν -score variable is determined by the “inner model”, which is a regression matrix for each ν -score and τ -score pair. This matrix \mathbf{B} is a diagonal matrix in which the regression coefficient for each pair of score variables are stored [125].

The projection of the k^{th} sample of the incoming process data, containing the predictor variables ξ_k and the predicted variables ψ_k onto their respective subspaces is as follows:

$$\mathbf{\tau}_k = \mathbf{R}^T \xi_k, \quad (7.14)$$

$$\mathbf{v}_k = \mathbf{Q}^T \psi_k, \quad (7.15)$$

where τ_k and v_k are the scores, i.e. the τ -scores and the v -scores that correspond to the projections of the predictor and response variables onto the corresponding subspaces. The columns of the matrix \mathbf{R} act as weightings that determine the τ -scores. The columns of \mathbf{Q} are the loadings of the response variables respectively and the matrix product \mathbf{QBR}^T is denoted as the regression model of the outer model, i.e. the regression model between the predictor and the response variables [119].

The score variables are a reduced variable set that describes the significant variation of the predictor and predicted variables. Consequently, under “in-control” operating conditions, the mismatch between the measured and reconstructed (or predicted) values of both X- and Y-Blocks describes “insignificant” variation and is determined as follows [128]:

$$\mathbf{e}_k = \psi_k - \mathbf{Q}\hat{\mathbf{v}}_k = \psi_k - \mathbf{QB}\tau_k = \psi_k - \mathbf{QBR}^T\xi_k, \quad (7.16)$$

$$\mathbf{f}_k = \xi_k - \mathbf{P}\tau_k = (\mathbf{I}_{n_x} - \mathbf{PR}^T)\xi_k, \quad (7.17)$$

where \mathbf{f}_k and \mathbf{e}_k are vectors storing the residuals of the predictor and the predicted variables and $\hat{\mathbf{v}}_k$ represents the v -scores for the k^{th} sample of the predicted variables. \mathbf{P} is a regression matrix that minimises the mismatch between recorded samples of the predictor variables and their projections onto the corresponding subspace. Given the above description of the PLS algorithm, three univariate statistics can be established:

- (1) A T^2 statistic which is based on the τ -score variables,
- (2) A Q statistic that corresponds to the residuals of the response variables,
- (3) A Q statistic which is based on the residuals of the predictor variables.

For the k^{th} instance, these statistics are defined as follows:

$$PLS - T_k^2 = \tau_k^T \mathbf{S}_n^{-1} \tau_k, \quad (7.18)$$

$$PLS - Q_k^{(\psi)} = \mathbf{e}_k^T \mathbf{e}_k, \quad (7.19)$$

$$PLS - Q_k^{(\xi)} = \mathbf{f}_k^T \mathbf{f}_k, \quad (7.20)$$

where $PLS - T_k^2$, $PLS - Q_k^{(y)}$ and $PLS - Q_k^{(x)}$ represent the T^2 statistic, the Q statistic of the predictor and the Q statistic of the response variables respectively, and S_{τ} is a diagonal matrix in which the values of the variance of the τ -score variables are stored in successive order.

The T^2 statistic for PLS has the same definition as the T^2 statistic for PCA and hence, the confidence limit for this statistic can be determined as outlined in Eq. 7.12 by replacing n_{PCA} with n_{PLS} where n_{PLS} represents the number of retained latent predictor variables, i.e. the dimension of both subspaces. The confidence limits of the Q statistics can be determined by utilising the approach by Box [111] as given in Eq. 7.13.

7.2.4 Dynamic PCA and Dynamic PLS

Ku [91] showed that a linear time-series relationship can be incorporated into the conventional PCA analysis. For example, a general set physical process variables can be arranged to represent an ARX model structure:

$$\zeta_{k,n,m}^T = \begin{pmatrix} Y_k^T & \cdots & Y_{k-n}^T & U_{k-1}^T & \cdots & U_{k-m}^T \end{pmatrix}, \quad (7.21)$$

where k represents an arbitrary sample point and $\zeta_{k,n,m}$ is the augmented set of variables, representing an ARX model of order n , m defines the input delay spread, and Y and U are the predicted and predictor measurement sequences respectively.

Utilising this “extended” set of process variables, a PCA analysis can be carried out as described above. The confidence limits of the T^2 and Q statistics can be determined as described in Eqs. 7.12 and 7.13. However, the total number of observations reduces to $k - (n, m)$ where (n, m) indicates n or m , whichever is the larger, and a different number of principal components, n_{DPCA} , may need to be retained.

In recent years, a growing number of applications of dynamic PLS have been reported in process monitoring and control, e.g. [129-131]. Thereby, the dynamics model structure is incorporated into the predictor variables:

$$Y_k = \mathbf{B}_0 U_k + \sum_{i=1}^m \mathbf{A}_i Y_{k-i} + \mathbf{B}_i U_{k-i} + \varepsilon_k \quad (7.22)$$

where m is the order of the ARX model structure, \mathbf{A}_i and \mathbf{B}_i are matrices containing the model coefficients and the vector ε_k contains the residuals of the response variables.

Utilising the augmented set of predictor variables, $\zeta_{k,n,m}$, the PLS algorithm can be employed for process monitoring as discussed above. Note that the dynamic model structure results in a reduction of the number of observations to $k - (n, m)$ and that a different number of lvs , n_{PLS} , may need to be retained. It should be noted that the above model structure also includes a non-causal term, i.e. $\mathbf{B}_0 U_k$. This term is generally required in the presence of controller feedback, caused by regulatory or model based controller(s), which manifests itself in the predictor variables.

7.3 CSTR Simulation

The CSTR simulation is modelled on a nonisothermal continuous stirred tank reactor (CSTR) [132]. A first order exothermic reaction takes place inside the mixing tank. The temperature (T) and concentration (C_A) of the outlet stream are controlled to set-point by manipulating the coolant flow (F_C) and reactant flow (F_A) respectively. Heat is removed from the tank through the cooling coils. A process flow diagram of the CSTR system appears in Figure 7.2.

The time step for data collection was set to 10 seconds, so that 1200 samples were collected in 200 minutes of process operation. There are nine predictor variables, and two predicted variables. The predictor variables are the two manipulated variables (reactant flow and coolant flow), and seven measured disturbances (solvent flow, solvent concentration, reactant concentration, inlet temperature, inlet concentration, inlet flow and coolant inlet temperature). The two predicted variables are product temperature and product concentration. Each of the measured disturbances is simulated as a first order autoregressive sequence, subject to zero-mean normally distributed measurement noise: $(x_k = ax_{k-1} + e_k)$. In addition there are two process constants (the reaction rate and the heat transfer rate), each of these is autocorrelated and is subject to an unmeasured stochastic influence, for more details see [100, 132].

Three process faults were created using the CSTR process simulation. In these experiments the outlet concentration controller was turned off so that only the outlet stream temperature is controlled to set-point using the cooling water flow. The faults are

- **FAULT A** Fault A is a reactant flow (F_A) bias of $-0.015 \text{ m}^3/\text{min}$, simulated to occur at $T=51$ minutes. This leads to a breakdown in the relationship $(F_A + F_S) = F_{AC}$. The fault also propagates through to the product concentration. Figure 7.5 shows the effect of the fault on the other process variables.

- **FAULT B** The process variables for Fault B are shown in Figure 7.10. Fault B is the simulation of a bias ($\Delta T = -1 \text{ degK}$) in the outlet stream temperature sensor, occurring at $T=51$ minutes. The controller acts on the apparent drop in the measured temperature and changes the set-point for the cooling water valve which leads (incorrectly) to a decrease in the flow of the coolant. Figure 7.10 shows the evolution of the affected process variables over time. The temperature in the reaction tank rises, leading to a rise in reaction rate, which in turn causes the outlet concentration to drift down from its steady state value of 0.8 mol/min .

- **FAULT C** The process variables for Fault C are shown in Figure 7.15. Fault C is a simulation of a process disturbance. The solvent flow (F_S) drops suddenly at $T=51$ minutes. The disturbance propagates through to the inlet flow (F_{AC}) measurement according to the relationship $(F_A + F_S) = F_{AC}$. As a result the inlet concentration rises leading to an increase in the outlet concentration which drifts upward.

The contribution charts for each of the models are made up of nine predictor variables: solvent flow (∇); reactant flow (T); cooling water flow ($+$); cooling water inlet temperature (\circ); solvent concentration (\square); inlet flow (x); reactant concentration (\triangleright); inlet temperature (\triangleleft) and inlet concentration (\triangle). The two predicted variables are product temperature (\diamond) and product concentration (\bullet). A full list of the process variables is given in Table 7.2.

7.4 Setting the Model Structure

The state space model structure for the subspace method and the ARX model structure for the other dynamic methods were set as detailed in the following sections.

7.4.1 The model structure for the Subspace Method

While the state matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ are not required explicitly in the condition monitor, they prove useful for optimising the user choices required for the subspace condition monitor. The three user choices required for the subspace method are the model order (n), the number of block rows (r) and the number of latent variables used to monitor the process (M), i.e. the model structure for the subspace monitor is completely determined by the triplet (n, r, M) .

The procedure for specifying the structure of the subspace model consisted of three stages as follows

- (1) First identify the order (n) to be possessed by the model. A low order model, (and consequently low dimension state sequences) was found to give satisfactory results (a second order state space model is used here).
- (2) Choose the number of block rows (r) to calculate the model, where $r = 5$ was settled upon, on the basis of it providing a stable state space model.
- (3) Choose the number of principal components (M) to use in the monitor. Cross-validation was applied to a PCA matrix made up of the process measurements, excluding the state sequences, to find the optimum number of principal components.

The best number of principal components to use remains an open question because the state sequences included in the analysis are orthogonal. An initial guess involved applying cross-validation to the PCA matrix (i.e. the data set without the state sequences). The number of principal components recommended from this cross validation was then be chosen as the number of latent variables to include in the subspace condition monitor. Since the state sequences are calculated from the same row and column space as the process data, it is expected that adding them to the PCA

analysis will not call for the inclusion of extra principal components in the process monitor. The model structure used to monitor the CSTR process was $(n, r, M) = (2, 5, 4)$.

7.4.2 The model structures for dynamic PCA and dynamic PLS

The DPCA and DPLS models were based on an ARX model structure. The ARX model structure was calculated on the basis of AIC, and was found to be a second order model, with a zero order lag and an input delay spread of three, i.e., for the DPCA analysis:

$$\mathbf{Z} = \left(\mathbf{Y}_k^T \quad \mathbf{Y}_{k-1}^T \quad \mathbf{Y}_{k-2}^T \quad \mathbf{U}_k^T \quad \mathbf{U}_{k-1}^T \quad \mathbf{U}_{k-2}^T \right)^T \in \mathbb{R}^{N \times 2 \times 3(l+m)}$$

For the DPLS analysis, the X- and Y-Blocks are

$$\mathbf{X} = \left(\mathbf{Y}_{k-1}^T \quad \mathbf{Y}_{k-2}^T \quad \mathbf{U}_k^T \quad \mathbf{U}_{k-1}^T \quad \mathbf{U}_{k-2}^T \right)^T \in \mathbb{R}^{N \times 2 \times (2l+3m)}$$

$$\mathbf{Y} = \mathbf{Y}_k \in \mathbb{R}^{N \times 2 \times l}.$$

The leave-one-out cross-validation procedure from the Matlab PLS Toolbox was used to calculate an “optimum” number of principal components to use with DPCA, where it was found $M = 10$. The calculation of the PRESS statistic for the PLS model revealed the number of latent variables required by the model to be ($lv = 6$).

A PCA analysis was also carried out, where the optimum number of principal components was found to be $M = 3$.

The results section is organised as follows. The results for the subspace method, DPCA and DPLS are presented together in Figures 7.6 to 7.19. Finally the results for the PCA monitor are presented in Figures 7.20 to 7.22. The PCA results are included to provide a reference point, since PCA is the most commonly used method in monitorMV. However the main thrust of the comparison is between the three dynamic modelling methods.

7.5 Results

The results for the CSTR simulation study are laid out in Figures 7.6 – 7.22. The horizontal dotted line in the Hotelling's T^2 and the Q statistics charts (e.g. Figure 7.6) indicates the 99% likelihood limit that the process remains within the bounds of normal operation. For clarity of presentation, the contribution charts for the T^2 and Q statistics are plotted at an interval of 1 min 40 secs. The magnitude of the confidence limits for the T^2 and Q statistics charts varies according to the number of latent variables (lvs) employed by each of the models. Note that the T^2 confidence limit is highest for the DPCA model (10 lvs) and smallest for the subspace model (4 lvs). The confidence limits for the DPLS model (6 lvs) lies between the other two for both the T^2 and Q statistics charts.

7.5.1 Fault A

Fault A involves a input (reactant flow) sensor bias. Figure 7.5 shows the effect of the fault on the process variables. Figure 7.6 shows the Q statistics for each of the models. The subspace method gives the clearest indication of a fault. The Q_{DPLS} statistic for the predicted variables rises slightly but does not consistently pass the alarm level. Figure 7.7 shows the Q statistic contributions charts. The Q_{DPCA} statistic does not pass the alarm limit - yet inlet concentration (Δ) and reactant flow (T) pass the 99% confidence limit. The density of the information due to the $3m+2l$ process variables used in DPCA presents a challenge for the presentation of a clear graphical analysis. The subspace method uses state sequences, in contrast to the time-shifted data used by DPCA, to capture the process dynamics. This means that the dimension of contributions plots is much smaller than for DPCA. The state sequences are low dimension latent directions that define the larger number of (usually correlated) process variables. Therefore, one advantage of the subspace methods is in the clarity of presentation, and therefore easier analysis of the contributions plots.

Figures 7.8 and 7.9 show the Hotellings T^2 charts. None of the process variables passes the confidence limit. A summary of the Q statistic contributions charts is included in Table 7.1 on page 190, where “NA” indicates “no alarm”. Fault A is a bias in the

reactant flow (T). It is reasonable to expect that the solvent and inlet flows (∇x), the outlet temperature and concentration ($\diamond \bullet$) and the cooling water flow (+) (due to controller action) are all affected by the fault. In fact the key relationship that is affected is $(F_A + F_S) = F_{AC}$, so that the key variables affected are likely to be ($\diamond \bullet \Delta \nabla x + T$). On the basis of the Q contributions, the subspace ($\Delta + T$), DPLS($\bullet \Delta T$), and DPCA ($\bullet \Delta T$) all provide information for fault diagnosis.

7.5.2 Fault B

Fault B is caused by a bias in the output temperature sensor. This causes the controller to increase the cooling water flow. Figure 7.10 shows the effect of the fault on the process variables. Figure 7.11 shows that for each of the dynamic models, the Q statistic passes the 99% confidence limit. Figure 7.12 shows the contributions to the Q statistics charts. The bias in the outlet temperature sensor (\diamond) causes the controller to close the cooling water flow valve (+). A summary of the Q statistic contributions for Fault B is given in Table 7.1. The main contributions to the Q Statistics are: subspace model ($\diamond \bullet \Delta + T$), DPCA ($\diamond \bullet \circ +$) and DPLS ($\diamond \bullet +$). The time-shifted values of the product concentration (\bullet) are also indicated by Q_{DPLS} and Q_{DPCA} . Each model indicates ($\diamond \bullet +$) contributions as possible causes of the fault.

The Hotellings T^2 charts are shown in Figure 7.13. Each of the charts passes the 99% confidence bounds. The contributions to Hotellings T^2 statistics are shown in Figure 7.14. The main subspace method contributions are the state sequences (---) and the cooling water flow (+). The contribution of the state sequences does not provide explicit diagnosis information but may be a consequence of the control action.

7.5.3 Fault C

Fault C is a bias in the solvent flow (∇). Figure 7.15 shows the effect of the fault on the process variables. The limits of each of the models in the Q statistics charts (Figure 7.16) and Hotellings T^2 charts (Figure 7.18) pass the 99% confidence limit. The Q statistic contributions for the subspace and DPCA models are similar, however the

subspace method Q statistic chart is easier to read due to the use of fewer process variables (Figure 7.17).

The contributions to the T^2 charts (Figure 7.19) also indicate similar results for the subspace method and DPCA. This may be because they both employ PCA, and both apply PCA to matrices that contain the vectors $(\mathbf{Y}_k \ \mathbf{U}_k)$, to calculate the latent directions in the data. However the subspace model provides additional compression of the process data (using the state sequences), that summarises the information contained in the time-shifted sequences used in DPCA.

Fault C is a solvent flow (∇) bias. The bias filters through to many of the process variables due to the dynamics of the process and the control action. A summary of the Q statistic contributions for Fault C is given in Table 7.2. The main contributions to the Q statistics are subspace method ($\Delta \nabla \times \square$), DPCA ($\Delta \nabla \times \square$) and DPLS ($\Delta \nabla \times +$). Note the presence of the solvent flow (∇) in the contributions to both T^2 and Q statistics for all three models.

Each of the methods described above calculates a latent variable space that maximises the fit between the model and the system measurements. PLS maximises the covariance between the X and Y Blocks, PCA maximises the variance while subspace methods maximise the correlation in the data to calculate the state sequences [97]. As a consequence, the latent directions differ for each model. In this study, DPCA uses 10 *lvs*, DPLS 6 *lvs* and subspace 4 *lvs*. The summary of the contributions in Table 7.1 indicates that all the models have captured the process anomalies in a similar but distinct way.

The dynamics of the process causes ambiguity when trying to isolate the source of each of the faults. Several authors e.g. [100, 101] have pointed to the use of fault signatures as a diagnostic aid for locating the source of process anomalies. The fault diagnosis capability of the dynamic models could be enhanced by creating a library of “footprints” for various faults, by using the information in Table 7.1.

	Contributions to Q statistic			Contributions to T ² statistic		
	Fault A	Fault B	Fault C	Fault A	Fault B	Fault C
Subspace	$\Delta+T$	$\diamond \bullet \Delta+T$	$\Delta \nabla x \square$	NA	$+X_k$	$\nabla x \square \triangleright$
DPCA	$\bullet \Delta T$	$\diamond \bullet \circ +$	$\Delta \nabla x \square$	NA	$\diamond \bullet +$	$\nabla x \square$
DPLS	$\bullet \Delta T$	$\diamond \bullet +$	$\Delta \nabla x +$	NA	$\diamond \circ \triangleleft +$	∇
PCA	NA(ΔT)	+	$\Delta \nabla x \square$	NA	NA	$\nabla x \square \triangleright$

Table 7.1 Contributions to the Q statistics charts for Faults A, B, and C.

7.5.4 Comparison between the subspace method and PCA

The comparison between the subspace method and PCA relates to a subspace model that used 4 lvs to monitor the process, and a PCA model that used 3 lvs to monitor the process. A direct comparison between the results for the subspace method and PCA invites itself for two reasons:

- (1) $Z_{SUBSPACE}$ consists of Z_{PCA} plus the state sequences \bar{X}_k and \bar{X}_{k+1} . A direct comparison of the T^2 and Q statistics, and Q_{CONT} and T_{CONT}^2 will lead to an assessment of the effect that the “extra information” brings to the analysis.
- (2) PCA is a natural choice for benchmarking the performance of the subspace method, because it is relatively simple to apply, and it is the most commonly used method in monitorMV.

Figures 7.20 to 7.22 describe the statistics generated by the PCA model for each of the CSTR faults. The effect of augmenting Z_{PCA} to create a PCA analysis using $(\bar{Z}_{PCA} \quad \bar{X}_k \quad \bar{X}_{k+1})$ is indicated by comparing the appropriate rows of Table 7.1 above. The results for Fault A and Fault C are similar for the subspace and PCA methods. The major difference is with Fault B, which is a fault that is associated with the control loop of the process. One interpretation of the results for Fault B is that the control action alters the dynamics of the process, and that this is more clearly indicated by the subspace method. Although the contributions summary in Table 7.1 indicates that the same process variables contributed most to the fault statistics, a comparison of the contribution charts, i.e. Figures 7.7 & 7.20, Figures 7.12 & 7.21, and Figures 7.17 & 7.22 are evidence that the subspace method analysis produces a distinctive picture of each of the special events.

PCA_{CV2} indicated three principal components capturing 63% of the variance for the PCA monitor and four principal components capturing 73.9% of the variance for the subspace method monitor. Tables 7.3 and 7.4 show the cumulative variance captured by PCA and the subspace method based on the training data. Considering the use of either three principal components or four principal components, the subspace method captures marginally more of the variance in the data than PCA.

7.5.4.1 The effect of scaling the states

Equation 6.9 (containing the original unscaled sequences) forms a valid TLS solution to the subspace system identification problem, where the matrix $\mathbf{Z}_{msc.} = [\mathbf{X}_{k+1}^T \quad \mathbf{Y}_k^T \quad \mathbf{X}_k^T \quad \mathbf{U}_k^T]^T$ contains unscaled state sequences. The contribution of the state sequences $\mathbf{X}_{msc.}$ to the formation of the process monitoring model $\Phi_{msc.}$ is plotted in bar charts in Figure 7.3. The first four bars indicate that the loadings of the states $\mathbf{X}_{k,msc.}$ (columns one and two) and $\mathbf{X}_{k+1,msc.}$ (columns three and four) to $\Phi_{msc.}$ are very small. In effect, the trivial effect of $\mathbf{X}_{msc.}$ on the formation of $\Phi_{msc.}$ means that the monitor is comprised of a loading matrix $\Phi_{msc.}$ that is very similar to Φ_{PCA} . It is therefore found that the application of the model $\Phi_{msc.}$ as a process monitor gives almost the same results as if employing Φ_{PCA} . This was also suggested by the application of PCA_{CV2} to $\mathbf{Z}_{msc.}$, which indicated that three principal components captured the important process variance, i.e. the same number as was found using PCA.

The effect of scaling the states is shown in the bar chart in Figure 7.4. The state sequences, $\bar{\mathbf{X}}_k$ (columns one and two) and $\bar{\mathbf{X}}_{k+1}$ (columns three and four) are seen to contribute significantly to the formation of the first four principal components. Because X_k^1 is orthogonal to X_k^2 and X_{k+1}^1 is orthogonal to X_{k+1}^2 , each of the directions defined by the sequences $\bar{\mathbf{X}}_k$ and $\bar{\mathbf{X}}_{k+1}$ contributes independently to the formation of the model. It is reasonable to expect that the contribution of the vectors of $\bar{\mathbf{X}}_k$ and $\bar{\mathbf{X}}_{k+1}$ in the analysis provides a dynamic model for the process and that can deal more easily with autocorrelated and dynamic data.

7.6 Conclusion

The CSTR continuous process simulation has been presented in this chapter to introduce the subspace method for process condition monitoring, which was compared with two alternative dynamic modelling procedures:

- (1) A DPCA method, based on an ARX model structure, which is part of the MSPC toolbox in monitorMV. The method was applied to the CSTR process to provide a basis for a comparison with the novel subspace method.
- (2) The PLS method has been used to construct a dynamic model based on an ARX model structure, where a reduced number of latent variables (6) were used to model the process. A Hotelling's T^2 was defined, based on the contributions of the predictor variables to the model predictions. Furthermore, Q statistics were defined for both the predictor and predicted variables of the process, as the distance from the inner model to the relevant predictor variables, and as the distance of the PLS outer regression model to the predicted variables.

The subspace method has been shown to be equivalent to DPCA in the sense that both DPCA and the subspace method are based on TLS solutions for well-known linear parametric model structures. A possible advantage enjoyed by the subspace method is that it uses significantly fewer latent variable to monitor the process and to develop its contribution charts. State space models do not require time-shifted process data in their model structures, and are therefore more concise than those that are based on dynamic PCA/PLS (using ARX model structures

The subspace method has provided a similar level of fault detection and diagnosis as DPCA. Given that the subspace method and DPCA are both linear methods, where both methods employ a TLS procedure to identify dynamic models, it is clear that the subspace method enjoys the following two advantages over DPCA,

- (i) A contributions analysis that employs fewer process variables in the model
- (ii) A simpler analysis, where fewer latent variables are used to monitor the process.

The subspace method may also be considered as a dynamic extension to PCA, where a PCA analysis is carried out on a matrix composed of the process measurements and the associated state sequences of the process. A PCA analysis of the CSTR process faults was used to compare the subspace method with PCA, which revealed that the subspace method has provided similar information to PCA on the basis of the more simple faults (Fault A and Fault C). The analysis of a more complex fault involving a feedback loop (Fault B), led to very different contributions plots for each of the methods, which may suggest that the added dynamic information contained in the states has provided further information not include in the PCA matrix.

The CSTR process simulation faults were clearly detected by each of the models, however each has given a different indication as to the cause of the fault. This suggests that a considerable amount of information can be drawn together by applying all the dynamic (and static) monitors simultaneously. The contribution plots have tended to yield ambiguous results, partly due to the effect of controller feedback. One way to integrate the strengths of each of the models is to combine the information for particular faults into a chart such as Table 7.1.

The subspace method is proposed as part of a total solution for condition monitoring of industrial plant. A popular approach in monitorMV is to pile all the measured variables into a single data matrix and then build a static PCA model. However, where necessary, an alternative strategy could be to run several dynamic models in parallel, each containing a subset of the set of all measured variables of the process. A combined modelling approach is therefore proposed with dynamic models providing intensive monitoring, where appropriate, while sitting above and independent of the dynamic models is a static PCA monitor filling its traditional role.

7.8 Figures and Tables

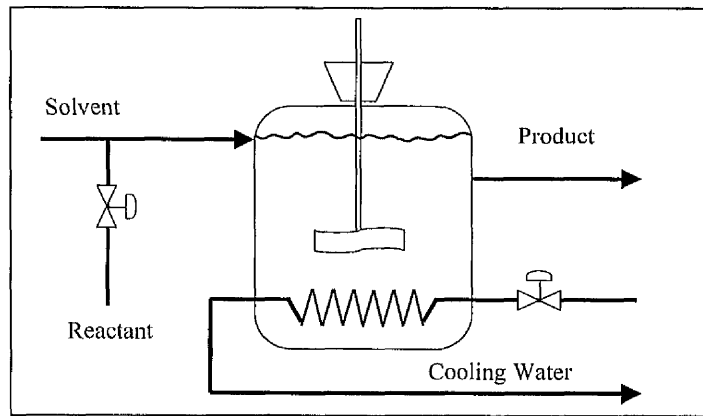


Figure 7.2 Process flow diagram for the CSTR system. The temperature and concentration of the product stream are controlled to set-point using the Cooling Water and Reactant valves respectively.

	PROCESS VARIABLE	SYMBOL
T	Reactor Outlet Temperature	\diamond
C_A	Reactor Outlet Concentration	\bullet
F_C	Cooling Water Flow	$+$
T_C	Cooling Water Inlet Temperature	\circ
C_{AC}	Inlet Concentration	\triangle
F_{AC}	Inlet Flow	\times
T_O	Inlet Temperature	\triangleleft
C_{AA}	Reactant Concentration	\triangleright
F_A	Reactant Flow	T
C_{AS}	Solvent Concentration	\square
F_S	Solvent Flow	∇

Table 7.2 List Of Process Variables

Principal Component Number	Eigenvalue of Cov (X)	% Variance Captured	
		This PC	Total
1	2.98	27.05	27.05
2	2.47	22.48	49.53
3	1.51	13.72	63.24
4	1.10	9.96	73.21
5	0.95	8.62	81.83
6	0.92	8.39	90.22
7	0.49	4.48	94.70
8	0.38	3.49	98.19
9	0.20	1.8	99.99
10	6×10^{-4}	0.01	100
11	3×10^{-13}	0.00	100

Table 7.3 Percent Variance Captured by PCA Model.

Principal Component Number	Eigenvalue of Cov (X)	% Variance Captured	
		This PC	Total
1	4.41	29.39	29.39
2	2.69	17.93	47.33
3	2.46	16.43	63.76
4	1.52	10.13	73.89
5	1.10	7.31	81.20
6	1.03	6.88	88.08
7	0.93	6.20	94.28
8	0.39	2.58	96.86
9	0.38	2.56	99.42
10	5×10^{-2}	0.33	99.75
11	2×10^{-2}	0.13	99.88

Table 7.4 Percent Variance Captured by the subspace method.

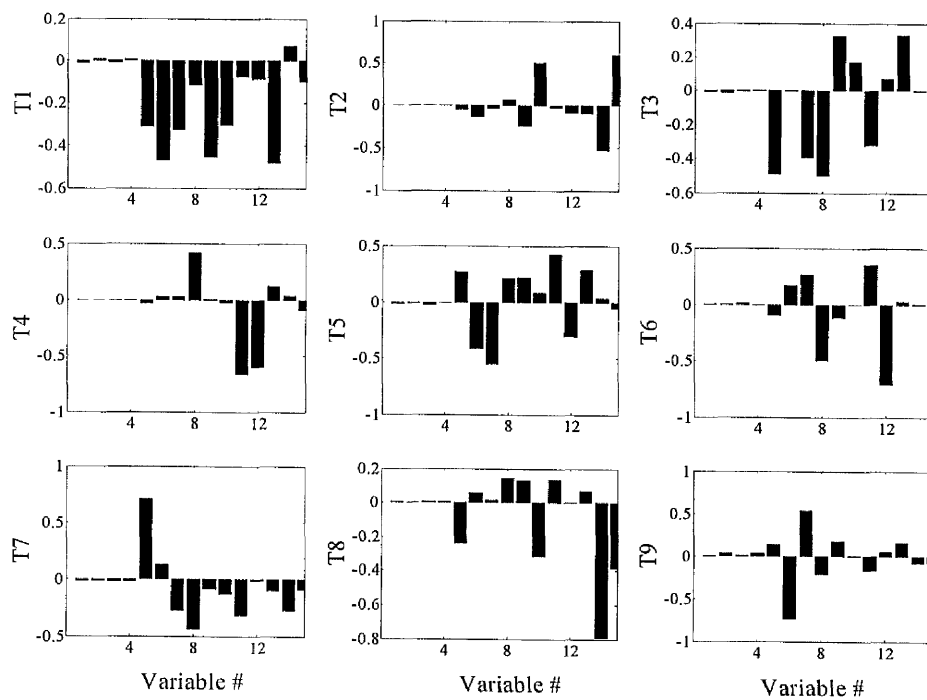


Figure 7.3 PCA loadings for a TLS model using unscaled states.

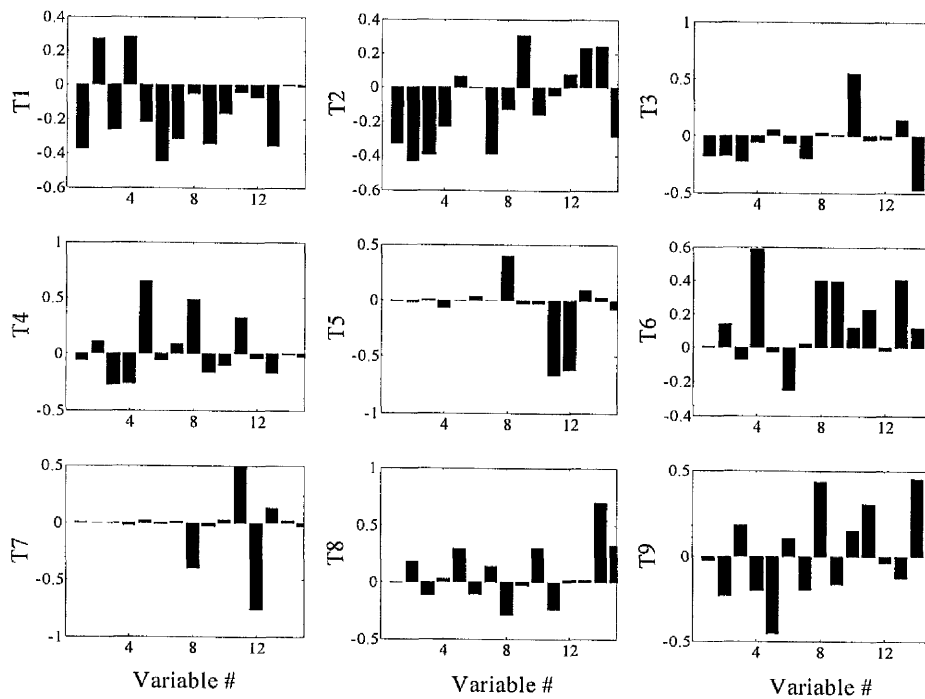


Figure 7.4 PCA loadings for a TLS model using scaled states.

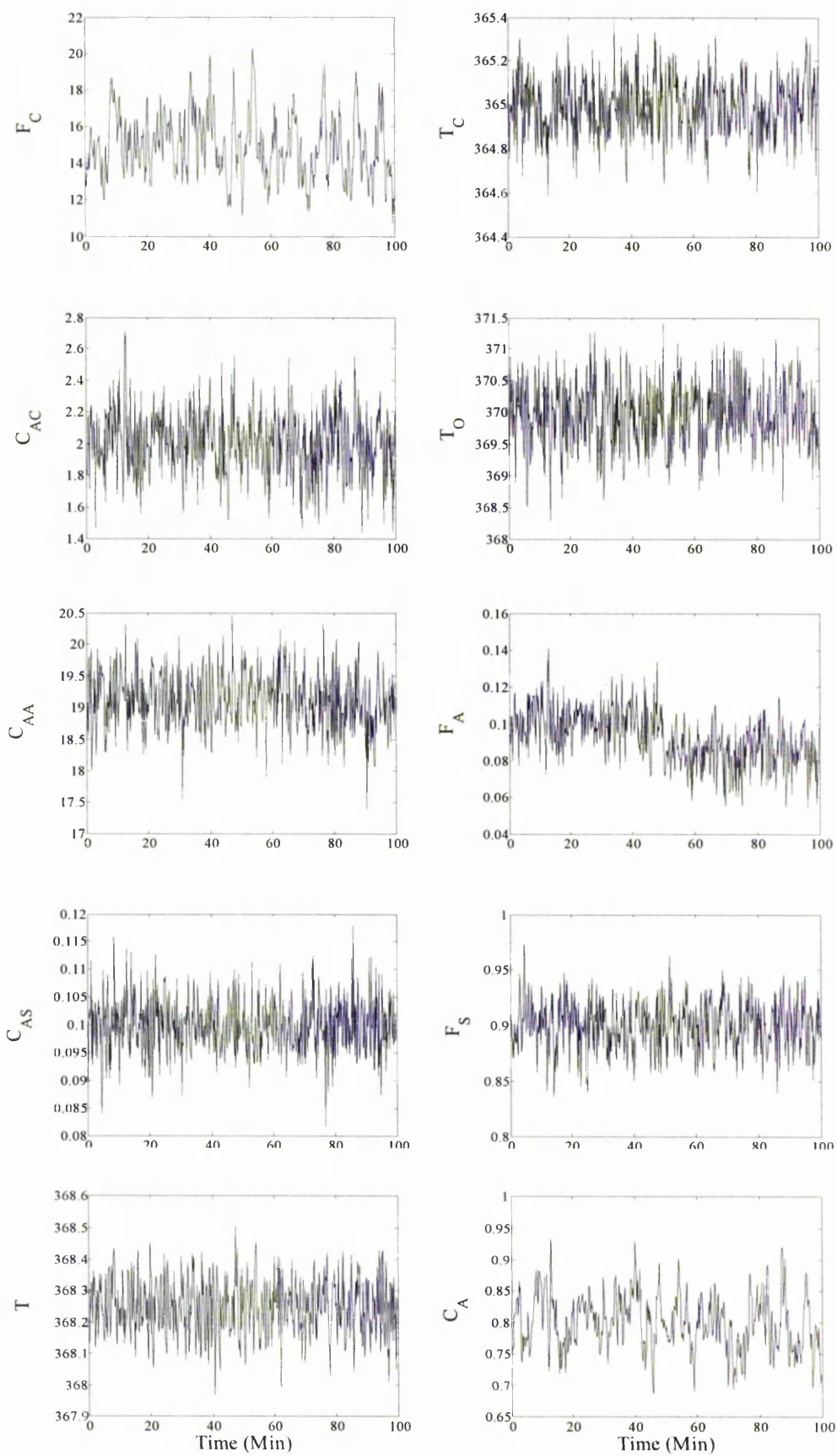


Figure 7.5 Process measurements for Fault A

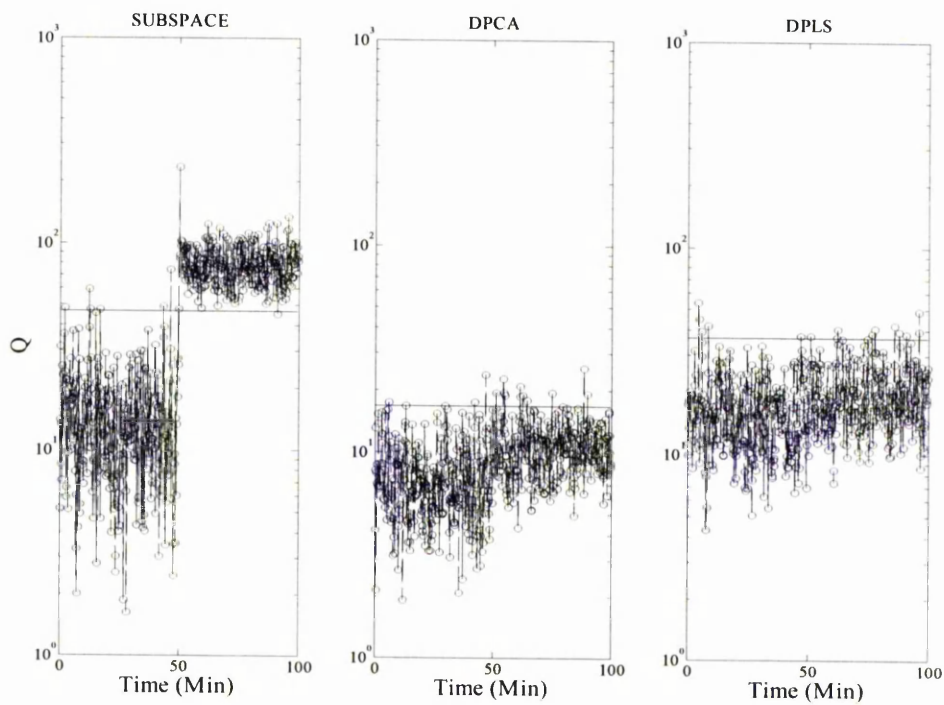


Figure 7.6 Q statistics. **FAULT A**

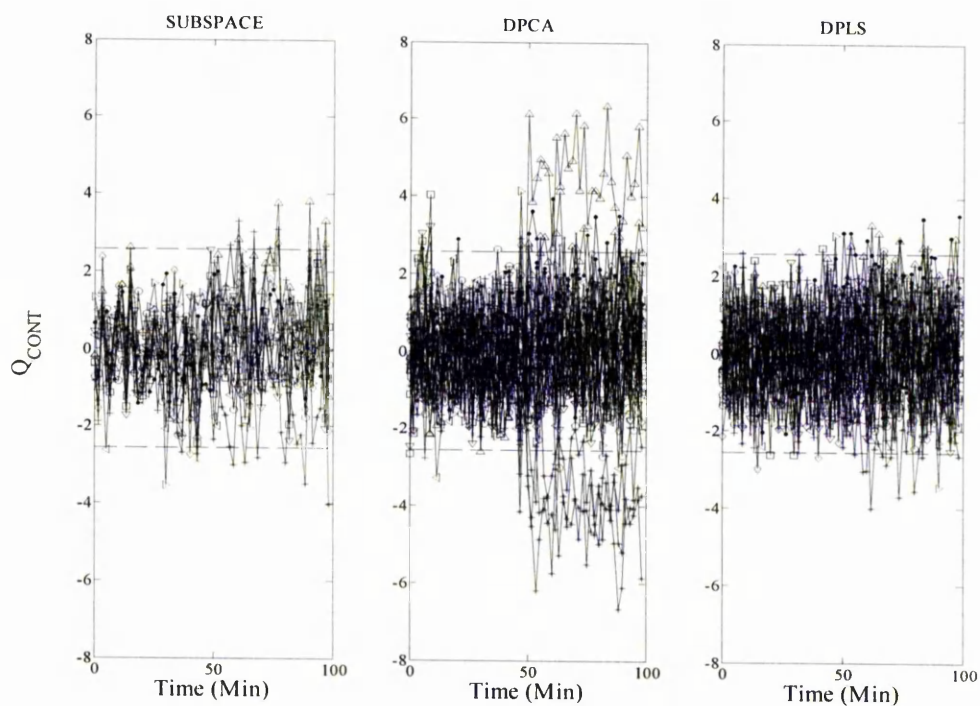


Figure 7.7 Contributions to Q statistic. **Fault A** F_S (∇); F_A (T); F_C (+); T_C (\circ); C_{AS} (\square); F_{AC} (x); C_{AA} (\triangleright); T_O (\triangleleft); C_{AC} (\triangle); T (\diamond); C_A (\bullet).

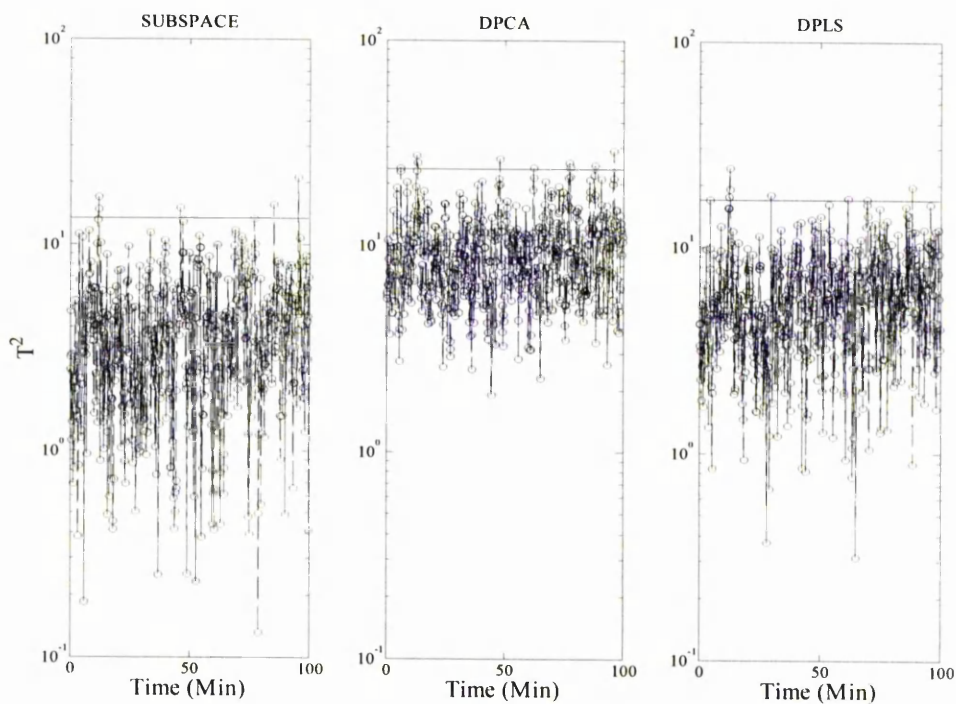


Figure 7.8 Hotelling's T^2 statistics. **FAULT A**

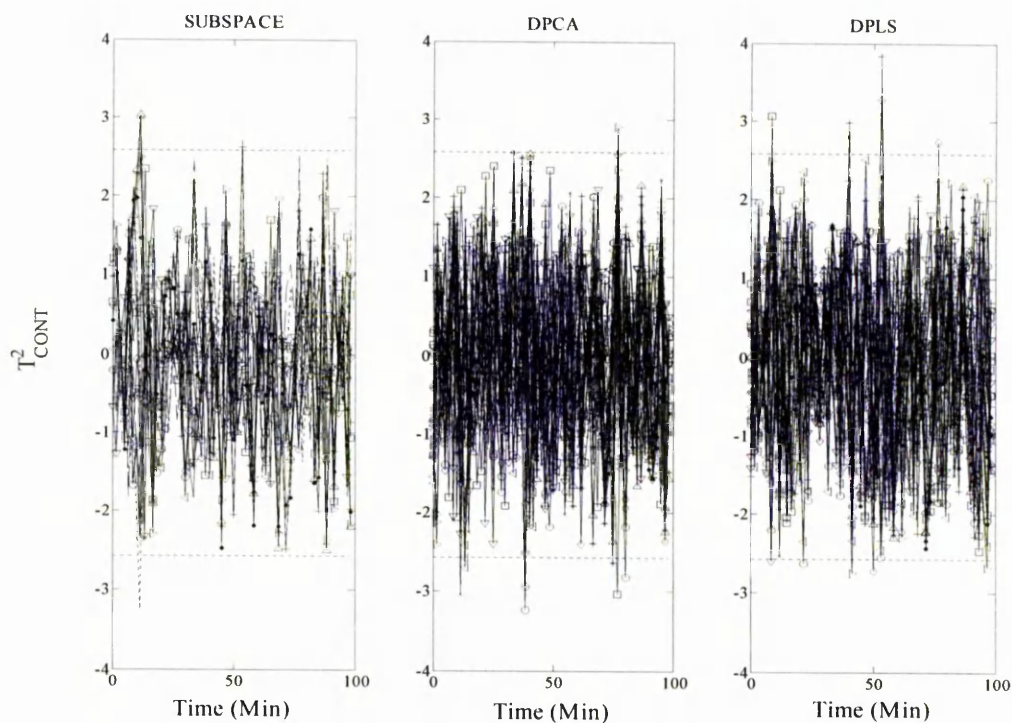


Figure 7.9 Contributions to T^2 statistic. Fault A F_s (∇); F_A (T); F_C (+); T_C (\circ); C_{AS} (\square); F_{AC} (x); C_{AA} (\triangleright); T_O (\triangleleft); C_{AC} (\triangle); T (\diamond); C_A (\bullet).

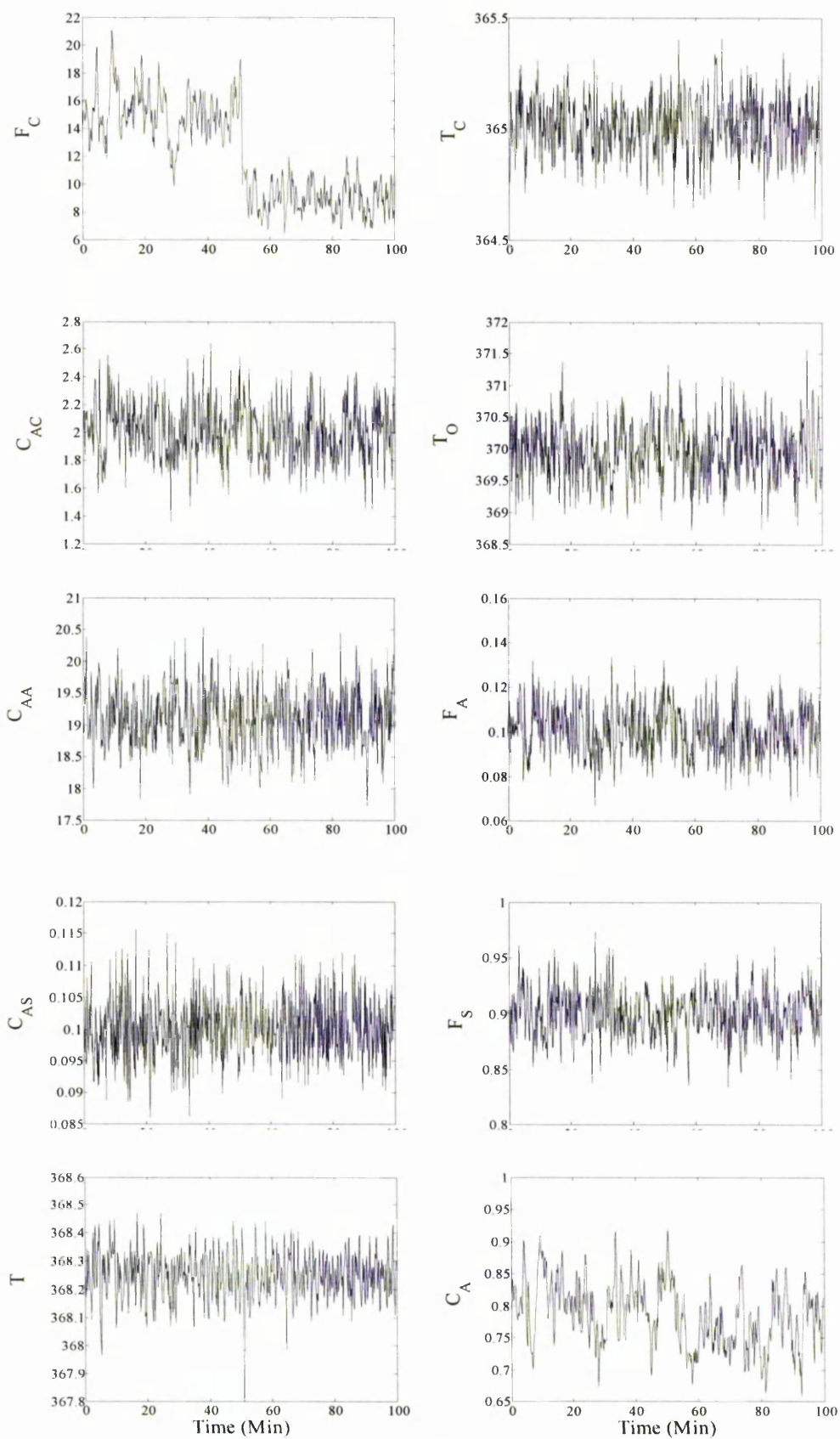


Figure 7.10 Process measurements for **Fault B**

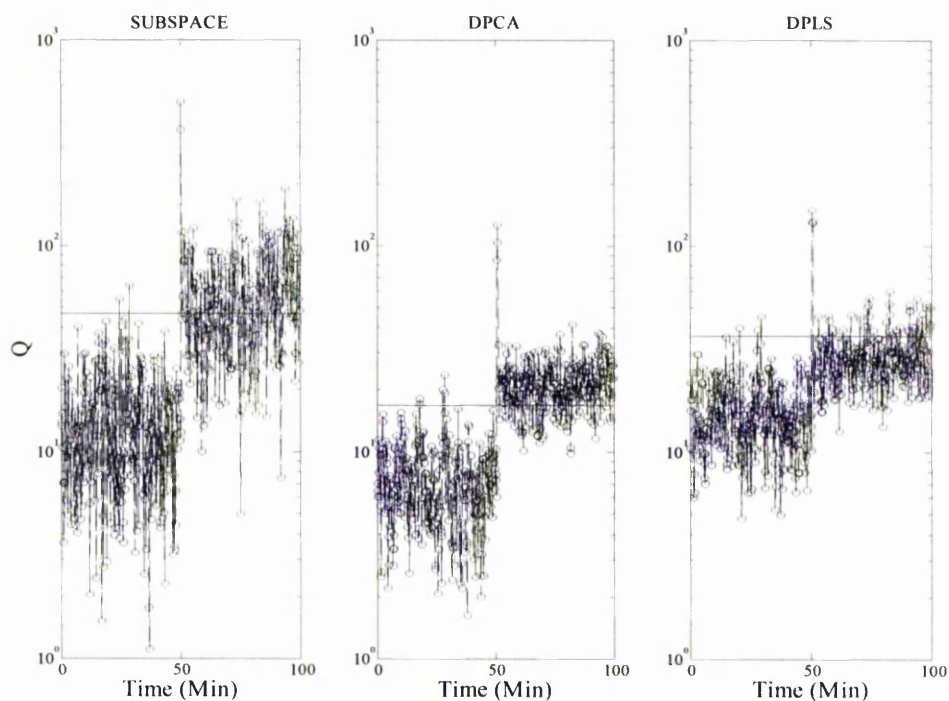


Figure 7.11 Q statistics. **FAULT B**

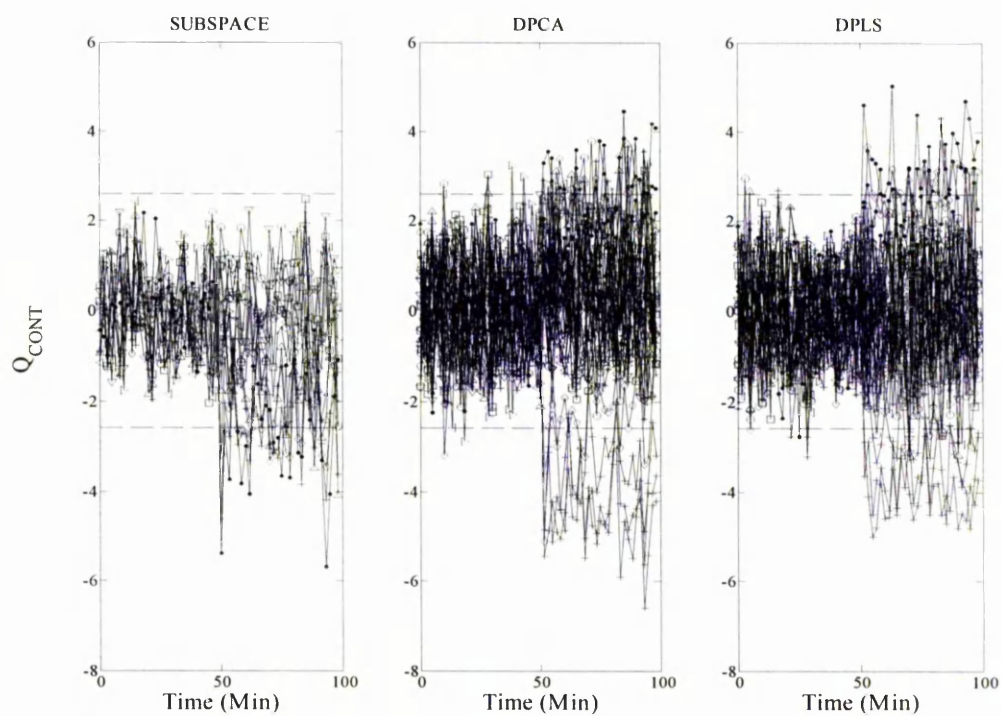


Figure 7.12 Contributions to Q statistic. **Fault B** F_S (∇); F_A (T); F_C (+); T_C (\circ); C_{AS} (\square); F_{AC} (x); C_{AA} (\triangleright); T_O (\triangleleft); C_{AC} (Δ); T (\diamond); C_A (\bullet).

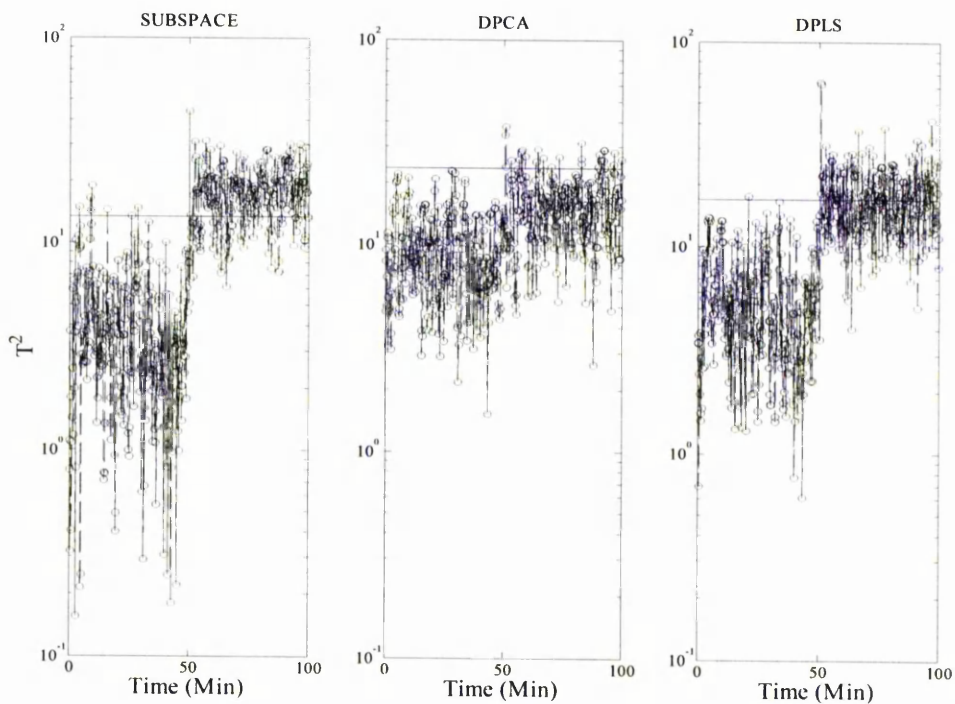


Figure 7.13 Hotelling's T^2 statistics. **FAULT B**

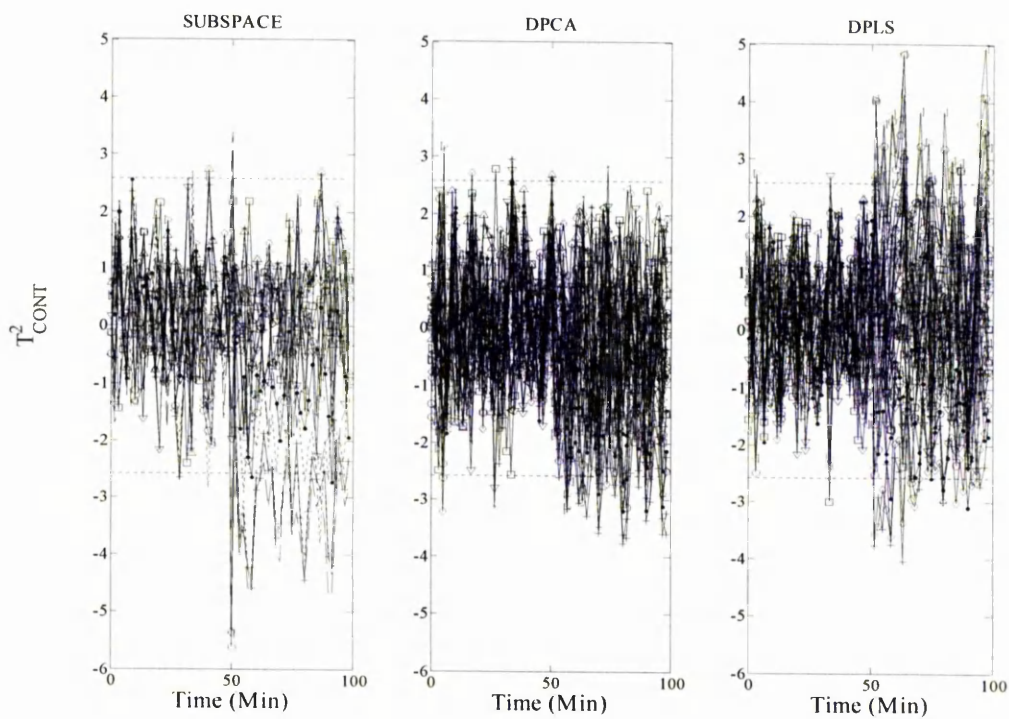


Figure 7.14 Contributions to T^2 statistic. **Fault B** F_S (∇); F_A (T); F_C ($+$); T_C (\circ); C_{AS} (\square); F_{AC} (\times); C_{AA} (\triangleright); T_O (\triangleleft); C_{AC} (Δ); T (\diamond); C_A (\bullet).

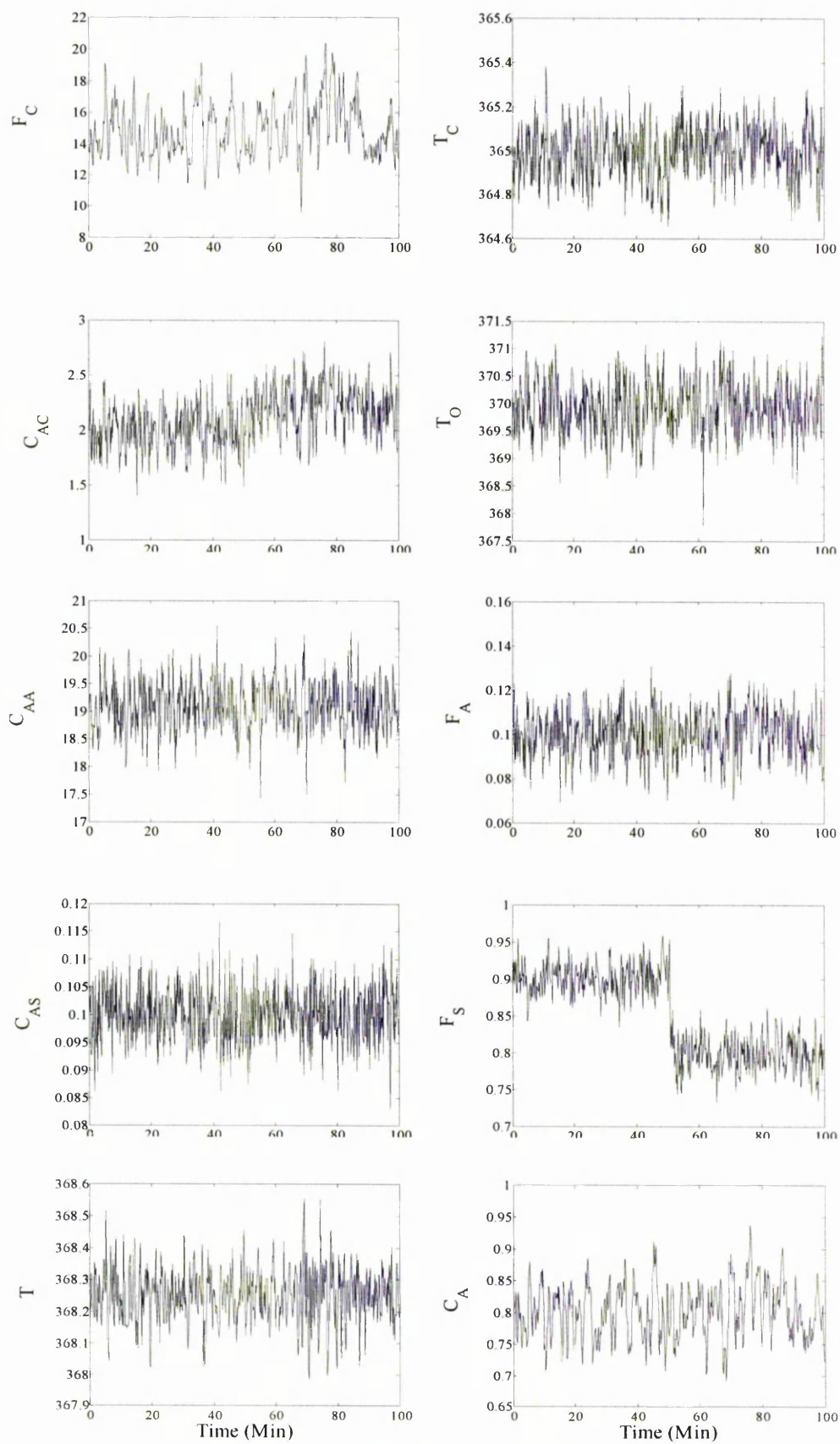


Figure 7.15 Process measurements for **Fault C**

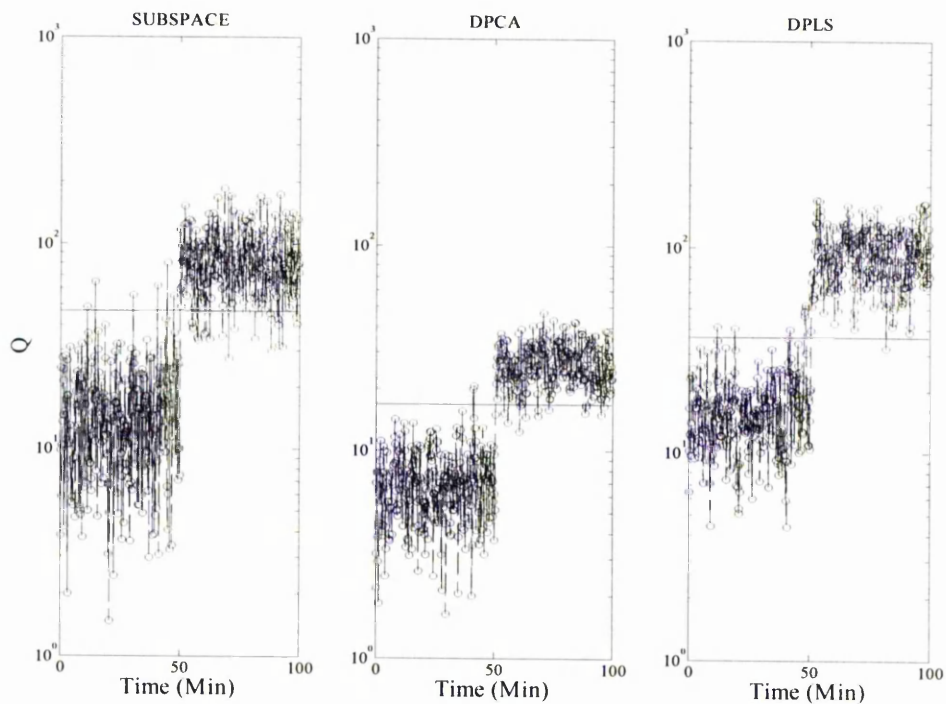


Figure 7.16 Q statistics. **FAULT C**

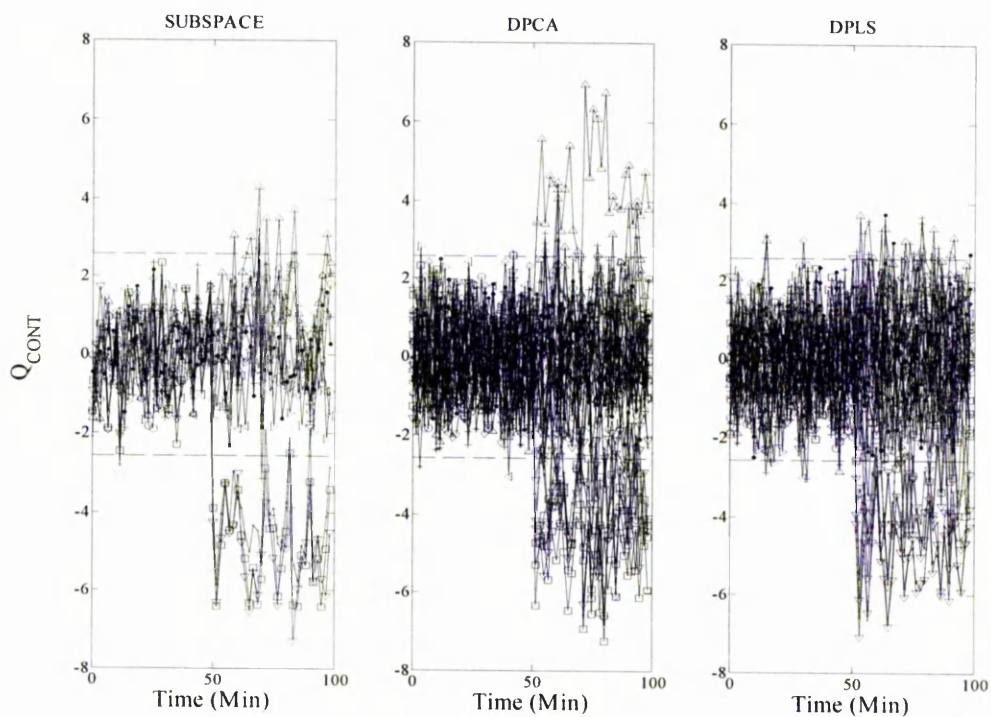


Figure 7.17 Contributions to Q statistic. **Fault C** F_S (∇); F_A (T); F_C (+); T_C (\circ); C_{AS} (\square); F_{AC} (x); C_{AA} (\triangleright); T_O (\triangleleft); C_{AC} (\triangle); T (\diamond); C_A (\bullet).

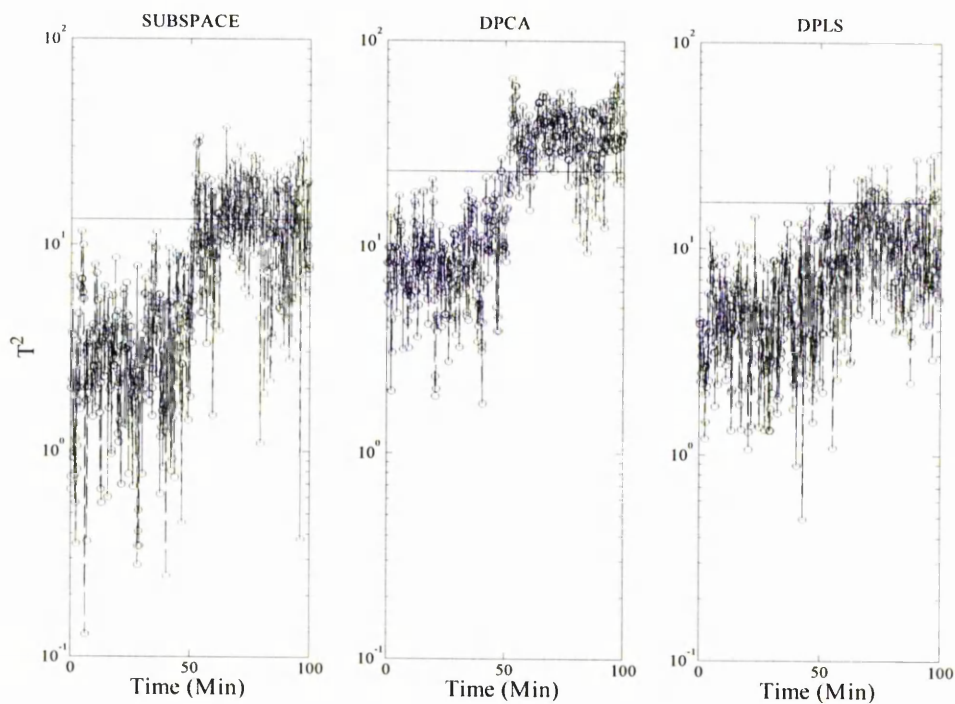


Figure 7.18 Hotelling's T^2 statistics. **FAULT C**

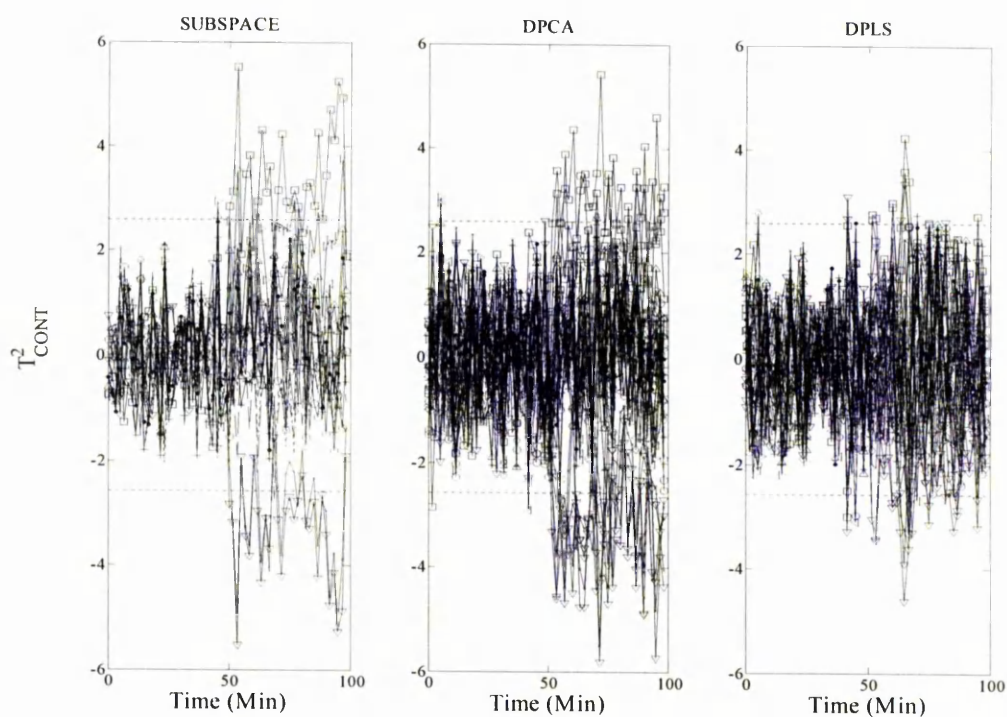


Figure 7.19 Contributions to T^2 statistic. **Fault C** F_S (∇); F_A (T); F_C ($+$); T_C (\circ); C_{AS} (\square); F_{AC} (\times); C_{AA} (\triangleright); T_O (\triangleleft); C_{AC} (\triangle); T (\diamond); C_A (\bullet).

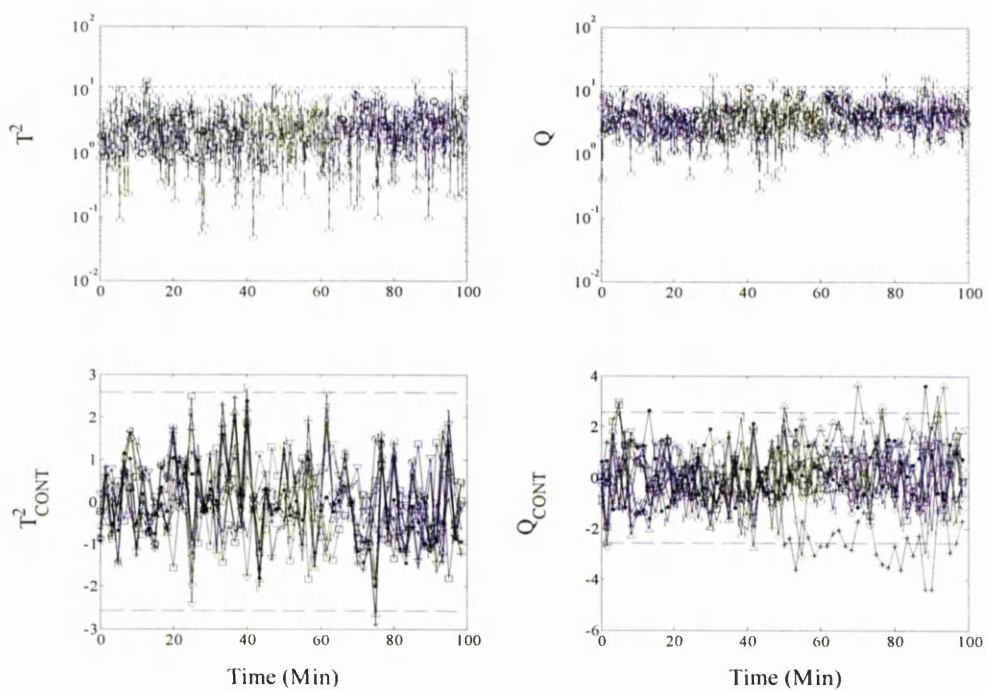


Figure 7.20 PCA T and Q Statistics Charts. **Fault A** F_S (∇); F_A (T); F_C (+); T_C (O); C_{AS} (\square); F_{AC} (x); C_{AA} (\triangleright); T_O (\triangleleft); C_{AC} (\triangle); T (\diamond); C_A (\bullet).

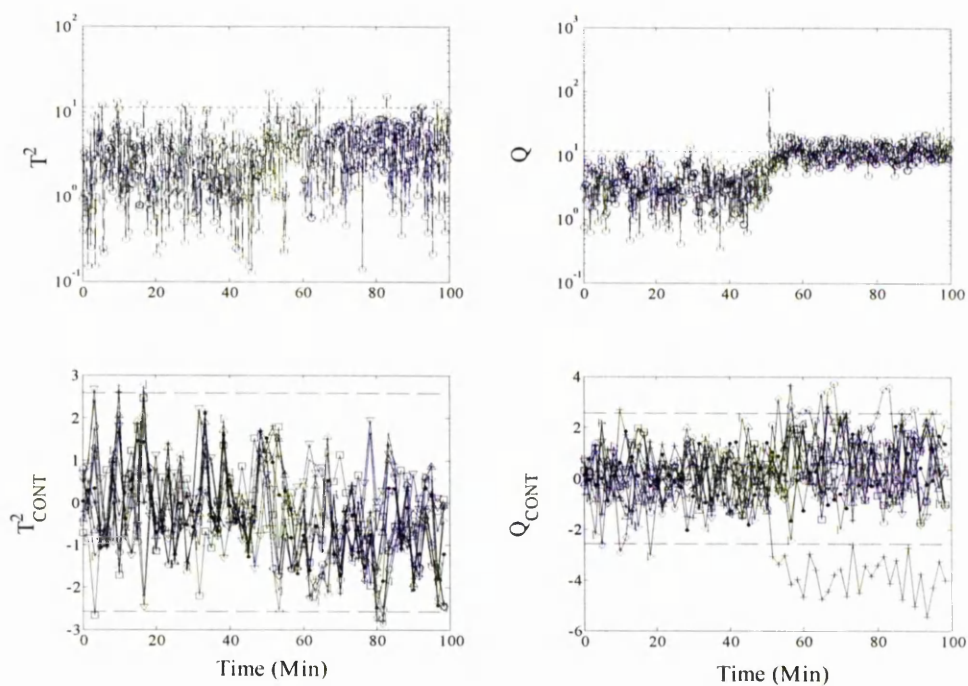


Figure 7.21 PCA T and Q Statistics Charts. **Fault B** F_S (∇); F_A (T); F_C (+); T_C (O); C_{AS} (\square); F_{AC} (x); C_{AA} (\triangleright); T_O (\triangleleft); C_{AC} (\triangle); T (\diamond); C_A (\bullet).

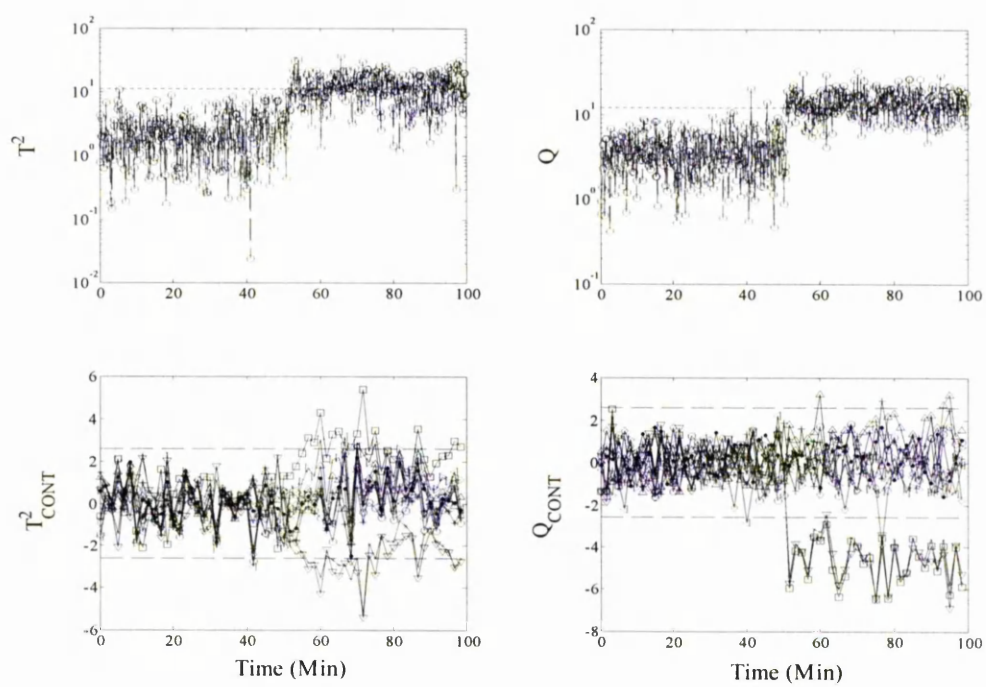


Figure 7.22 PCA T^2 and Q Statistics Charts. **Fault C** F_S (∇); F_A (T); F_C (+); T_C (\circ); C_{AS} (\square); F_{AC} (x); C_{AA} (\triangleright); T_O (\triangleleft); C_{AC} (\triangle); T (\diamond); C_A (\bullet).

Chapter 8

The Subspace Method and DPCA

The Subspace Method is similar in approach to dynamic PCA. Each of the two modelling approaches uses well-known linear parametric model structures to capture both static and dynamic information from the system. Recent studies have proved that under strict conditions, there is a basic equivalence in the two methods. In this chapter it is asserted that although the Subspace Method is a very close relation of dynamic PCA, important differences and possible advantages exist with the new method. These are demonstrated using simple simulation studies. A further simulation study is used to show that the Subspace Method, in common with dynamic PCA, may provide potential advantages over using PCA alone, sometimes providing a more robust model that is less prone to false alarms caused by dynamic transients that may exist within process data with autocorrelations.

8.1 Introduction

The relationship between the dimension reduction employed by DPCA and the Subspace Method is illustrated in the Figure 8.1 below, where the data matrix used in the DPCA analysis (LHS) is compared with that of the Subspace Method (RHS). The Subspace Method uses a state space model to capture the dynamics of the process, as opposed to DPCA, which uses an ARX model structure.

$$\begin{bmatrix} \mathbf{Y} \\ \mathbf{U} \\ \mathbf{Y}_{k-1} \\ \mathbf{U}_{k-1} \\ \vdots \\ \mathbf{Y}_{k-n_{DPCA}} \\ \mathbf{U}_{k-n_{DPCA}} \end{bmatrix} \rightarrow \begin{bmatrix} \mathbf{Y} \\ \mathbf{U} \\ \mathbf{X}_k \\ \mathbf{X}_{k+1} \end{bmatrix}$$

Figure 8.1 Model dimension reduction. The time-shifted process measurements of the ARX model structure used in DPCA (left side) are replaced by state sequences when applying the Subspace Method (right side).

A linear equivalence between the Subspace Method and DPCA is apparent because

- (1) Each model uses well-known linear parametric model structures, i.e. the Subspace Method uses a state space model structure, and DPCA uses an ARX model structure.
- (2) There is a direct linear equivalence between the state space model structure used by the Subspace Method and the ARX model structure used by DPCA.

The linear equivalence of the two modelling methods is proved in the next section, where it is demonstrated using a SISO system, that a state space structure can be transformed into an ARX model structure. It is also easy to show that the subspace method applies PCA to a data structure that conforms to a state space model of the system, in exactly the same way that DPCA is a PCA analysis of an ARX model structure (see Equations 8.6 and 8.7 below). In this respect, both the subspace method and DPCA may be regarded as versions of “dynamic” PCA because each method applies PCA to a set of vectors that conforms to a linear parametric model of the dynamic behaviour of the system.

As mentioned above, the Subspace Method uses a state space model structure and thereby reduces the number of variables required for the model, because a reduced number of states are able to capture the dynamics of the process. In contrast, the DPCA

model uses an ARX model structure. DPCA therefore uses a larger number of internal variables to model the process, i.e. the process measurements and their time-lagged counterparts. For large-scale MIMO processes, the number of internal variables used by DPCA may often be prohibitive. This may lead to cumbersome fault diagnosis due to the large number of time-lagged process measurements that contribute to the residual statistic Q .

Therefore, the Subspace Method might be considered to enjoy an advantage because fewer internal variables are required in the model. However, a possible drawback of the subspace method is that the non steady-state Kalman states that are identified in the modeling procedure have no real physical meaning and therefore contribute little meaning to any analysis using contributions plots. In fact, the state sequences used by the Subspace Method are “dummy” variables that are important only to the internal modeling process, and since no real physical meaning can be attached to them, they provide no indication regarding the origin of abnormal process behaviour.

In the next section, the basic equivalence of the model structures used by the Subspace Method and DPCA is proven. It is also demonstrated using a SISO system with random input that the Subspace Method is able to model the process using fewer principal components than DPCA. A SISO first order system is then simulated to demonstrate that although, under strict conditions, the respective methods have been proven to be equivalent [105], that the results obtained may differ considerably in the presence of an unmeasured disturbance.

A second simulation study is presented, which aims to build on from the work of Ku [91]. In [91], the DPCA approach is introduced and possible advantages are suggested with respect to PCA. The simulation study used by Ku is repeated here, and then the data is used to compare the results obtained using the Subspace Method, with respect to PCA and DPCA.

Finally a relatively simple, 2nd order, 2-input, 2-output simulation is presented to further examine the relationship between the Subspace Method and DPCA. A direct comparison of the contributions plots for each of the models reveals that although the models are in some ways equivalent, that when faced with a process disturbance, the

contributions of the measured process variables to the residual statistic Q , are significantly different.

8.2 Comparison of model structures

The Subspace Method is equivalent to DPCA in the sense that both DPCA and the Subspace Method are based on TLS [98-99] solutions to the error-in-variables (EIV) problem for well-known linear parametric model structures. Application of the Subspace Method involves finding a solution to an EIV formulation for a state space model structure. Application of DPCA involves finding a solution to an EIV formulation for a ARX model structure. The equivalence (subject to strict conditions regarding the measurement uncertainty) of the methods is due to the equivalence of the state space model structure used by the Subspace Method and the ARX model structure used by DPCA.

The equivalence of the two model structures is proved below. An EIV problem is then formulated for each of the model structures, leading to a direct comparison between the Subspace Method and DPCA.

Consider the following SISO state space system:

$$\begin{aligned}x_{k+1} &= ax_k + bu_k + e_{k+1} \\ y_k &= cx_k + du_k + f_k\end{aligned}\tag{8.1}$$

e_k and f_k are independent, zero mean white noise terms with variance σ_e and σ_f :

$$\begin{aligned}E\{e_i \cdot f_i\} &= 0 \\ E\{e_i \cdot e_j\} &= \sigma_e \delta_{ij} \\ E\{f_i \cdot f_j\} &= \sigma_f \delta_{ij}\end{aligned}\tag{8.2}$$

The state space model (Eq. 8.1) can be converted to an ARX model structure as follows:

$$x_k = \frac{1}{c} y_k - \frac{d}{c} u_k - \frac{1}{c} f_k\tag{8.3}$$

$$x_{k+1} = \frac{1}{c} y_{k+1} - \frac{d}{c} u_{k+1} - \frac{1}{c} f_{k+1} \quad (8.4)$$

i.e.

$$\frac{1}{c} y_{k+1} - \frac{d}{c} u_{k+1} - \frac{1}{c} f_{k+1} = a \left(\frac{1}{c} y_k - \frac{d}{c} u_k - \frac{1}{c} f_k \right) + b u_k + e_{k+1}$$

and

$$y_{k+1} = a y_k + d u_{k+1} + (bc - ad) u_k + g_{k+1} \quad (8.5)$$

where $g_{k+1} = f_{k+1} - a f_k + c e_{k+1}$.

Given the assumptions that are imposed on the noise terms, the variance of the noise on y_k is $\text{var}\{g_k\} = \text{var}\{f_k\} - a \text{var}\{f_{k-1}\} + c \text{var}\{e_k\} = (1-a)\sigma_f + c\sigma_e$.

The equivalence of the models used by the Subspace Method and DPCA has been proved by the straightforward algebraic manipulation that transforms the state space model (Eq. 8.1) into an ARX model structure (Eq. 8.5). The following EIV formulations described by Equations 8.6 and 8.7 describe the subsequent PCA analysis that is used in each of the two condition monitoring methodologies.

If we now assume that the input variables are also measured inaccurately, then the following two EIV problems can be formulated:

$$\begin{pmatrix} y_k \\ u_k \\ x_k \\ x_{k+1} \end{pmatrix} = \begin{pmatrix} \hat{y}_k \\ \hat{u}_k \\ \hat{x}_k \\ \hat{x}_{k+1} \end{pmatrix} + \begin{pmatrix} f_k \\ h_k \\ e_k \\ e_{k+1} \end{pmatrix} \quad (8.6)$$

$$\begin{pmatrix} y_{k+1} \\ y_k \\ u_{k+1} \\ u_k \end{pmatrix} = \begin{pmatrix} \hat{y}_{k+1} \\ \hat{y}_k \\ \hat{u}_{k+1} \\ \hat{u}_k \end{pmatrix} + \begin{pmatrix} f_{k+1} \\ f_k \\ h_{k+1} \\ h_k \end{pmatrix} \quad (8.7)$$

where h_k represents the measurement uncertainty of the input variable u_k . Note that Equations 8.6 and 8.7 contain the same number of process variables. However, in the case of large-scale industrial processes with highly correlated process measurements, significant dimension reduction can be achieved when applying PCA to Eqs. 8.6 and 8.7

where the dimension of both process input and process output variables can be reduced without significant loss of accuracy. Furthermore, the dynamic relationships between the input and output variables admit to further dimension reduction using the Subspace Method, where relatively few (orthogonal) state sequences are used in the model, in place of the time-lagged process variables used in DPCA.

The Subspace Method and DPCA are equivalent if the measurement uncertainties that are imposed in the process variables are independent and identically distributed [105]. However the Subspace Method generally provides a condition monitor using fewer principal directions, as demonstrated using the following simple analysis. Consider the following deterministic system

$$\begin{aligned} x_{k+1} &= ax_k + bu_k \\ y_k &= cx_k + du_k \end{aligned} \quad (8.8)$$

Equation 8.8 corresponds to the following ARX model structure:

$$y_{k+1} = ay_k + du_{k+1} + (bc - ad)u_k \quad (8.9)$$

Assuming the system is stable and the input sequence u_k to be a random sequence, then a PCA analysis applied to Eq. 8.8 reveals that the Subspace Method fully describes the process using two principal directions, i.e. both y_i and x_{i+1} are linear combinations of x_i and u_i :

$$\begin{pmatrix} y_k \\ u_k \\ x_k \\ x_{k+1} \end{pmatrix} = \begin{pmatrix} c & d \\ 0 & 1 \\ 1 & 0 \\ a & b \end{pmatrix} \begin{pmatrix} x_k \\ u_k \end{pmatrix} \quad (8.10)$$

where Eq. 8.10 corresponds to $Z = PT$, the two principal directions are described by

the sequences x_k and u_k , with loading matrix $\begin{pmatrix} c & 0 & 1 & a \\ d & 1 & 0 & b \end{pmatrix}^T$.

Equation 8.10 also defines the maximum number of dimensions required by the Subspace Method analysis for large-scale MIMO processes, i.e. the maximum number of dimensions required is:

$$\dim_{\max} \leq \dim(U) + \dim(X) \quad (8.11)$$

where U is a vector containing the process inputs and the matrix X contains the state sequences.

In contrast, DPCA is based on an ARX model structure and requires three principal directions to describe the same process:

$$\begin{pmatrix} y_{k+1} \\ y_k \\ u_{k+1} \\ u_k \end{pmatrix} = \begin{pmatrix} a & d & h \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} y_k \\ u_{k+1} \\ u_k \end{pmatrix} \quad (8.12)$$

where Eq. 8.12 corresponds to $Z = PT$, the three principal directions are described by

the sequences y_k , u_{k+1} and u_k , with loading matrix $\begin{pmatrix} a & 1 & 0 & 0 \\ d & 0 & 1 & 0 \\ h & 0 & 0 & 1 \end{pmatrix}^T$.

As proved in [105], the Subspace Method and DPCA are equivalent if the measurement uncertainties that are imposed in the process variables are independent and identically distributed (i.i.d). Although the states are a linear combination of the past inputs and outputs (up to within a similarity transform), the use of SVD to describe the process using new orthogonal directions (i.e. the state sequences) may yield significantly different results when the measurement uncertainty is anything but i.i.d. (as is always the case in industry). The two methods can produce significantly different results with regards to the contributions analysis, as was found in the CSTR analysis of Chapter 7 and as is demonstrated in the simulation studies that follow.

8.3 Simulation Studies

Li and Qin [105], proved that if the measurement uncertainties of the process variables are i.i.d., that an EIV approach based on state space and ARX structures is equivalent. However, the three simulation studies that follow demonstrate that if the i.i.d condition is not satisfied, then differences exist between the results obtained using the two modelling methods.

8.3.1 Simulation Study 1: A Deterministic 1st order system

A 1st order deterministic system was used to generate 2000 points of training data:

$$\begin{aligned} x_{k+1} &= 0.2x_k + 0.7u_k \\ y_k &= 0.7x_k + 0.05u_k \end{aligned} \quad (8.13)$$

By applying Eq. 8.9, the state space model is transformed to an equivalent ARX model structure, i.e.

$$y_{k+1} = 0.2y_k + 0.05u_{k+1} + 0.48u_k \quad (8.14)$$

The system is driven by the random input u_k , with the inputs u_k , the state sequence x_k , and the output y_k all measured exactly.

The application of the Subspace Method and DPCA involves the construction of training data matrices, $Z_{SUBSPACE}$ and Z_{DPCA} , as follows

$$Z_{SUBSPACE} = \begin{pmatrix} Y_{1,N-1} & U_{1,N-1} & X_{1,N-1} & X_{2,N} \end{pmatrix}$$

$$Z_{DPCA} = \begin{pmatrix} Y_{2,N} & Y_{1,N-1} & U_{2,N} & U_{1,N-1} \end{pmatrix}$$

with

$$Y_{2,N} = \begin{pmatrix} y_2^T & y_3^T & \cdots & y_N^T \end{pmatrix}^T$$

and similar notation applying for the process inputs U and the state sequences X .

Each of the columns of $Z_{SUBSPACE}$ and Z_{DPCA} were scaled to zero mean and unit variance to form $\bar{Z}_{SUBSPACE}$ and \bar{Z}_{DPCA} . A PCA analysis was then applied to obtain the scores and loadings, i.e.

$$Z = T_M \Phi_M^T, \quad (8.15)$$

where M is the number of PC's required to model the system, and Φ_M contains the loadings that constitute the model for normal operation. In accordance with Equations

8.10 and 8.12, the system is fully described by 2 PCs in the case of the Subspace Method and 3 PCs in the case of DPCA.

A second sequence of 1000 data points was generated with an unmeasured unit step disturbance applied to the input u_k , from data point 300 to data point 600 (see Figure 8.1). The change in y_k , which increases significantly between sample points 300 and 600 is measured, however the change in u_k (also shown in Figure 8.1) is not. Each of the two models was then used to analyse the data containing the unmeasured disturbance. The results are illustrated in Figures 8.2-8.6.

Discussion of results from simulation 1

Figure 8.2 shows the Hotelling's T^2 and Q statistics for the Subspace Method and DPCA. The results appear to be similar for each of the models. Between sample points 300 and 600 (in both cases) the T^2 statistic shows a slight rise but does not consistently violate the 99% confidence limit. For the process model residual statistic Q , the process upset is clearly indicated between sample points 300 and 600.

Figure 8.3 shows the T Scores and Loadings for the Subspace Method model. The process upset has affected both T scores. The negative loadings on PC#1 have driven the associated T score negative between sample points 300 and 600. For PC#2, since only the change in y_k is measured, (the change in u_k is not measured), it is the positive loading on the variable y_k that has driven the T score positive. In each case, the 99% confidence limits are violated at several points, however the T scores generally stay within bounds.

Figure 8.4 shows the T Scores and Loadings for the DPCA model. The T score on PC#1 has been driven negative by the negative loadings on all 4 variables, however it does not permanently cross the 99% confidence limit. The T score for PC#3 has also been driven negative towards the 99% confidence limit, by the effect of the negative loading on y_k .

Figure 8.5 shows a comparison of contributions to the model residual statistic, Q , for the Subspace Method (left) and DPCA (right). Contributions charts are considered important in the aid of fault diagnosis because they give an indication of the likely origin of process abnormalities. Figure 8.5 illustrates the difference in the residuals

contributions associated with each of the models. Both charts map the contributions of the measured process variables $y_k(\bullet)$, and $u_k(x)$. However, both charts also show contributions from “dummy” variables – variables that are used in the modelling process, but have no real physical meaning, i.e. the state sequences in the case of the Subspace Method, and the time-lagged input and output variables in the case of DPCA. In Figure 8.5, similar and large contributions between sample points 300 and 600 are indicated for the Subspace Method “dummy” variable $x_{k+1}(\triangleright)$ and the DPCA “dummy” variable $y_{k+1}(\diamond)$. (The relationship between these variables is explored further below). The contribution generated by $u_k(x)$ (which is the process variable most closely associated with the process upset) is clearly indicated on each of the contributions charts. In terms of the contributions of the external process variables, it appears that the Subspace Method has provided a clearer indication of the unmeasured disturbance on u_k , because the u_k residual is far more significant than the y_k residual, which is of the order 10^{-3} . In contrast, the DPCA analysis also shows a large residual generated by the process output $y_k(\bullet)$. There may also be ambiguity in the DPCA analysis due to the large negative residual generated by $y_{k+1}(\diamond)$, yet there is a significant positive residual generated by $y_k(\bullet)$.

The magnitude of the contributions from the process variables to Q is governed by the relationship

$$\mathbb{C}_Q = \mathbf{Z}(\mathbf{I}_n - \Phi_M \Phi_M^T) \quad (8.16)$$

where \mathbf{Z} corresponds to $Z_{SUBSPACE}$ or Z_{DPCA} , Φ_M is a matrix containing the respective loadings, and \mathbf{I} is an identity matrix of dimension n , n is the number of variables in \mathbf{Z} , and M is the number of PCs. The contributions from each of the model variables to the model residual statistic Q are calculated from the symmetric matrix given by the quantity $\mathbf{I}_n - \Phi_M \Phi_M^T$. Figure 8.6 provides numerical values for the model given by Eq. 8.16, governing how the contributions are generated for each of the two models.

The process residuals are calculated using scaled process data (the scaling factors were calculated using the reference data, where each of the process measurement sequences

is scaled to zero mean and unit standard deviation). The scaling factors are shown in Table 8.1 below, detailing the mean(μ) and standard deviation(σ) of each of the model variables calculated from the reference data set.

		y_k	u_k	x_k	x_{k+1}	y_{k-1}	u_{k-1}
SM	μ	0.0007	0.0018	0.0009	0.0009		
	σ	0.2906	1.0007	0.4087	0.4087		
DPCA	μ	0.0007	0.0018			0.0007	0.0018
	σ	0.2906	1.0007			0.2906	1.0007

Table 8.1 The scaling factors used for the Subspace Method and DPCA analysis.

In Figure 8.5, it can be seen that the contributions of the Subspace Method variable $x_{k+1}(\triangleright)$ and the DPCA variable $y_{k+1}(\diamond)$ appear to be very similar in the respective contributions charts. The two models are compared directly (below) in regards to the contributions generated for these two variables. The model residual is calculated using scaled process variables, i.e. (from Figure 8.6)

$$_{SUBSPACE} \mathbb{C}_{x_{k+1}} \cong 0.52\bar{x}_{k+1} - 0.49\bar{u}_k - 0.09\bar{y}_k - 0.01\bar{x}_k \quad (8.17)$$

$$_{DPCA} \mathbb{C}_{y_{k+1}} \cong 0.52\bar{y}_{k+1} - 0.48\bar{u}_k - 0.10\bar{y}_k - 0.09\bar{u}_{k+1}$$

Because the data is auto-scaled, each of the variables is zero mean and unit variance. This allows a direct comparison of each of the coefficients in the expression Eq. 8.17, where it can be seen that the corresponding coefficients for the Subspace Method variable $x_{k+1}(\triangleright)$ and the DPCA variable $y_{k+1}(\diamond)$ are very nearly the same.

The similarity between these residuals derives from the close linear relationship described exactly by Equation 8.4, i.e.

$$x_{k+1} = 1.43y_{k+1} - 0.07u_{k+1} \quad (8.18)$$

Figure 8.5 also shows that the contribution generated for the variable $u_k(\times)$ is very similar for each of the models. This is also indicated in Figure 8.6 where Eq. 8.16 yields the following expressions for each of the two modelling methods:

$$_{SUBSPACE} \mathbb{C}_{u_k} \cong 0.48u_k - 0.49x_{k+1} + 0.09x_k + 0.00y_k \quad (8.19)$$

$$_{DPCA} \mathbb{C}_{u_k} \cong 0.44u_k - 0.48y_{k+1} + 0.08u_{k+1} + 0.10y_k$$

For each of the models, it can be seen that the size of the residual is dictated in a similar way by the positive influence of the input u_k and the negative influences of the corresponding variables x_{k+1} and y_{k+1} . In Figure 8.5, the magnitude of the DPCA contribution for u_k is slightly larger than for the Subspace Method. Although the Subspace Method gain on u_k is larger than that for DPCA (0.48 as opposed to 0.44), the step increase in u_k is unmeasured, and therefore it is the larger DPCA gain on y_k (0.10 as opposed to 0.00) that is responsible.

The different model structures (Eqs. 8.13 and 8.14) used by the Subspace Method and DPCA are reflected in the quantity $\mathbf{I}_n - \Phi_M \Phi_M^T$. For example in Figure 8.6, the Subspace Method contribution generated for y_k (column 1), depends on a large positive gain on the quantity \bar{y}_k (0.5197) that is opposed by the large negative gain for \bar{x}_k (-0.4906). In column 2, the contribution for \bar{u}_k is characterised by a large positive gain on \bar{u}_k (0.4770) that is directly opposed by the large negative gain on \bar{x}_{k+1} (-0.4904). Similar relationships exist in columns three and four, all of which relate directly to Eq. 8.13 where y_k depends directly on x_k , and x_{k+1} depends directly on u_k , i.e.

$$\begin{aligned} x_{k+1} &= 0.2x_k + 0.7u_k \\ y_k &= 0.7x_k + 0.05u_k \end{aligned} \quad (8.20)$$

In respect to the DPCA model, the contribution of each of the process residuals is governed by strong and opposite gains on the quantities \bar{y}_k and \bar{x}_{k-1} , which reflect the ARX model structure for the system, described by Eq. 8.14.

8.3.2 Simulation Study 2: Auto-correlated process data

Simulation study 2 is aimed at providing an example of where the use of dynamic model structures such as the Subspace Method and DPCA may lead to a clearer picture of the likely origin of process abnormalities. In such situations, the widely applied PCA method might be used in conjunction with the Subspace Method to obtain more information regarding the process.

The PCA model works on the assumption that the system is static, i.e. the process variables are independent in time, however many processes are driven by random noise and uncontrollable disturbances [91]. The measured process variables may contain

dynamic characteristics of the process, and may exhibit a degree of autocorrelation. To accommodate the autocorrelated nature of the data, Ku [91] suggested integrating the ARX model structure used in system identification of time-series models into the PCA analysis. This leads to a DPCA model that contains time-shifted process variables and makes it appropriate for modelling autocorrelated data.

The connection has been made between the Subspace Method and DPCA, where it was outlined that each method applies a PCA analysis to data matrices that conform to well known linear model structures for system identification. The subspace method is therefore a likely candidate for modelling autocorrelated data, just as was demonstrated in the paper by Ku [91], using DPCA.

The simulation presented here is based on [91]. The input $\mathbf{w}(k)$ is a zero mean random sequence with variance 1. The inputs $\mathbf{u}(k)$ and the outputs $\mathbf{y}(k)$ are measured, however $\mathbf{z}(k)$ and $\mathbf{w}(k)$ are not. The output $\mathbf{y}(k)$ is subject to a random noise sequence $\mathbf{v}(k)$, with zero mean and variance 0.1.

A sequence of 1000 samples was created to train each of the models and to calculate the confidence limits. The data containing the unmeasured disturbance also consisted of 1000 samples, with a unit step disturbance in $\mathbf{w}(k)$ introduced from sample 300 through to sample 600. The equations describing the simulation are as follows

$$\mathbf{u}(k) = \begin{pmatrix} 0.811 & -0.226 \\ 0.477 & 0.415 \end{pmatrix} \mathbf{u}(k-1) + \begin{pmatrix} 0.193 & 0.689 \\ -0.320 & -0.749 \end{pmatrix} \mathbf{w}(k-1) \quad (8.21)$$

$$\mathbf{z}(k) = \begin{pmatrix} 0.118 & -0.191 \\ 0.847 & 0.264 \end{pmatrix} \mathbf{z}(k-1) + \begin{pmatrix} 1 & 2 \\ 3 & -4 \end{pmatrix} \mathbf{u}(k-1) \quad (8.22)$$

$$\mathbf{y}(k) = \mathbf{z}(k) + \mathbf{v}(k) \quad (8.23)$$

The objective is to compare the performance of the dynamic and static models in the presence of an unmeasured disturbance. A very important part of the monitoring process is the decision regarding how many principal components to include in the model. Several methods for determining the proper number of principal components to include in the monitor have been suggested; see for example [91]. However as noted in [91],

variability from process to process, or even between operating regions means that no single method for determining the number of latent variables is foolproof, suggesting that it may be prudent to apply several approaches in an effort to reach a consensus.

5 principal components and 2 principal components were chosen to model the system with the DPCA and PCA models respectively, on the basis of the models used in the Ku analysis [91], where the data vector used to build the DPCA model is in the form of $(Y_{2,N} \ Y_{1,N-1} \ U_{2,N} \ U_{1,N-1})$. In the case of the Subspace Method, 2 principal components were used to model the system.

The results are shown in Figures 8.7 – 8.9. Figure 8.7 shows the T^2 statistics for the three methods. For all three models, the 99% confidence limit for the T^2 statistic is violated as indicated by the sharp rise in the T^2 statistic from sample 300 – sample 600. The step change in $\mathbf{w}(k)$ has driven the T^2 scores past the control limits. This may be explained by the extra power brought by the step increase in the random vector driving the system input, which drives the system leading to an increase in the magnitude of the principal components.

Figure 8.8 shows the Q statistics for the three methods. The $PCA-Q$ statistic passes the 99% confidence limit, indicating that a fault, or a process abnormality has occurred within the system. The $PCA-Q$ statistics contributions chart in Figure 8.9 suggests that the process abnormality can be associated with y_1 (\diamond) and u_1 (+). In contrast, the Q statistics for the subspace method and DPCA remain below the control limits.

On the basis of the Q statistics comparison, a possible conclusion is that the unmeasured disturbance has caused a false alarm in the $PCA-Q$ monitor. In contrast, the subspace and DPCA model structures are able to incorporate the autocorrelation into their dynamic model structure, and therefore only the T^2 statistic passes the control limit. This simple example suggests that under certain circumstances dynamic model structures such as the Subspace Method and DPCA bring extra information to the analysis, and therefore may provide an advantage over using PCA alone, in that they provide a more robust approach to process monitoring, leading to fewer false alarms in certain circumstances, such as if a system encounters trivial dynamic transients.

8.3.3 Simulation Study 3: A Deterministic 2nd order system

The 2nd order system is illustrated in Figure 8.10. The system has two random inputs u_k and v_k . The two system outputs are g_k and h_k . Both outputs are subject to white measurement noise. The system output g_k has been appended with a normally distributed random sequence with variance $0.2\% \text{ var}(g_k)$. The system output h_k has been appended with a normally distributed random sequence with variance $0.05\% \text{ var}(h_k)$

There are two measured outputs g and h , described by the output equations

$$g_k = 0.2x_k - 0.2y_k + 0.9u_k + 1.9v_k \quad (8.24)$$

$$h_k = 0.8x_k + 0.2y_k + 2.3u_k - 3v_k \quad (8.25)$$

Two unmeasured internal states x and y are driven by measured inputs u and v :

$$y_{k+1} = 0.8x_k - 0.2y_k + 0.2u_k + 0.7v_k \quad (8.26)$$

$$x_{k+1} = 0.5x_k + 0.4y_k - 0.3u_k - 0.7v_k \quad (8.27)$$

The deterministic part of the system is described exactly using the subspace algorithm \mathcal{M}_1 , which identified a 2nd order state space model with state matrices

$$A = \begin{pmatrix} 0.7412 & -0.2131 \\ 0.3751 & 0.4588 \end{pmatrix}, \quad B = \begin{pmatrix} -0.1354 & -0.4340 \\ 0.0965 & 0.2066 \end{pmatrix}$$

$$C = \begin{pmatrix} -0.2008 & 0.2126 \\ -0.3586 & -0.2327 \end{pmatrix}, \quad D = \begin{pmatrix} 0.4292 & 0.8894 \\ 0.6000 & -0.7682 \end{pmatrix}$$

The Matlab System Identification Toolbox was used to identify the following ARX model that also describes the deterministic part of the system exactly. The system is described by the 2nd order ARX model structure

$$\begin{aligned} g_k &= 1.2g_{k-1} - 0.42g_{k-2} + 0.4292u_k - 0.4673u_{k-1} + 0.1459u_{k-2} + 0.8894v_k - 0.9362v_{k-1} + 0.2753v_{k-2} \\ h_k &= 1.2h_{k-1} - 0.42h_{k-2} + 0.6u_k - 0.6939u_{k-1} + 0.2656u_{k-2} - 0.7682v_k + 1.0294v_{k-1} - 0.3047v_{k-2} \end{aligned} \quad (8.28)$$

With the knowledge that the above 2nd order state space and ARX model structures fully describe the deterministic part of the system, appropriate data matrices for the PCA analysis were constructed as follows.

For the Subspace Method, two state variable sequences are included:

$$X_k = (x_k \quad x_{k+1} \quad x_{k+2} \quad \dots)$$

$$Y_k = (y_k \quad y_{k+1} \quad y_{k+2} \quad \dots)$$

$$\mathbf{Z}_{k, \text{SUBSPACE}} = [g_k \quad h_k \quad u_k \quad v_k \quad x_k \quad y_k \quad x_{k+1} \quad y_{k+1}].$$

For DPCA, a 2nd order ARX structure is used, with each row of \mathbf{Z}_{DPCA} composed as follows

$$\mathbf{Z}_{k, \text{DPCA}} = [g_k \quad h_k \quad g_{k-1} \quad h_{k-1} \quad g_{k-2} \quad h_{k-2} \quad u_k \quad v_k \quad u_{k-1} \quad v_{k-1} \quad u_{k-2} \quad v_{k-2}]$$

Note that for the Subspace Method there are 8 columns of data used in the PCA model, and for DPCA there are twelve columns of data used.

The simulation was run to generate 2000 points of training data. The process monitoring data was generated by subjecting the system to a disturbance d_k , that arrives in the form of an unmeasured step input to u_k , from sample 300 to sample 600. Figure 8.11 shows the system measurements. The two random inputs u_k and v_k are shown with the effect of the step disturbance on the input u_k clearly indicated. The effect of the step disturbance has filtered through to the system outputs g_k and h_k as can be seen in the respective subplots.

Discussion of results for Simulation 3

Figure 8.12 shows an eigenvalue plot for the Subspace Method model. It can be seen that the subspace model describes the system using 4 principal components. Figure 8.13 shows an eigenvalue plot for the DPCA model. The DPCA model describes the system using 6 principal components. These figures demonstrate that the Subspace Method reduces the dimensionality of the problem by using a reduced number of (orthogonal) state sequences to describe the input-output relationships of the system. In fact, as

proven using the SISO system described above, and in each of the simulations 1,2 and 3, it has been found that the Subspace Method requires fewer principal components to model the system than DPCA.

Figure 8.14 compares the T^2 statistics for each of the two models. In each case, the T^2 measure of the process remains within the bounds of normal operation.

Figure 8.15 shows the process residuals produced by the subspace model. The appearance of the process upset is clearly indicated between sample 300 and sample 600, where the magnitude of Q clearly violates the 95% confidence limit. Figure 8.16 shows the process residuals produced by the DPCA model. The appearance of the process upset is clearly indicated between sample 300 and sample 600, where the residuals statistic clearly violates the 95% confidence limit. A comparison of Figures 8.15 and 8.16 reveals that for the training data set, the confidence limit for the Subspace Model Q statistic is markedly less than that for DPCA. This is most likely due to the magnitude of the 7th principal component in Figure 8.13, which was excluded from the DPCA model. In addition, the number of variables used to generate the residual Q is greater in the case of DPCA (12 as opposed to 8).

Figure 8.17 shows the contributions of each of the Subspace Method model variables to the residual statistic Q . The contribution of each of the process variables has been scaled to unit variance with reference to the training data set. Several of the model residuals violate the confidence limit, including the measured outputs g_k (\diamond) and h_k (x), the process input u_k (\triangleleft) and the state sequences x_k (\bullet), y_k (o), and y_{k+1} ($*$). The large contribution of the process input u_k (\triangleleft) to the residual statistic provides an indication of the origin of the process upset. However the dynamic nature of the process is indicated by several other important contributions from variables not directly responsible for the upset, leading to ambiguity in the analysis. Note that there is a significant contribution from the process output $h_k(x)$, that is a direct consequence of the relationship described by Eq. 8.25, where it can be seen that h_k depends largely on the value of u_k .

Figure 8.18 shows the contributions of each of the DPCA model variables to the DPCA residuals statistic Q . Again, the contribution of each of the process variables has been scaled to unit variance with reference to the training data set. Ignoring the lagged terms,

which are essential to the modelling process, but have no real physical meaning, the main contributions are the process outputs $g_k(\diamond)$ and $h_k(x)$ and the process input $u_k(\triangleleft)$. The large contribution of the process input $u_k(\triangleleft)$ to the residual statistic provides an indication of the origin of the process upset. However, as with the subspace analysis, the dynamic nature of the process is indicated by several other important contributions from variables not directly responsible for the upset, leading to ambiguity in the analysis. Note that there is more significant contribution from the process output $g_k(\diamond)$ than for the process output $h_k(x)$, which at first glance seems at odds with the relationships described by Equations 8.24 and 8.25, where it can be seen that h_k depends more largely on the value of u_k .

Professor Kalman [12, 26], who was instrumental in the introduction of the concept of state space, stressed that the states of the system generally have no real physical meaning. This raises a problem when applying the contributions analysis, as there is a need to interpret the meaning of contributions from the states to Q . Consequently, in terms of fault diagnosis using the Subspace Method, one solution is to ignore the “meaningless” state variables and consider only the measured process inputs and outputs in the contributions analysis. Therefore, it may (in some cases) be prudent to exclude them from the contributions charts. Fault diagnosis is then centred solely on the measured variables of the process. Such an approach makes sense considering that anything “unmeasured or meaningless” cannot be detected as the origin of a fault. Note that these “dummy” variables still contribute to both the Hotelling’s T statistic and the Q statistic. It may also be worth noting that when using DPCA, there is sometimes ambiguity in the interpretation of the contributions from the time-lagged variables in Z_{DPCA} . (Neither the states used in the state space formulation, nor the time-lagged process variables used in the DPCA formulation are directly measured, therefore both might be considered as internal variables only).

Figure 8.19 excludes any variables that are not measured, providing a direct comparison of the measured variables (with real physical meaning) that contribute to the process residual statistics for each of the two models. In both cases, the main contributions come from the process outputs $g_k(\diamond)$ and $h_k(x)$ and the process input $u_k(\triangleleft)$. However

there is a distinct difference, because the contribution of h_k is far more significant in the case of the Subspace Method.

Li and Qin [105] claim that the Subspace Method and DPCA may be regarded as versions of “dynamic” PCA. Each method applies a PCA analysis to a set of vectors that conforms to a linear parametric model of the dynamic behaviour of the system. Li and Qin [105] have proven that EIV approaches using state space and ARX model structures, i.e. the Subspace Method and DPCA algorithms are equivalent if the measurement uncertainties that are imposed in the process variables are independent and identically distributed (i.i.d.). They also show in [105] that when the measurement and/or process noise is not i.i.d, or in the presence of a process disturbance, the subspace approach can yield consistent results where DPCA does not. Further to this, in Simulations 1 and 3 above, where unmeasured disturbances affect a system, it has been demonstrated that a significant difference in the outcome of the two approaches may result, as shown in Figure 8.19. The dynamic nature of any process can lead to an ambiguous contributions analysis, therefore alternative model structures might be used in conjunction to gain alternative viewpoints of the same event.

8.4 Conclusion

Dynamic models for MSPC such as the well-known DPCA method, and the more novel Subspace Method are aimed at capturing both static information and dynamic information from the system. The Subspace Method and DPCA may be regarded as versions of “dynamic” PCA because each method applies PCA to a set of vectors that conforms to a linear parametric model of the dynamic behaviour of the system. In this chapter, a direct comparison of the Subspace Method and DPCA has been made.

The linear equivalence between the state space model structure used by the Subspace Method and the ARX model structure used by DPCA has been proven. It has also been proven that a SISO system may be monitored using fewer principal components using the Subspace Method than DPCA. This is only proven on the basis of the SISO analysis, however an extension to the MIMO situation follows logically. To this end, a simple MIMO system has been used to show that again, the Subspace Method provides a lower dimension dynamic condition monitor than DPCA. For industrial systems where process measurements are higher correlated, the advantage enjoyed by the Subspace

Method regarding the ability to monitor in fewer dimensions is expected to be even greater.

A disadvantage of the Subspace Method is that the states of the system generally have no real physical meaning. This raises a problem when applying a contributions analysis to diagnose the origin of process abnormalities, as there may be the need to interpret the meaning of contributions from the states to the model residual Q . Unfortunately the state variables are “meaningless” and must be ignored in terms of interpreting contributions charts. Such an approach seems prudent because anything unmeasured cannot be detected as the origin of a fault. In fact, it has been found in all the work reported in this thesis (where simple, easy-to-detect faults were generated), that it is the measured variables that manifest themselves as the principal contributors to the residual Q , with the states contributing very little.

As proved in [105], the Subspace Method and DPCA are equivalent if the measurement uncertainties that are imposed in the process variables are independent and identically distributed (i.i.d). Although the states are a linear combination of the past inputs and outputs (up to within a similarity transform), the use of SVD to describe the system using new orthogonal directions (i.e. the state sequences), may yield significantly different results when the measurement uncertainty is anything but i.i.d. (as is always the case in industry). The simulation studies show that the two methods can produce significantly different results with regards to the contributions analysis. For example, if unmeasured disturbances affect a system, it has been demonstrated that a difference in the outcome of the two approaches may result. However, for fault diagnosis, each of the model structures could be used in conjunction to gain alternative viewpoints of the same event.

A simulation inspired by the paper by Ku [91] has been used to demonstrate a potential advantage of using dynamic models such as the Subspace Method and DPCA. It is proposed that under certain circumstances dynamic model structures such as the Subspace Method and DPCA bring extra information to the analysis. They may therefore provide an advantage over using PCA alone, in that they provide a more robust approach to process monitoring, leading to fewer false alarms if a system encounters trivial dynamic transients.

8.5 Figures

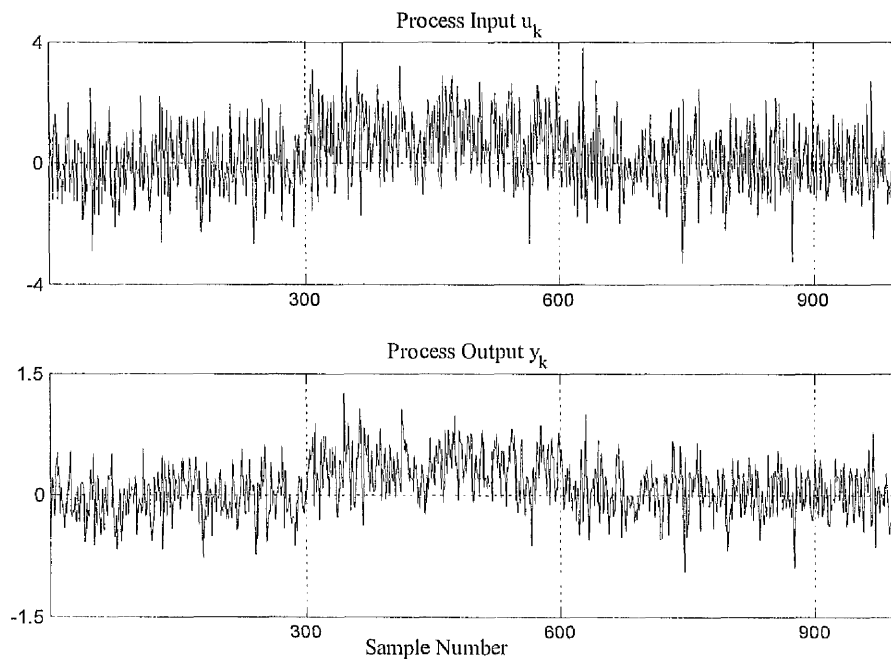


Figure 8.1 SISO process input u_k and process output y_k .

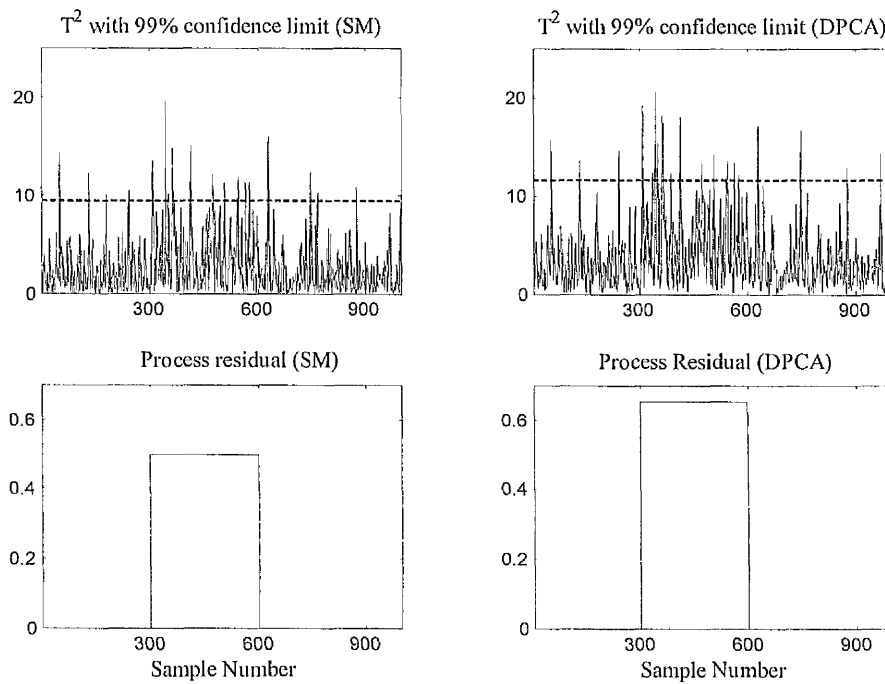


Figure 8.2 T² and Q statistics for Subspace Method and DPCA.

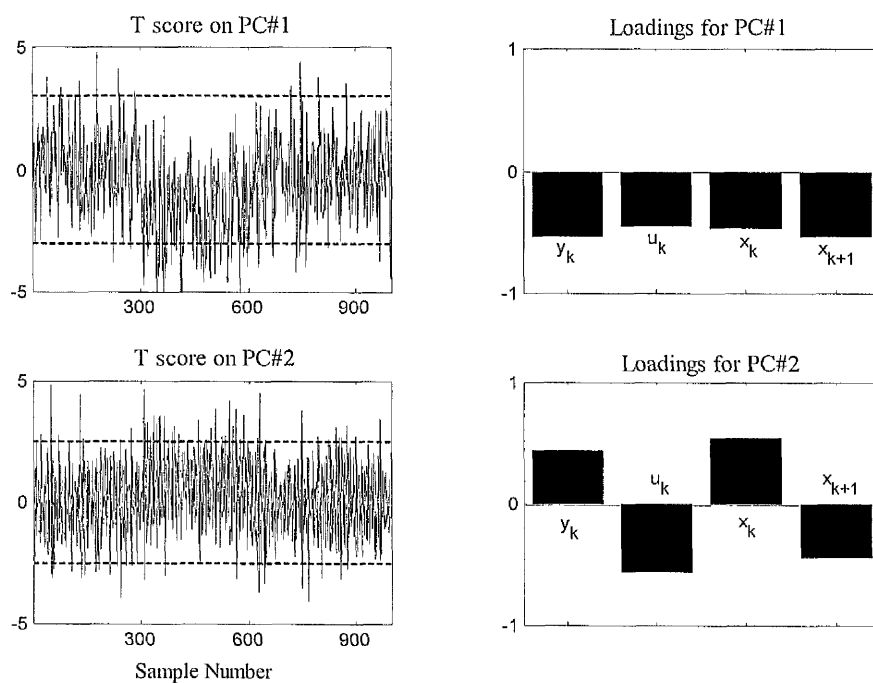


Figure 8.3 T Scores and Loadings for the Subspace Method model.

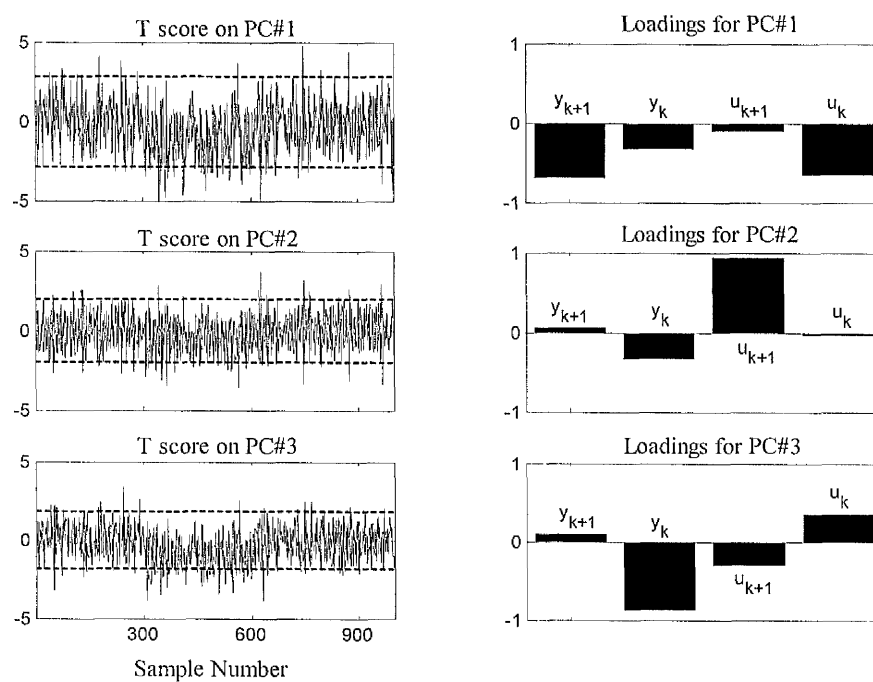


Figure 8.4 T Scores and Loadings for DPCA model.

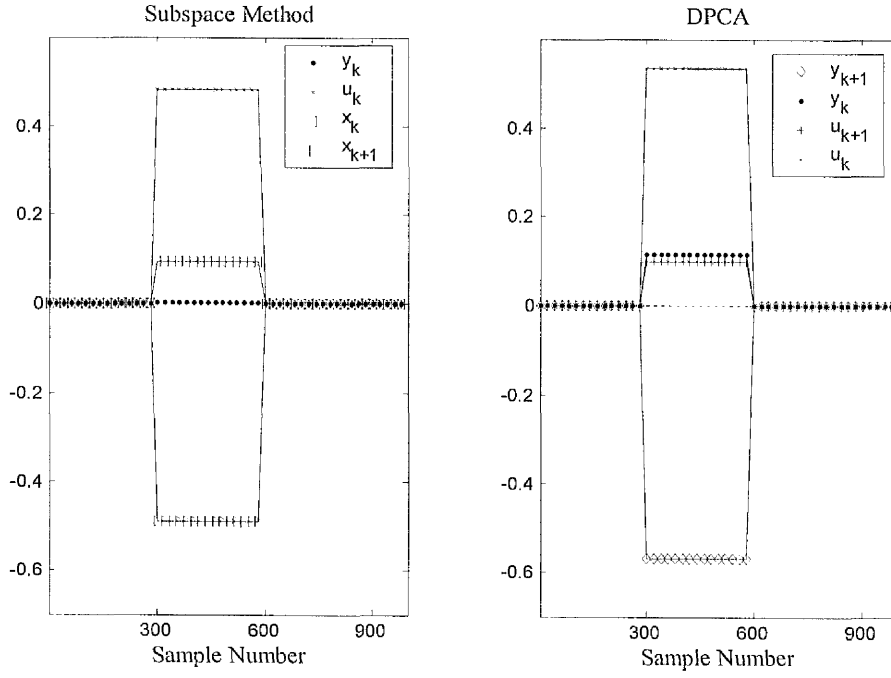


Figure 8.5 Comparison of contributions to Process Residual Q , for Subspace Method (left) and DPCA (right).

$$\mathbb{C}_{SUBSPACE} = \begin{pmatrix} y_k & u_k & x_k & x_{k+1} \end{pmatrix} \begin{pmatrix} 0.5197 & 0.0036 & -0.4906 & -0.0946 \\ 0.0036 & 0.4770 & 0.0946 & -0.4904 \\ -0.4906 & 0.0946 & 0.4831 & -0.0112 \\ -0.0946 & -0.4904 & -0.0112 & 0.5201 \end{pmatrix}$$

$$SUBSPACE \mathbb{C}_{y_k} \cong 0.52y_k + 0.00u_k - 0.49x_k - 0.09x_{k+1}$$

$$SUBSPACE \mathbb{C}_{u_k} \cong 0.00y_k + 0.48u_k + 0.09x_k - 0.49x_{k+1}$$

$$SUBSPACE \mathbb{C}_{x_k} \cong -0.49y_k + 0.09u_k + 0.48x_k - 0.01x_{k+1}$$

$$SUBSPACE \mathbb{C}_{x_{k+1}} \cong -0.09y_k - 0.49u_k - 0.01x_k + 0.52x_{k+1}$$

$$\mathbb{C}_{DPCA} = \begin{pmatrix} y_{k+1} & y_k & u_{k+1} & u_k \end{pmatrix} \begin{pmatrix} 0.5218 & -0.1043 & -0.0890 & -0.4803 \\ -0.1043 & 0.0209 & 0.0178 & 0.0961 \\ -0.0890 & 0.0178 & 0.0152 & 0.0819 \\ -0.4803 & 0.0961 & 0.0819 & 0.4422 \end{pmatrix}$$

$$DPCA \mathbb{C}_{y_{k+1}} \cong 0.52y_{k+1} - 0.10y_k - 0.09u_{k+1} - 0.48u_k$$

$$DPCA \mathbb{C}_{y_k} \cong -0.10y_{k+1} + 0.02y_k + 0.02u_{k+1} + 0.10u_k$$

$$DPCA \mathbb{C}_{u_{k+1}} \cong -0.09y_{k+1} + 0.02y_k + 0.02u_{k+1} + 0.08u_k$$

$$DPCA \mathbb{C}_{u_k} \cong -0.48y_{k+1} + 0.10y_k + 0.08u_{k+1} + 0.44u_k$$

Figure 8.6 Contributions coefficients for the Residual Statistic Q , for Subspace Method and DPCA.

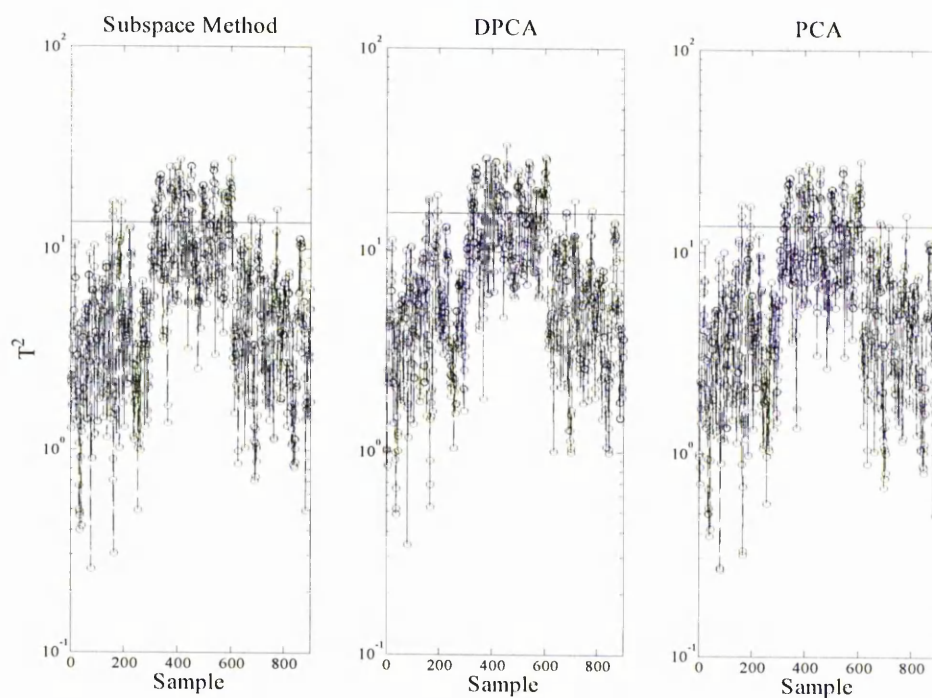


Figure 8.7 Hotelling's T^2 statistic for step increase $w = 1.0$. Autocorrelated process data set.

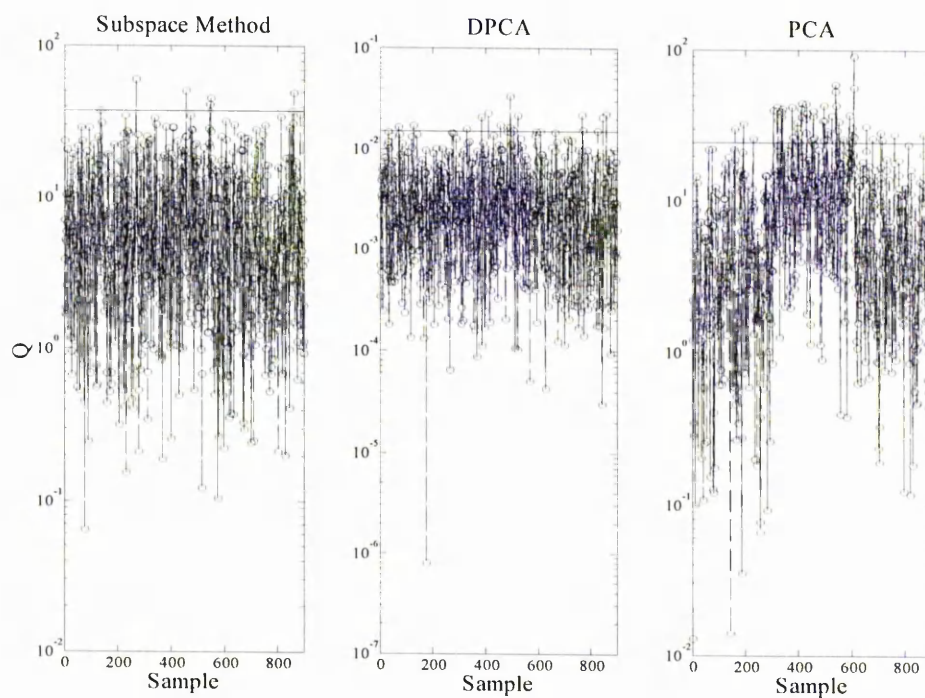


Figure 8.8 Q statistic for step increase $w = 1.0$. Autocorrelated process data set.

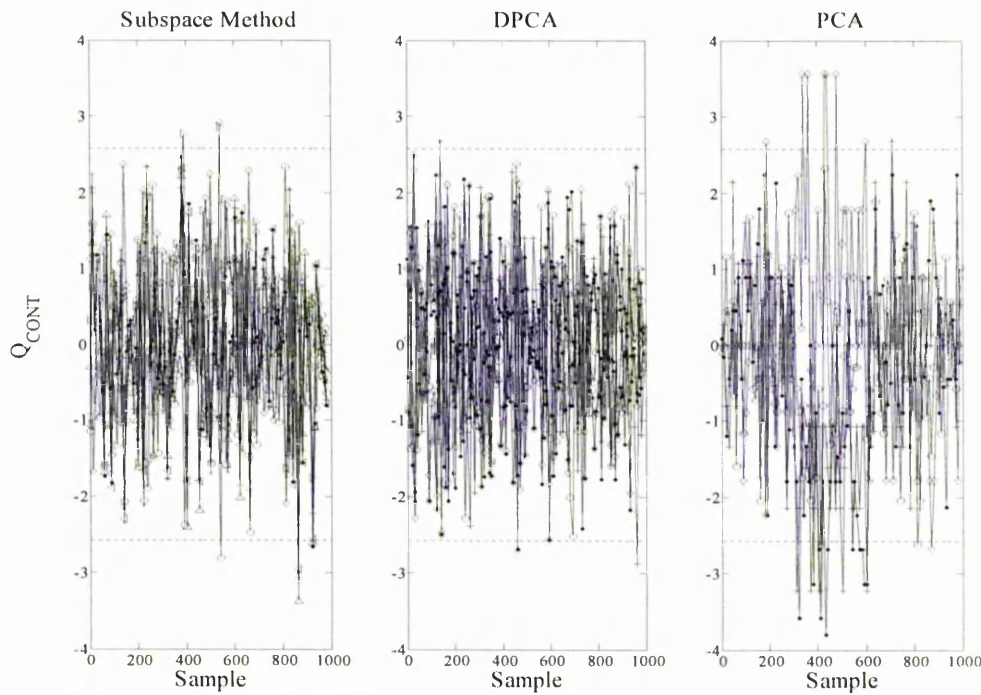


Figure 8.9 Contributions to the Q statistic for the autocorrelated process data set for the process variables $y_1(\diamond)$, $y_2(\bullet)$, $u_1(+)$, and $u_2(o)$, and Subspace Method states ($\triangleleft, \triangle, \mathbf{x}, \triangleright$).

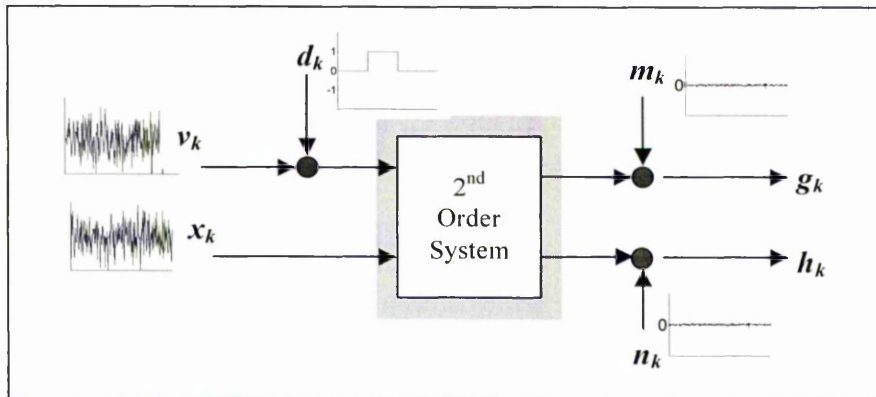


Figure 8.10 The 2nd order system has random inputs v_k and x_k . The system outputs are g_k and h_k . g_k is appended with a normally distributed random sequence with 0.2% variance g_k . h_k is appended with a normally distributed random sequence with 0.05% variance h_k . The unmeasured disturbance d_k is a step input to v_k from sample 300 to sample 600.

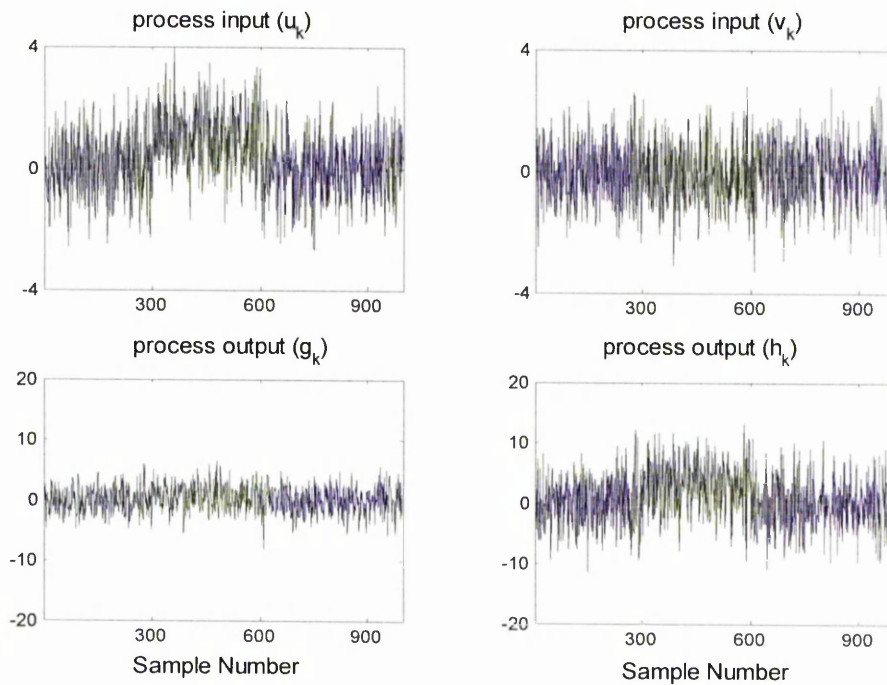


Figure 8.11 The measurement data for the system. A fault (sensor bias in u_k) arrives in the system at sample point 300.

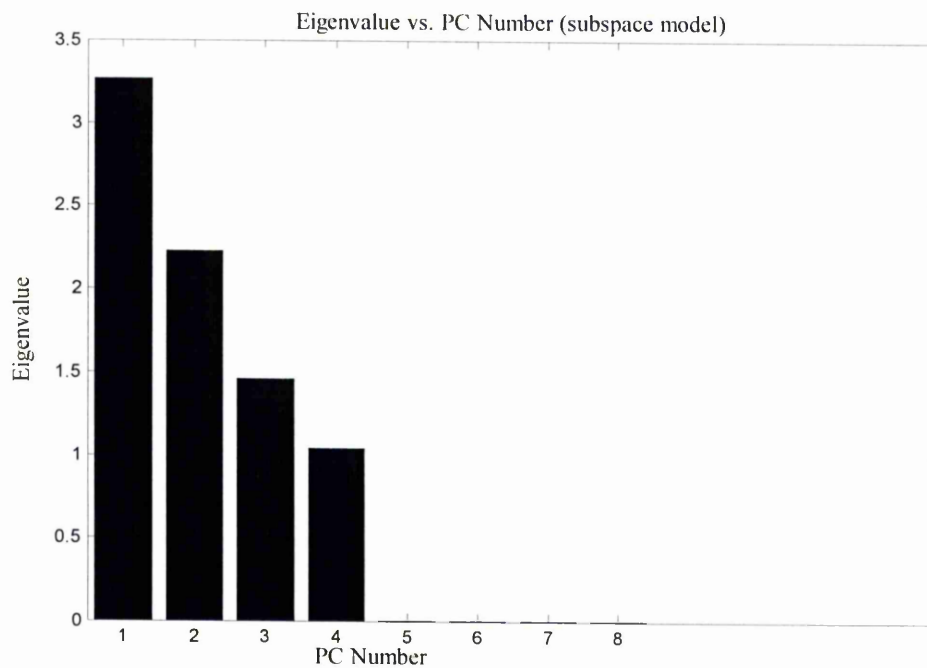


Figure 8.12 The PCA analysis applied to $Z_{SUBSPACE}$ reveals 4 principal directions.

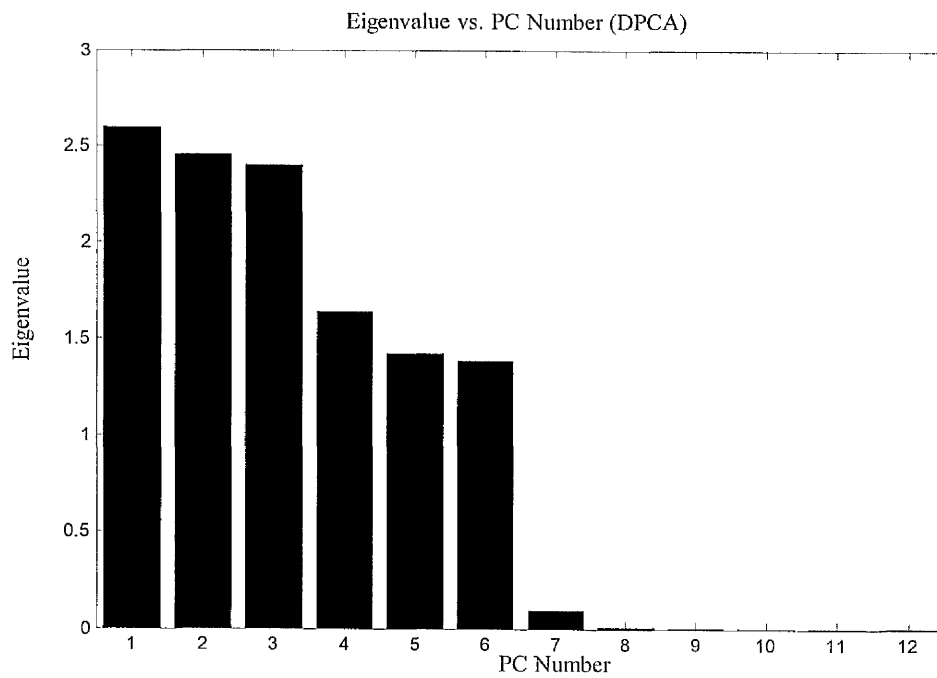


Figure 8.13 PCA analysis applied to Z_{DPCA} reveals 6 principal directions.

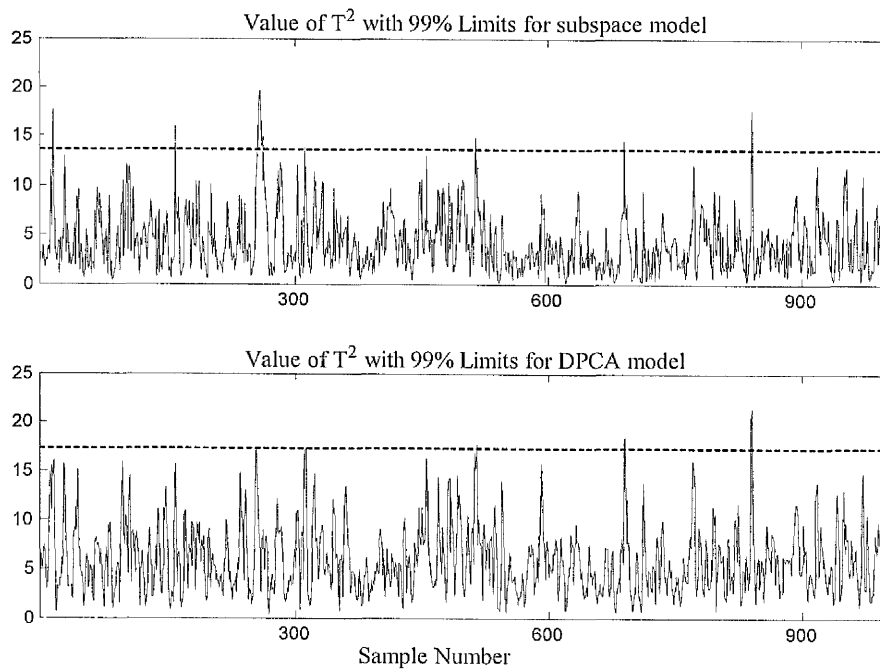


Figure 8.14 The T^2 statistic remains within the 99% limits for both models.

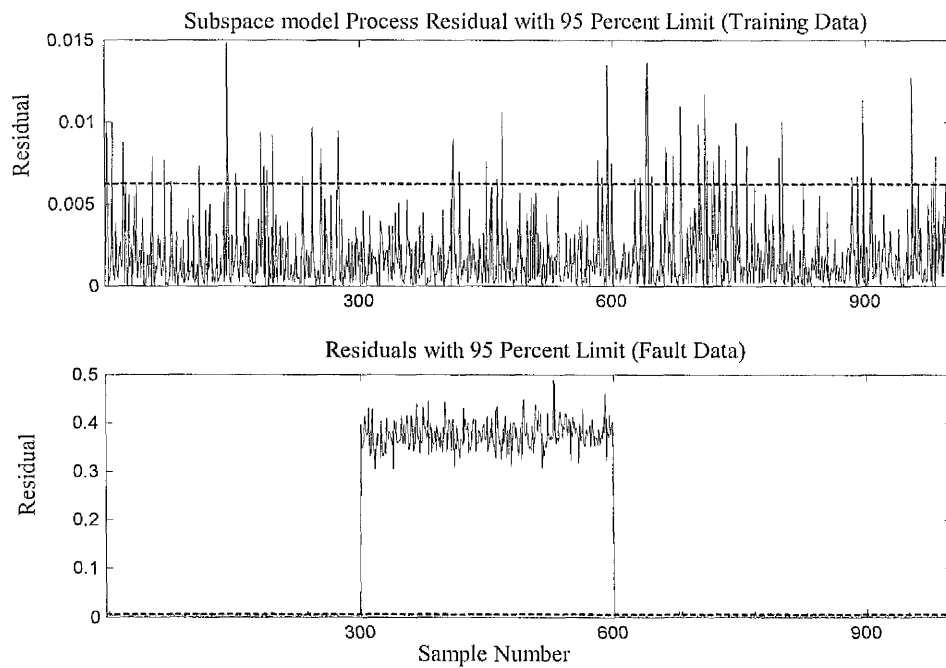


Figure 8.15 Process residual statistic for the subspace method.

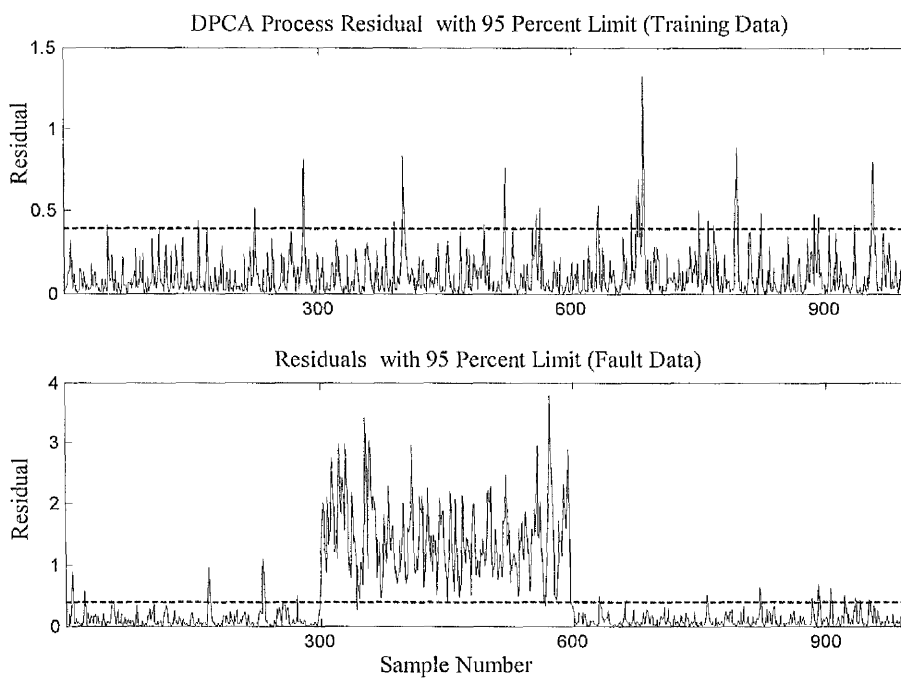


Figure 8.16 Process residual statistic for DPCA model.

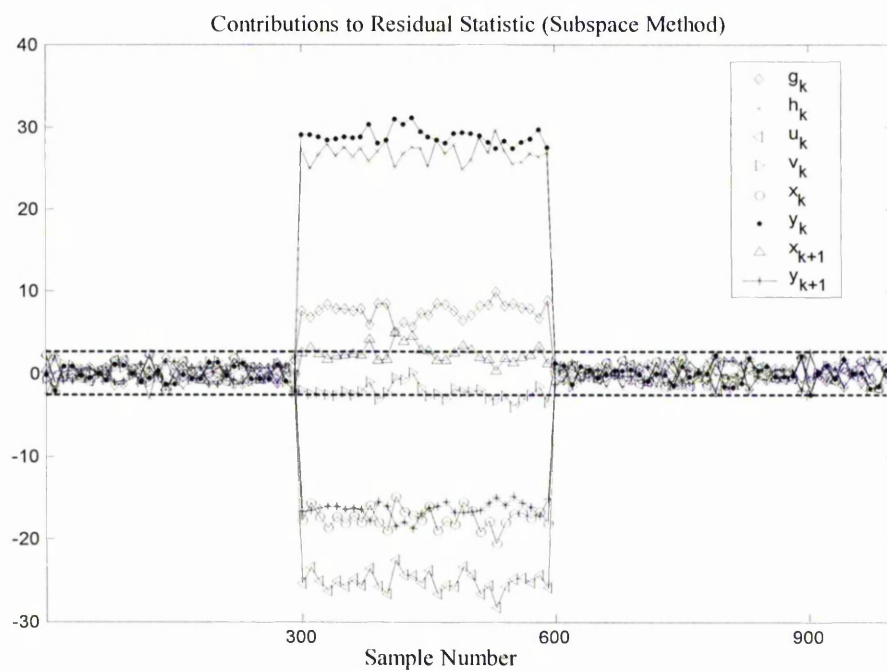


Figure 8.17 Contribution chart for subspace model residuals.

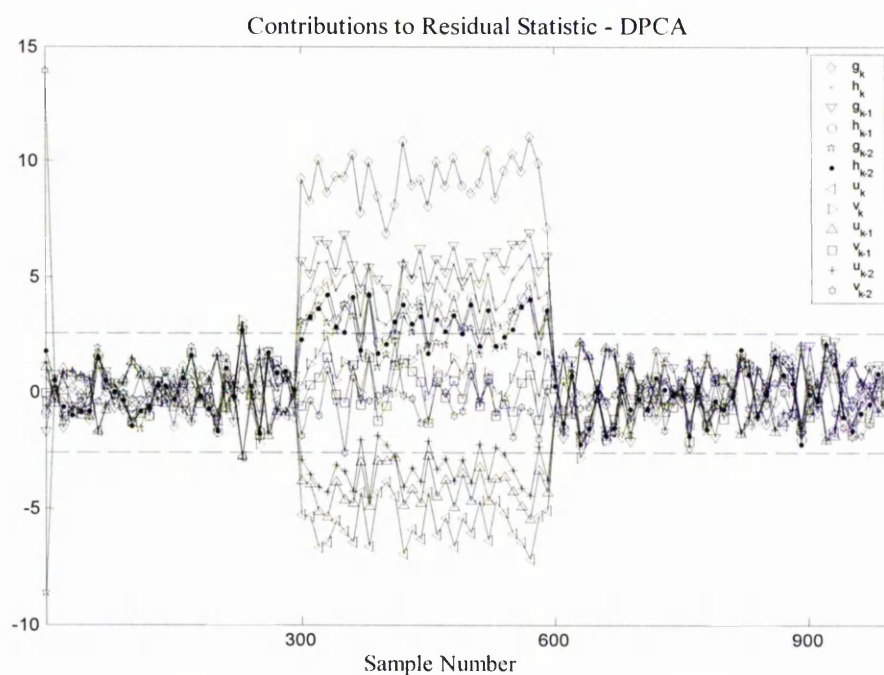


Figure 8.18 Contribution chart for DPCA model residuals.

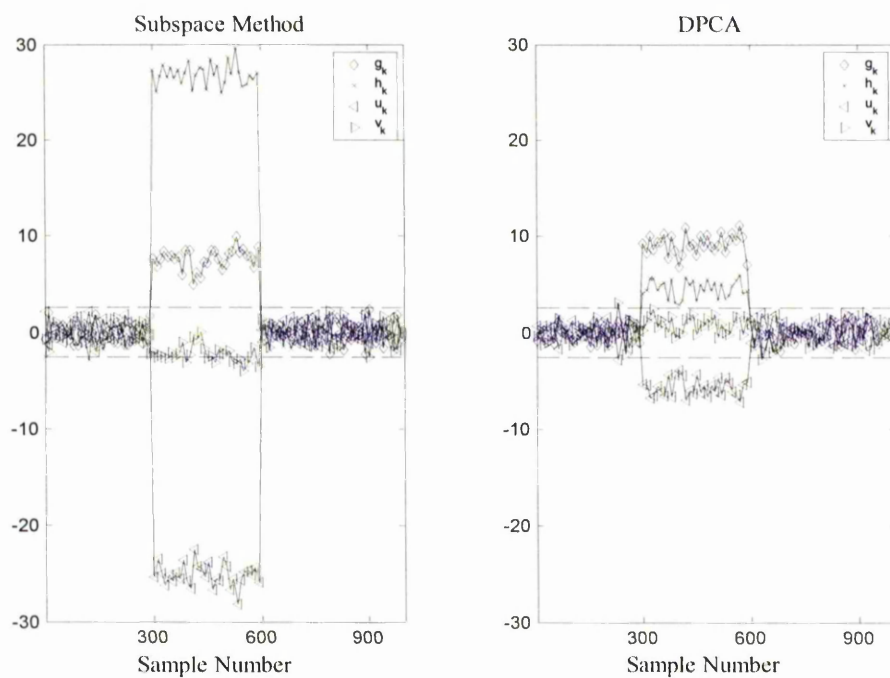


Figure 8.19 Direct comparison of the contributions of the measured variables for the Subspace Method (left) and DPCA (right).

Chapter 9

Conclusions and further work

9.1 Conclusions

The main achievement of this project has been the adaptation and application of subspace system identification to the multivariate statistical process control problem.

The main contribution comes in Part II of the dissertation. A novel subspace method for condition monitoring of industrial processes is developed. The method has been cast into a multivariate statistical process control framework, by defining Hotelling's T^2 and Q statistics. It is suggested that the subspace system identification condition monitor will provide a valuable addition to the monitorMV process condition monitoring software package, by extending its dynamic modelling capability. There are several possible advantages over the current technology available in monitorMV:

- (1) More robust modelling of processes, where dynamic transients lead to excessive false alarms when monitoring with a PCA model.
- (2) A more flexible model that deals with shifts in operating point better than PCA.

- (3) An alternative model structure and contributions analysis, to the dynamic modelling method currently used in monitorMV, i.e. dynamic PCA.

The advantages over PCA, (1) and (2) above, come directly from employing a dynamic model structure, as opposed to a static model structure. This may require further investigation, as to the *significance* of the advantage obtained, ideally using data from industrial applications.

Linear relationships have been drawn between the subspace method, PCA and DPCA. The main points are

- (1) The subspace method corresponds to an error-in-variables approach to the subspace system identification problem, where TLS is used to calculate a state space model from the state sequences. The key step has been the use of scaled states in the calculation of the TLS solution. The scaled states do not affect the input-output behaviour of the model, however, the scaling is important for the ensuing PCA analysis that is used to calculate the principal directions of the subspace monitor.
- (2) The connection between the subspace method and the PCA method used in monitorMV is in the data matrices used by each. The subspace method uses an “augmented” data matrix, comprised of the process measurements, and the associated state sequences, i.e. it is a PCA analysis with the state sequences appended to the matrix of process measurements.
- (3) A linear relationship between the subspace method and DPCA is readily apparent when it is considered that the subspace method applies PCA to a data structure that conforms to a state space model of the system, in exactly the same way that DPCA conforms to a PCA analysis of an ARX model structure. The advantages that the subspace method enjoys over DPCA all come from the use of the state space model structure, that provides a more parsimonious description of the system.

The subspace method has been compared with DPCA and PCA using a simple open-loop simulation, which was used to demonstrate an advantage gained by incorporating dynamics into the condition monitoring model. The possible advantage enjoyed by

dynamic models comes from their ability to accommodate auto-correlated and dynamic data correctly in their model structures. This was demonstrated in Chapter 8, where the subspace method was found to be more robust than PCA, when an unmeasured disturbance affected the system.

Part I of the dissertation develops the theoretical background for the subspace methods that are applied to the process condition monitoring problem in part II. In part I, subspace theory has been treated from a system identification point of view. It is demonstrated, using a simulation of a complex industrial process simulation of an FCCU, that subspace methods can be used for modelling large-scale industrial processes. Several aspects of how to obtain good models have been discussed.

The main points are

- (1) The effect of the number of block rows (r) that is used in the algorithms has a significant effect on model accuracy. There is, as yet, (to the best of this author's knowledge), no guaranteed method for optimising this user choice. However, the choice of the number of block rows can be automated, for example, by calculating an error statistic based on cross-validation, where one or several segments of data can be retained for model validation.
- (2) Two methods for determining the order of dynamics to be possessed by the model have been considered. In the system identification part of the dissertation, AIC was used, however, for the N4SID approach that is used in the subspace method in part II, the starting point for estimating the system order is the relative magnitude of the singular values. Following this, cross-validation can be applied to evaluate the results. Clearly, prior knowledge of the system under consideration, or knowledge of the dynamics involved, can be very helpful in this endeavour. In general, it has been found that the use of AIC to find the optimum model structure is likely to lead to higher order models than are actually required.
- (3) Advantages of using a fully parameterised state space models over ARX and FIR model structures have been highlighted. The subspace models use powerful latent directions (the orthogonal state sequences) to describe the system dynamics. In contrast, the ARX model structure uses the process measurements

to describe the system dynamics, which for large-scale MIMO processes, requires a large number of time-shifted process variables to be included in the model structure. The concise nature of the fully parameterised state space model structure provides the subspace methods employed in part II of the dissertation, with a practical advantage over current monitorMV technology, namely DPCA.

A case study involving a 3DOF mass-spring-damper system was presented to demonstrate some of the user critical issues associated with the application of the algorithms. The study was used as a vehicle for coming to grips with aspects of choosing the model order (n), and the effect of model order choice on model stability. The very important issue of the effect of the number of block rows (r) used in the input-output Hankel matrices on prediction accuracy was also investigated. It has been concluded that there is no straightforward answer regarding the number of block rows to use in each of the algorithms. Success was achieved by iterating through a range of values (subject to $r > n$), and then using cross-validation to assess the results.

The point is made in Overschee (1996) [1] that the variability of the results, between the three most often cited subspace algorithms, is due to each of the subspace algorithms calculating a different orthogonal state basis to describe the system. Subject to certain mild conditions, the state basis used by each of the algorithms is equivalent up to within a similarity transformation, however in the face of noise and unmeasured disturbances, this may have considerable influence on prediction accuracy. The comparisons made in this study of the MSIT implementation (M_3) and the subid.m (M_1) algorithm presented in [1], using the several simulations and industrial data, yielded conflicting results, suggesting that it may be prudent to try different algorithms on each data set and assess the results.

Unmeasured disturbances, process and measurement noise all exert a major influence on the model prediction error for each of the linear methods considered. However, it is concluded that subspace methods offer the following attractive features, (not all of which are possessed by the FIR and ARX time-series modelling methods):

- (1) Subspace methods use robust numerical techniques from linear algebra, that are well known and well understood.

- (2) Use of the singular value decomposition gives the subspace methods the ability to deliver low order, fully parameterised, state space model structures, in which the (orthogonal) state sequences are able to capture the important dynamics. This leads to a model structure with fewer parameters than can be obtained using time-series models.
- (3) Further optimisation is possible using iterative search routines and there is also the added flexibility of being able to obtain a range of analytical solutions on the basis of using different numbers of block rows (r).
- (4) The balanced realisation provided by subspace identification makes the state space model order reduction problem both intuitive and easy to execute.
- (5) Subspace system identification methods identify models in state space, which are naturally suited to the multivariable control problem, for which there exists bountiful control theory in support.

A challenge, in the application of subspace methods for system identification, is the trial and error approach that is required, due to the effect of the number of block rows (r) on the results. It has been observed that a change in r can alter the results significantly, for example it may mean the difference between a stable and unstable model being identified. When applying the subspace method for condition monitoring, the choice of (r) has a noticeable effect on the contributions analysis, meaning that a generous amount of time should be dedicated to it. However, for the monitorMV software package, an implementation is required where the user choices (n, r) are automated, i.e. where default choices are available.

9.2 Further Work

It is suggested that the subspace system identification condition monitor will provide a valuable addition to the monitorMV process condition monitoring software package by extending its dynamic modelling capability. A possible advantage over the current technology available in monitorMV is that the subspace method offers more robust modelling of processes where dynamic transients lead to excessive false alarms, than a

PCA model. This claim has been made on the basis of results obtained from a simple simulation, and also based on information in the paper by Ku, Storer and Georgakis (1995) [91]. Alternative scenarios need to be investigated using industrial data. A useful test case will be a process where false alarms ring on a regular basis.

The statistical procedures for processing the information generated by condition monitors are well established. However, in terms of the Subspace Method model, there are several areas open for research, regarding improvements to the Subspace Method algorithm and the methodology associated with it. The relationship between the number of block rows and model accuracy needs to be further investigated, in particular in the context of the error-in-variables approach to subspace system identification. The numerical procedure employed in the algorithms might be improved, leading to less sensitivity in the choice of r . The relationship between the order of the state space model, n , and the number of block rows, r , could also warrant further investigation.

The relationship between the order of the state space model, the number of process variables, and the maximum number of principal components required in the monitor has been described. However aspects of dimension reduction need further understanding. A trial and error approach was applied in the studies described here, however a more analytical approach would be helpful.

A continuous stirred tank reactor process simulation was presented in Chapter 7, to demonstrate three dynamic modelling approaches for MSPC. An ARX model structure was used in conjunction with PCA and PLS. These were compared and contrasted with the subspace method, where the emphasis of the comparison was placed on both the detection and the isolation of faults. It was found that each method provided a different footprint of the faults, most likely due to the different way in which the latent variables are calculated by each method. Since each of the methods provides a valid linear representation of the data, it is reasonable to believe that each of the dynamic methods has the potential to deliver important information regarding the process. This suggests that each of the methods could be run in parallel, thus providing a collection of footprints that can be used to build up a layered fault signature.

A second subspace approach might be developed where an alternative approach to generating the latent variables is used. This will involve applying PLS, rather than TLS,

to calculate the latent variables with which to build the condition monitor. Note that the subspace method produces a matrix containing the state sequences where the cause/effect structure is as follows

$$\begin{aligned}\mathbf{X}_k &\rightarrow \mathbf{X}_{k+1} \\ \mathbf{U}_k &\rightarrow \mathbf{Y}_k\end{aligned}$$

The subspace method calculates a TLS solution, i.e. SVD is applied to a matrix composed of the data sequences $(\bar{\mathbf{X}}_k \quad \bar{\mathbf{X}}_{k+1} \quad \mathbf{U}_k \quad \mathbf{Y}_k)$. However a PLS approach can also be applied, where the data is divided into two blocks, i.e.

$$PLS - X - Block \rightarrow [\bar{\mathbf{X}}_k \quad \mathbf{U}_k]$$

$$PLS - Y - Block \rightarrow [\bar{\mathbf{X}}_{k+1} \quad \mathbf{Y}_k]$$

Then proceed using PLS and develop the associated statistics for the model, as described in section 7.2.3.

9.3 Publications

The following conference papers were presented in support of this work:

Treasure R.J., U. Kruger, & J.E. Cooper,

Identification of Continuous Industrial Processes using Subspace System Identification Methods. Condition Monitoring and Diagnostic Engineering Management 2001, 1999: p. 615-623.

Treasure, R.J., U. Kruger, & J.E. Cooper,

System Identification Methods for Industrial Plant. Proceedings of the IASTED International Conference on Modelling, Identification and Control, Innsbruck, Austria. Feb 18-21, 2002. 1: p. 415-419.

The following journal paper has been accepted for publication in the Journal of Process Control:

Treasure R.J., U. Kruger, & J.E. Cooper,

Dynamic Multivariate Statistical Process Control using Subspace Identification.

Appendix

Details of the mass-spring-damper simulation used in Chapter 4 are as follows:

The state space formulation of the system is:

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} + \mathbf{v}$$

$$\mathbf{y} = \mathbf{Cx} + \mathbf{Du} + \mathbf{w}$$

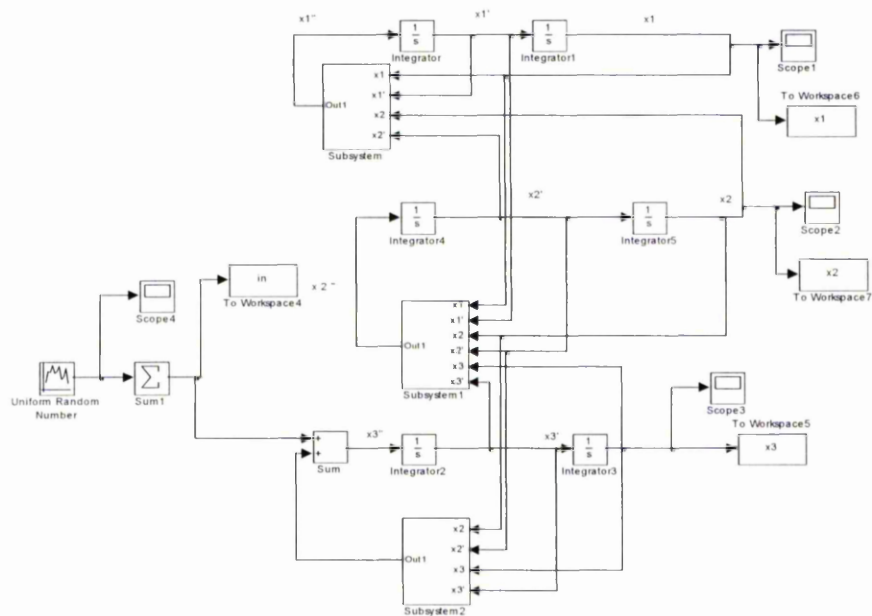
$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -9500 & 4500 & 0 & -25 & 10 & 0 \\ 4500 & -10500 & 6000 & 10 & -15 & 5 \\ 0 & 6000 & -6000 & 0 & 5 & -5 \end{pmatrix};$$

$$\mathbf{B} = (0 \ 0 \ 0 \ 0 \ 0 \ 1)^T; \ \mathbf{C} = (0 \ 1 \ 0 \ 0 \ 0 \ 0); \ \mathbf{D} = 0.$$

MODE 1		MODE 2		MODE 3	
Frequency	Damping	Frequency	Damping	Frequency	Damping
4.96	0.026	14.53	0.070	20.55	0.041

Table A2.1 Shows the modal frequencies (in Hertz) and damping of the system.

Simulation:



Simulation Details:

Solver: Fixed-Step ode5 (Dormand-Prince)

Start Time: 0.0

Stop Time: 40

Input: Uniform Random Number.

Input Minimum: -0.2

Input Maximum: 0.2

Initial seed: 7

Sample Time of Input: 0.5

Sample Time of Output: 0.01

References

1. Van Overschee P and B. De Moor, Subspace Identification for linear systems. 1996: Kluwer Academic Publishers.
2. Willems, J.C., From Time Series to Linear Systems. Part I. Finite Dimensional Linear Time Invariant Systems. Automatica, 1986. 22(5): p. 561-580.
3. Chou, C.T., Geometry of linear systems and identification. 1994, Cambridge University.
4. Qin, S.J. and T.A. Badgwell, A Survey of Industrial Model Predictive Control Technology (Draft). Internal Report, May 2001.
5. Soderstrom, T. and P. Stoica, System Identification. 1989: Prentice Hall International (UK).
6. Ljung, L., System Identification. 1999, New Jersey: Prentice Hall PTR.
7. Akaike, H., Canonical Correlation analysis of time series and the use of an information criterion. 1977.
8. Favoreel, W., B. De Moor, and P. Van Overschee, Subspace state space system identification for industrial processes. Journal of Process Control, 2000. 10(2): p. 149-155.
9. Cooper, J.E., Modal Parameter Identification Using Time Domain Methods, in Queen Mary College. 1988, University of London.
10. Box, G.E.P. and G.M. Jenkins, Time series analysis : forecasting and control. 1970, San Francisco: Holden-Day.
11. Strang, G., Introduction to linear algebra. 1993: Wellesley-Cambridge Press.
12. Kalman, R.E., On the general theory of control systems. Proceedings of the 1st International Congress on Automatic Control, Moscow. 1960. 481 - 492.
13. Kalman, R.E., Mathematical description of linear dynamical systems. SIAM Journal on Control, 1963.
14. Kailath, T., A. Sayed, and B. Hassibi, Linear Estimation. 2000: Prentice Hall, NJ.

15. Moore, B.C., Principal component analysis in linear systems - controllability, observability and model reduction. *IEEE Transactions on Automatic Control*, 1981. AC-26(1): p. 17-32.
16. Ramirez, W.F. and J.M. Maciejowski, Balanced realization for state-space identification and optimal output regulation. *AIChE Journal*, 1995. 41, 5: p. 1217-1228.
17. Viberg, M., Subspace-based methods for the identification of linear time-invariant systems. *Automatica*, 1995. 31, 12: p. 1835-1851.
18. Larimore, W.E., Canonical variate analysis in identification, filtering, and adaptive control, in *Proceedings of the IEEE Conference on Decision and Control*. 1990, Publ by IEEE. p. 596-604.
19. Larimore, W.E., System identification, reduced order filtering and modeling via canonical variate analysis, in *Proceedings of the American Control Conference*. 1983, IEEE. p. 445-451.
20. Larimore, W.E., Canonical Variate Analysis in Control and Signal Processing, in *Statistical Methods in Control and Signal Processing*. 1997. p. 83-119.
21. Verhaegen, M., Subspace Model Identification, Part 3, Analysis of the ordinary output-error state-space model identification algorithm. *Int. J. Control*, 1993. 58: p. 555-586.
22. Verhaegen, M. and D. Dewilde, Subspace model identification. Part Two: Analysis of the elementary output error state-space model identification algorithm. *Int. J. Control*, 1992. 56(5): p. 1187-1210.
23. Verhaegen, M., A novel non-iterative mimo state space model identification technique. *IFAC system identification conf. Hungary*, 1991.
24. Verhaegen, M. and D. Dewilde, Subspace model identification. Part One: The output-error state-space model identification class of algorithms. *Int. J. Control*, 1992. 56(5): p. 1187-1210.
25. Van Overschee, P. and B. De Moor, N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 1994. 30(1): p. 75-93.
26. Ho, B.L. and R.E. Kalman, Effective construction of linear state-variable models from input/output functions. *Regelungstechnik*, 1966: p. 545-548.

27. Zeiger, H.P. and A.J. McEwen, Approximate Linear Realizations of Given Dimension via Ho's Algorithm. *IEEE Transactions on Automatic Control*, 1974. AC-19(2): p. 153.
28. Kung, S.Y., A New Identification and Model Reduction Algorithm Via Singular Value Decompositions. *Proceedings of the 12th Asimolar Conference on Circuits, Systems and Computers*, 1978: p. 705-714.
29. Akaike, H., Markovian representation of stochastic processes by canlnical variables. *SIAM J Control*, 1975. 13(1): p. 162-173.
30. Verhaegen, M., Identification of the deterministic part of MIMO state space models given in innovations form from input-output data. *Automatica*, 1994. 30(1): p. 61-74.
31. Ljung, L. and T. McKelvey, Least squares interpretation of sub-space methods for system identification. *Proceedings of the IEEE Conference on Decision and Control*, 1996. 1.
32. Van Overschee, P. and B. De Moor, Unifying theorem for three subspace system identification algorithms. *Proceedings of the American Control Conference*, 1994. 2: p. 1645-1649.
33. De Moor, B., M. Moonen, L. Vandenberghe and J. Vandewalle, Geometrical approach for the identification of state space models with singular value decomposition, in *Proceedings - ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing*. 1988, IEEE. p. 2244-2247.
34. Bauer, D., M. Deistler, and W. Scherrer, User choices in subspace algorithms. *Proceedings of the IEEE Conference on Decision and Control*, 1998. 1: p. 731-736.
35. Bauer, D., On data preprocessing for subspace methods, in *Proceedings of the IEEE Conference on Decision and Control*. 2000. p. 2403-2408.
36. Catarina, J.M., P. Delgado, P. Lopes dos Santos, and J.L. Martins de Carvalho, Numerical algorithm for recursive subspace identification. *Proceedings of the IEEE Conference on Decision and Control*, 1998. 2: p. 1848-1849.
37. Moonen, M. and J. Vandewalle, QSVD approach to on- and off-line state-space identification. *International Journal of Control*, 1990. 51(5): p. 1133-1146.

38. Chou, C.T. and M. Verhaegen, Subspace algorithms for the identification of multivariable dynamic errors-in-variables models. *Automatica*, 1997. 33(10): p. 1857-1869.
39. Gustafsson, T., Subspace identification using instrumental variable techniques. *Automatica*, 2001. 37(12): p. 2005-2010.
40. Moonen, M. and J. Ramos, Subspace algorithm for balanced state space system identification. *IEEE Transactions on Automatic Control*, 1993. 38(11): p. 1727-1729.
41. Ljung, L., Interpretation of subspace methods:consistency analysis. 11th IFAC Symposium on SYSID, Japan, 1997.
42. Ljung, L. and T. McKelvey, Subspace identification from closed loop data. *Signal Processing*, 1996. 52, 2: p. 209-215.
43. Verhaegen, M., Application of a subspace model identification technique to identify LTI systems operating in closed loop. *Automatica*, 1993. 29(4): p. 1027-1040.
44. Van Overschee, P. and B. De Moor, Closed loop subspace system identification, in *Proceedings of the IEEE Conference on Decision and Control*. 1997, IEEE. p. 1848-1853.
45. Van Overschee, P. and B. De Moor, Subspace algorithms for the stochastic identification problem. *Automatica*, 1993. 29(3): p. 649-660.
46. Akaike, H., Stochastic theory of minimum realization. *IEEE Trans Autom Control*, 1974. AC-19(6): p. 667-674.
47. Van Overschee, P. and B. De Moor, Choice of state-space basis in combined deterministic-stochastic subspace identification. *Automatica*, 1995. 31, 12: p. 1877-1883.
48. Enns, D., Model Reduction for Control System Design. Ph.D. Dissertation.Stanford University., 1984.
49. Maciejowski, J.M., Guaranteed stability with subspace methods. *Systems & Control Letters*, 1995. 26(2): p. 153-156.
50. Ottersten, V., A subspace based instrumental variable method for state-space sys ID. *Proceedings of SYSID'94*, 1994. 12: p. 139-144.

51. Verhaegen, M. and E. Deprettere, A fast, recursive MIMO state space model identification algorithm. Proceedings of the IEEE Conference on Decision and Control, 1991. 2: p. 1349-1354.
52. Juang, J., Applied System Identification. 1994: Prentice Hall, NJ.
53. Juang, J., J.E. Cooper, and J.R. Wright, An Eigensystem Realisation Algorithm using Data Correlations (ERA/DC) for modal parameter identification. Control theory and advanced technology, 1988. 4(1): p. 1-14.
54. Hunter, N.F., Comparing CVA and ERA in transfer function measurements for lithography applications, in Proceedings of the American Control Conference. 1999, IEEE. p. 1171-1175.
55. Kailath, T., Linear System Theory. 1980: Englewood Cliffs, N.J: Prentice Hall.
56. Simoglou, A., P. Argyropoulos, E.B. Martin, K. Scott, A.J. Morris and W.M. Taama, Dynamic modelling of the voltage response of direct methanol fuel cells and stacks Part I: Model development and validation. Chemical Engineering Science, 2001. 56(23): p. 6761-6772.
57. Verhaegen, M., Identification of the deterministic and stochastic part of mimo state space models under the presence of process and measurement noise. Automatica Vol 30, No 1, pp 61-74, 1994. 30(1): p. 61-74.
58. Van Overschee, P. and B. De Moor, Subspace algorithms for the stochastic identification problem, in Proceedings of the IEEE Conference on Decision and Control. 1991, Publ by IEEE. p. 1321-1326.
59. Van Overschee, P., B. De Moor, W. Dehandschutter and J. Swevers, Subspace algorithm for the identification of discrete time frequency domain power spectra. Automatica, 1997. 33(12): p. 2147-2157.
60. De Moor, B., Mathematical concepts and techniques for modeling of static and dynamic systems, in Department of Electrical Engineering,. 1988, Katholieke Universiteit: Leuven, Belgium.
61. Abdelghani, M., C.T. Chou, and M. Verhaegen, Using subspace methods in the identification and modal analysis of structures, in Proceedings of the International Modal Analysis Conference - IMAC. 1997, SEM. p. 1392-1398.
62. Brewer, J.W., Kronecker Products and Matrix Calculus in System Theory. IEEE Transactions on Circuits and Systems, 1978. 25(9): p. 772-780.

63. Favoreel, W., B. De Moor, P. Van Overschee, and M. Gevers, Model-free subspace-based LQG-design. Proceedings of the American Control Conference. San Diego. 1999. IEEE. p. 3372-3376.
64. Marjonovic, O., Constrained LQR. PhD Thesis. University of Manchester, 2002.
65. Schaper, C.D., W.E. Larimore, D.E. Seborg, and D.A. Mellichamp, Identification of chemical processes using canonical variate analysis, in Proceedings of the IEEE Conference on Decision and Control. 1990, Publ by IEEE. p. 605-610.
66. Ljung, L., System Identification Toolbox - Matlab User's Guide. Version 5. 2000.
67. Allen, D.M., The relationship between variable selection and data augmentation and a method of prediction. *Technometrics*, 1971. 13: p. 469-475.
68. Candy, J.V., T.E. Bullock, and M.E. Warren, Invariant description of the stochastic realisation. *Automatica*, 1979. 15: p. 493-495.
69. Draper, N.R. and H. Smith, *Applied regression Analysis*. 2nd ed. 1981, New York: John Wiley & Sons.
70. Ansari, R.M. and M.O. Tadé, Constrained nonlinear multivariable control of a fluid catalytic cracking process. *Journal of Process Control*, 2000. 10: p. 539-555.
71. McFarlane, R.C., R.C. Reineman, J.F. Bartee and C. Georgakis, Dynamic simulator for a model IV fluid catalytic cracking unit. *Computers & Chemical Engineering*, 1993. 17(3): p. 275-300.
72. Prett, D.M. and R.D. Gillette, Optimisation and constrained multivariate control of a catalytic cracking unit. *Proc. JACC*, 1980.
73. Monge, J.J. and C. Georgakis, Multivariable control of a catalytic cracking process. *Chemical Engineering Community*, 1987. 61: p. 197-225.
74. Camacho, E.F. and C. Bordons, *Model Predictive Control*. 1998, London: Springer.
75. Treasure, R.J., Identification of Continuous Industrial Processes using Subspace System Identification Methods. *Condition Monitoring and Diagnostic Engineering Management* 2001, 1999: p. 615-623.

76. MacGregor, J.F. and T. Kourti, Statistical process control of multivariate processes. *Control Engineering Practice*, 1995. 3, 3: p. 403-414.
77. Kruger, U., Q. Chen, D.J. Sandoz and R.C. McFarlane, Extended PLS Approach for enhanced condition monitoring of industrial processes. *AIChE Journal*, 2001. 47(9): p. 2076-2091.
78. Lennox, B., Multivariate Statistical Process Control. CTC Internal Report, 2000.
79. Kourti, T., J. Lee, and J.F. MacGregor, Experiences with industrial applications of projection methods for multivariate statistical process control. *Computers & Chemical Engineering*, 1996. 20, Suppl pt A: p. S745-S750.
80. MacGregor, J.F., Marlin, T.E., Kresta, J., and Skagerberg, B., Multivariate statistical methods in process analysis and control. *Proceedings of the 4th International Conference on Chemical Process Control*. AIChE Publication 67, 1991: p. 79-99.
81. Martin, E.B. and A.J. Morris, An overview of multivariate statistical control in continuous and batch performance monitoring. *Trans. Inst. MC*, 1996. 18(1): p. 51-60.
82. Kourti, T. and J.F. MacGregor, Process analysis, monitoring and diagnosis using multivariate projection methods. *Chemometrics and Intelligent Laboratory Systems*, 1995. 28: p. 3-21.
83. Mardia, K.V., J.T. Kent, and J.M. Bibby, *Multivariate Analysis*. 1979, London: Academic Press.
84. Wise, B.M. and N.B. Gallagher, The process chemometrics approach to process monitoring and fault detection. *Journal of Process Control*, 1996. 6(6): p. 329-348.
85. Chen, Q., Wynne, R., Goulding, P.R., and Sandoz, D.J., The Application of Principal Component Analysis and Kernel Density Estimation to Enhance Process Monitoring. *Control Engineering Practice*, 2000. 8(5): p. 531-543.
86. Simoglou, A., E.B. Martin, and A.J. Morris, Multivariate statistical process control of an industrial fluidised-bed reactor. *Control Engineering Practice*, 2000. 8: p. 893-909.
87. Pranatyasto, T.N. and S.J. Qin, Sensor validation and process fault diagnosis for FCC units under MPC feedback. *Control Engineering Practice*, 2001. 9: p. 877-888.

88. Gallagher, N.B., Wise, B. M., Butler, S.W., White, D.D., and Barna, G.G., Development and Benchmarking of Multivariate Statistical Process Control Tools for a Semiconductor Etch Process. Proceedings of ADCHEM 97. Banff. Canada., 1997: p. 78-83.
89. Li, W., and Qin, S.J., Recursive PCA for adaptive process monitoring. Journal of Process Control, 2000. 10: p. 471-486.
90. Wang, X., U. Kruger, and B. Lennox, Recursive partial least squares algorithms for monitoring complex industrial processes. Accepted for Publication in Control Engineering Practice.
91. Ku, W., R.H. Storer, and C. Georgakis, Disturbance Detection and Isolation by Dynamic Principal Component Analysis. Chemometrics and Intelligent Laboratory Systems, 1995. 30: p. 179-196.
92. Abdelghani, M., Assessment of subspace fault detection algorithms on a realistic simulator-based example. Shock and Vibration Digest, 2000. 32(1): p. 58.
93. Basseville, M., M. Abdelghani, and A. Benveniste, Subspace-based fault detection algorithms for vibration monitoring. Automatica, 2000. 36(1): p. 101-109.
94. Russel, E.L., L.H. Chiang, and R.D. Braatz, Data-driven techniques for fault detection and diagnosis in chemical processes. Advances in Industrial Control. 2000, London: Springer.
95. Norvilas, A., Negez, A., DeCicco, J., and Cinar, A., Intelligent process monitoring by interfacing knowledge-based systems and multivariate statistical monitoring. Journal of Process Control, 2000. 10(4): p. 341-350.
96. Simoglou, A., E.B. Martin, and A.J. Morris, Canonical Correlation Analysis in Process Fault Detection. IFAC Conference SAFEPROCESSES 2000, Budapest, Hungary., 2000: p. 1038-1043.
97. Shi, R. and J.F. MacGregor, Modeling of dynamic systems using latent variable and subspace methods. Journal of Chemometrics, 2000. 14(5-6): p. 423-439.
98. Van Huffel, S. and J. Vandewalle, Algebraic connections between the least squares and total least squares problems. Numerische Mathematik, 1989. 55: p. 431-449.
99. Van Huffel, S. and J. Vandewalle, Algebraic relationships between classical regression and total least squares estimation. Linear algebra and its applications, 1987. 93: p. 148-160.

100. Yoon, S. and J.F. MacGregor, Statistical and causal model-based approaches to fault detection and isolation. *AIChE Journal*, 2000. 46(9): p. 1813-1824.
101. Gertler, J., Fault detection and isolation using parity relations. *Control Engineering Practice*, 1997. 5(5): p. 653-661.
102. Treasure, R.J., U. Kruger, and J.E. Cooper, System Identification Methods for Industrial Plant. *Proceedings of the IASTED International Conference on Modelling, Identification and Control*, Innsbruck, Austria. Feb 18-21, 2002. 1: p. 415-419.
103. Negiz, A. and A. Cinar, Statistical monitoring of multivariable dynamic processes with state-space models. *AIChE Journal*, 1997. 43(8): p. 2002-2020.
104. Larimore, W.E., S. Mahmood, and R.K. Mehra, Adaptive model algorithmic control. 1983, IFAC by Pergamon Press.
105. Li, W. and S. Qin, Consistent dynamic PCA based on errors-in-variable subspace identification. *Journal of Process Control*, 2000. 11(6): p. 661-678.
106. Wang, J. and S.J. Qin, A new subspace identification approach based on principal component analysis. *Journal of Process Control*, 2002(In Press).
107. Kourtis, T. and J.F. MacGregor, Multivariate SPC methods for process and product monitoring. *Journal of Quality Technology*, 1996. 28(4): p. 409-428.
108. Miller, P., R. Swanson, and C. Heckler, Contribution Plots: A missing link in multivariate quality control. *Applied Mathematics and Computer Science*, 1998. 8(4): p. 775-792.
109. Jackson, J.E., *A Users Guide to Principal Components*. 1991, New York: Wiley.
110. Jackson, J.E. and G.S. Mudholkar, Control Procedures for Residuals Associated with Principal Components Analysis. *Technometrics*, 1979. 21: p. 341-349.
111. Box, G.E.P., Some Theorems on Quadratic Forms Applied in the Study of Analysis of Variance Problems: Effects of Inequality of Variance in One-Way Classification. *Ann. Math. Stat.*, Vol. 25, 1954: p. 290-302.
112. Gleser, L.J., Estimation in a multivariate "error in variable" regression model. *Annals of Statistics*, 1981. 9: p. 24-44.
113. Gleser, L.J., Measurement error models. *Chemometrics and Intelligent Laboratory Systems*, 1991. 10: p. 45-57.

114. Hoskuldsson, A combined theory for PCA and PLS. *Journal of Chemometrics*, 1995. 9: p. 91-123.
115. Kaspar, M.H. and W.H. Ray, Chemometric methods for process monitoring and high-performance controller design. *AIChE Journal*, 1992. 38(10): p. 1593-1604.
116. Kresta, J.V., J.F. MacGregor, and T.E. Marlin, Multivariate statistical monitoring of process operating performance. *Canadian Journal of Chemical Engineering*, 1991. 69(1): p. 35-47.
117. Jackson, J.E., Principal Components and Factor Analysis: Part 1 - Principal Components. *Journal of Quality Control*, 1980. 12(4): p. 201-213.
118. Wise, B.M., Gallagher N.B. and Ricker, N.L., Theoretical basis for the use of principal component models for monitoring multivariate processes. *Process Control and Quality*, 1990. 1(1): p. 41-51.
119. Wold, S., K. Esbensen, and P. Geladi, Multi-Way Principal Components and PLS-Analysis. *Journal of Chemometrics*, 1987. 1: p. 41-56.
120. Wold, S., K. Esbensen, and P. Geladi, Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 1987. 2: p. 37-52.
121. Krzanowski, W.J., Cross-Validatory choice in principle components analysis: Some sampling results. *Journal of Statistical Computation and Simulation*, 1983. 18: p. 299-314.
122. Valle, S., W. Li, and S.J. Qin, Selection of the Number of Principal Components: The Variance of the Reconstruction Error Criterion with Comparison to other Methods. *Industrial & Engineering Chemistry Research*, 1999. 38: p. 4389-4401.
123. Joliffe, I.T., *Principal Component Analysis*. 1986, New York: Springer Verlag.
124. Hoskuldsson, PLS Regression methods. *J. Chemometrics*, 1988.
125. Geladi, P., Partial Least Squares Regression: A Tutorial. *Analytica Chimica Acta*, 1986. 185: p. 1-17.
126. De Jong, S., SIMPLS, An Alternative Approach to Partial Least Squares Regression. *Chemometrics and Intelligent Laboratory Systems*, 1993. 18: p. 251-263.

127. Kaspar, M.H. and W.H. Ray, Dynamic PLS modelling for process control. Chemical Engineering Science, 1993. 48(20): p. 3447-3461.
128. Lakshminarayanan, S., S. Shah, and K. Nandakumar, Modeling and Control of Multivariable Processes: Dynamic PLS Approach. 1997. 43(9): p. 2307-2322.
129. Chen, J. and K.C. Liu, On-line batch process monitoring using dynamic PCA and dynamic PLS models. Chemical Engineering Science, 2002. 57(1): p. 63-75.
130. Kano, M., Miyazaki, K., Hasebe, S, and Hashimoto, I., Inferential control system of distillation compositions using dynamic partial least squares regression. Journal of Process Control, 2000. 10(2): p. 157-166.
131. Simoglou, A., E.B. Martin, and A.J. Morris, A Comparison of Canonical Variate Analysis and Partial Least Squares for the identification of dynamic processes, in Proceedings of the American Control Conference. 1999, IEEE. p. 832-837.
132. Marlin, T.E., Process Control: Designing Processes and Control Systems for Dynamic Performance. 1995, Singapore: McGraw-Hill.
133. Nimmo, I., Adequately address abnormal situation operations. Chemical Engineering Progress., 1995. 91(1): p. 1361-1375.
134. Vedam, H. and V. Venkatasubramanian, PCA-SDG based Process Monitoring and Fault Detection. Control Engineering Practice, 1999. 7: p. 903-917.
135. Khalilian, M, and R. Dhib, Online Identification and Control of a Fluid Catalytic Cracking Unit. Proceedings of the IASTED International Conference on Modelling, Identification and Control, Innsbruck, Austria. Feb 18-21, 2002. 1: p. 448-452.

