© Copyright by

Barry M. Wise

1991

Adapting Multivariate Analysis for Monitoring and Modeling of Dynamic Systems

by

BARRY M. WISE

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

Doctor of Philosophy

University of Washington

1991

Approved by ____________________________________
(Chairperson of Supervisory Committee)

Program Authorized
to Offer Degree ____________________________________

Date ____________________________________

In presenting this dissertation in partial fulfillment of the requirements for the Doctoral degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of this dissertation is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for copying or reproduction of this dissertation may be referred to University Microfilms, 300 North Zeeb Road, Ann Arbor, Michigan 48106, to whom the author has granted "the right to reproduce and sell (a) copies of the manuscript in microform and/or (b) printed copies of the manuscript made from microform."

Signature ____________________

Date ____________________

University of Washington

Abstract

ADAPTING MULTIVARIATE ANALYSIS FOR MONITORING AND MODELING OF DYNAMIC SYSTEMS

by Barry M. Wise

Chairperson of the Supervisory Committee: Professor N. Lawrence Ricker
Department of Chemical Engineering

This work considers the application of several related multivariate data analysis techniques to the monitoring and modeling of dynamic processes. Included are the method of Principal Components Analysis (PCA), and the regression technique Continuum Regression (CR), which encompasses Principal Components Regression (PCR), Partial Least Squares (PLS) and Multiple Linear Regression (MLR), all of which are based on eigenvector decompositions.

It is shown that proper application of PCA to the measurements from multivariate processes can facilitate the detection of failed sensors and process upsets. The relationship between PCA and the state-space process model form is shown, providing a theoretical basis for the use of PCA in dynamic systems. For processes with more measurements than states, the deterministic variation in the output data is redundant and PCA modeling can be applied. Under these conditions the residuals of the PCA model are related only to the process measurement noise; the state of the process does not affect the residuals. Statistical limits, which define the normal amount of process noise, can be calculated for the process residuals. Failed sensors or process upsets manifest themselves as changes in the PCA residuals and can be detected through the application of statistical tests.

Collections of PLS models are used in a manner analogous to PCA for the failure detection problem. This technique can be more effective than PCA monitoring. However, the method suffers because, unlike PCA models, it maps state information into the residuals. Statistical limits on the residuals must account for this. Changes in the process inputs invalidates the calculated limits.

CR is applied to the identification of Finite Impulse Response (FIR) and Auto-Regressive eXtensive variable (ARX) dynamic models. In FIR identification, the frequency domain effects of CR, and in particular PCR, are investigated from a theoretical perspective. This results in a fundamental understanding of the effects of CR on FIR identification. Observed trends in CR identification are consistent with the theoretical understanding. CR appears to be a great advantage over existing methods for the identification of FIR models, but offers only moderate improvements for ARX models.

TABLE OF CONTENTS

List of Figures iv

List of Tables ix

1.0 Introduction and Overview 1
- 1.1 Motivation...1
- 1.2 Scope, Goals and Approach 4
- 1.3 Review: The State of Process Fault and Upset Detection.............6
- 1.4 Review: The State of Process Model Identification....................7

2.0 Background 9
- 2.1 Principal Components Analysis (PCA)...................................9
  - 2.1.1 The PCA Method...10
  - 2.1.2 The Q and T^2 Statistics 12
  - 2.1.3 Statistics Associated with PCA Residuals....................15
- 2.2 Principal Components Regression (PCR)...............................21
  - 2.2.1 Motivation for PCR..21
  - 2.2.2 The PCR Method...22
- 2.3 Partial Least Squares (PLS) Regression.................................24
  - 2.3.1 Motivation for PLS 25
  - 2.3.2 The PLS Method 25
  - 2.3.3 PLS and PCR with Non-Linear Inner Relationships 29
- 2.4 Continuum Regression 30
  - 2.4.1 Motivation...30
  - 2.4.2 Continuum Regression Method 32
- 2.5 Data Pretreatment and Scaling 35
  - 2.5.1 Types of Scaling 35
  - 2.5.2 Outliers Analysis 37
  - 2.5.3 Linearization 38
- 2.6 Statistical Process Control..38
- 2.7 Process Models..42
  - 2.7.1 State-Space Models 42
  - 2.7.2 Finite Impulse Response (FIR) Models 43
  - 2.7.3 Auto-Regressive Extensive Variable (ARX) Models 45
- 2.8 Common Process Model Identification Methods 46
  - 2.8.1 FIR Model Identification 46
  - 2.8.2 MLR and Instrumental Variables for ARX Models 48
- 2.9 The Liquid-Fed Ceramic Melter 50

3.0 Process Monitoring with PCA 53
- 3.1 A Theoretical Basis for PCA Monitoring.................................53
  - 3.1.1 Immediately Observable States 54
  - 3.1.2 The Basis for PCA Monitoring.................................58
  - 3.1.3 Obtaining an Accurate Estimate of $\mathbf{P}_q$.........................59
  - 3.1.4 Using the PCA Residuals 60
- 3.2 Some Examples of PCA Monitoring.......................................60
  - 3.2.1 PCA Monitoring Example Using Synthetic Data............60
  - 3.2.2 PCA Monitoring Example Using LFCM Data...............65
- 3.3 Effect of Scaling on PCA Monitoring.....................................69
- 3.4 Treating Autocorrelation in PCA Residuals 70

- 3.5 Robust PCA for Multiple Failures........73
- 3.6 Conclusions Concerning PCA Monitoring........81

4.0 Process Monitoring with PLS........83
- 4.1 A Basis for PLS Monitoring........83
- 4.2 Generating and Using PCA-Like Residuals with PLS........88
- 4.3 Comparison of PCA and PLS Monitoring........89
  - 4.3.1 Examples with Synthetic Data........89
  - 4.3.2 Examples with LFCM Data........97
- 4.4 Robust PLS for Multiple Failures........108
- 4.5 Conclusions Concerning PLS and PCA Monitoring........109

5.0 Dynamic Process Model Identification with CR........111
- 5.1 General Methodology........111
- 5.2 Continuum Regression for FIR Model Identification........112
  - 5.2.1 Eigenvector Decomposition of the ACM........112
  - 5.2.2 Frequency Domain Effects of PCR........121
  - 5.2.3 Effect of Filtering on ACM Decomposition........126
  - 5.2.4 Frequency Domain Effects of CR........128
  - 5.2.5 Effect of Identification Conditions on FIR Modelling........130
    - 5.2.5.1 Effect of Process Measurement Noise........131
    - 5.2.5.2 Effect of Process Dynamics........134
    - 5.2.5.3 Effect of Input Excitation Level........136
    - 5.2.5.4 Effect of Time Delays........137
    - 5.2.5.5 Effect of CR on FIR Coefficients........139
- 5.3 Continuum Regression for ARX Model Identification........141
- 5.4 PCR and PLS for Non-Linear FIR Model Identification........148
  - 5.4.1 Non-Linear Versions of PCR and PLS........148
  - 5.4.2 An Example Using Non-Linear PCR and PLS........149
- 5.5 Conclusions Concerning CR for Dynamic Modelling........154

6.0 Conclusions........155
- 6.1 Process Monitoring with PCA........155
- 6.2 Process Monitoring with PLS........157
- 6.3 Continuum Regression for Process Modeling........159
- 6.4 Suggestions for Future Work........160

References........162

Appendix 1. Dynamic Process Investigation with PCA........169
- A1.1 PCA Study of Melter Foaming........169
  - A1.1.1 Review of Original Foaming Study........170
  - A1.1.2 Re-analysis of Foaming Study Data........174
- A1.2 PCA Analysis of SF-11: The Effect of Scaling........178
- A1.3 Assessment........187

Appendix 2. Equivalence of Replacement and Rebuilding........189

Appendix 3. Comparison of t- and T^2- Statistics........193

LIST OF FIGURES

Figure 2.1. Example of Typical PRESS Plot. 24

Figure 2.2. Relationship Between Continuum Regression and PCR, PLS and MLR. 31

Figure 2.3 Continuum Regression PRESS Surface. 34

Figure 2.4. Cumulative Sum Chart. 40

Figure 2.5. Schematic Drawing of Liquid-Fed Ceramic Melter 51

Figure 3.1. Autocorrelation Function for Outputs and Residuals for Test Data Set. 63

Figure 3.2. Detection Limits for Changes in Noise Mean and Standard Deviation in Original Units. 64

Figure 3.3. Autocorrelation Function of SF-11 Data and 4 and 7 PC Model Residuals. 66

Figure 3.4. Detection Limits for Changes in Mean Residual From Mean Centered SF-11 Data Using 4 and 7 Principal Component Models. 67

Figure 3.5. Detection Limits for Changes in Standard Deviation of Residual From SF-11 Data Using 4 and 7 Principal Component Models. 68

Figure 3.6. Autocorrelation Function of SF-11 Data and 4 and 7 PC Model Residuals from Autoscaled Data. 69

Figure 3.7. Detection Limits for Changes in Mean From SF-11 Data Using 4, 7 and 10 Principal Component Models from Autoscaled Data. 71

Figure 3.8. Detection Limits for Changes in Variance From SF-11 Data Using 4, 7 and 10 Principal Component Models from Autoscaled Data. 71

Figure 3.9. Obtaining $\mathbf{R}_{11}$ and $\mathbf{R}_{21}$ from Original Rm Matrix. 76

Figure 3.10. Mapping of Regression Matrix into "Replacement" Matrix. 76

Figure 3.11. Projection of Sample Onto Model. 79

Figure 3.12. Projection of Corrected Sample $\mathbf{x}_c$ onto Model $\mathbf{p}$. 80

Figure 3.13. Projection of Corrected Sample $\mathbf{x}_c$ onto Model $\mathbf{p}$ Shown with Projection of Reduced Sample $\mathbf{x}_r$ onto New Model $\mathbf{p}_c$. 81

Figure 4.1. Percent Improvement in Sum of Squared Prediction Error for PLS Models Over PCA Models. 86

Figure 4.2. Autocorrelation Function of Process Outputs and PLS Residuals from Data Set with White Noise Input. 91

Figure 4.3. Autocorrelation Function of Process Outputs and PLS Residuals from Data Set with PRBS Input. 91

Figure 4.4. Autocorrelation Function of Process Outputs and PLS Residuals Based on PRBS Data from Data Set with PRBS Input. 92

Figure 4.5. Detection Limits for Changes in Measurement Noise Mean and Standard Deviation for PCA (solid lines) and PLS Models (dashed lines). 93

Figure 4.6. Scaled Mean of PCA Residuals and Bias Error Detection Limits Showing Apparent Bias on Variables 2 and 7.............................94

Figure 4.7. Ratio of Mean of PCA Residuals to Bias Error Detection Limits Showing Largest Bias on Variable 2. 94

Figure 4.8. Number of Correct Responses to Bias Test for Autoscaled PCA Model. 98

Figure 4.9. Number of Correct Responses to Bias Test for Mean Centered Model. 99

Figure 4.10. Number of Correct Responses to Bias Test for PLS Model.............99

Figure 4.11. Number of No Indications to Bias Test for Autoscaled PCA Model. 100

Figure 4.12. Number of No Indications to Bias Test for Mean Centered PCA Model. 101

Figure 4.13. Number of No Indications to Bias Test for PLS Model. 101

Figure 4.14. Number of Incorrect Responses to Bias Test for Autoscaled PCA Model. 102

Figure 4.15. Number of Incorrect Responses to Bias Test for Mean Centered PCA Model..103

Figure 4.16. Number of Incorrect Responses to Bias Test for PLS Model...........103

Figure 4.17. Number of Correct Responses to Bias Test for "Best" Models..........105

Figure 4.18. Number of Incorrect Responses to Bias Test for "Best" Models........106

Figure 4.19. Number of No Responses to Bias Test for "Best" Models..............106

Figure 4.20. Calculated Bias Detection Limits for "Best" Models Based on Observed Residuals in Calibration Set.....................................107

Figure 4.21. Calculated Noise Detection Limits for "Best" Models Based on Observed Residuals in Calibration Set. 108

Figure 5.1. Arrangement of Data for FIR Model Identification. 113

Figure 5.2. Arrangement of Autocorrelation Matrix for FIR Model Identification. 113

Figure 5.3. Coefficients in First Five Eigenvectors of Autocorrelation Matrix of White Noise Process Through First Order Filter. 115

Figure 5.4. Points where n = Tangent (a/n) Showing the Frequencies of the Cosine Eigenvectors.. 118

Figure 5.5. Points where n = -1/Tangent (a/n) Showing the Frequencies of the Sine Eigenvectors.. 119

Figure 5.6. Coefficients of the Cosine Eigenvectors of the Discrete Matrix Shown with Eigenfunction Solutions to the Continuous Problem.. 119

Figure 5.7. Coefficients of the Sine Eigenvectors of the Discrete Matrix Shown with Eigenfunction Solutions to the Continuous Problem.. 120

Figure 5.8. Eigenvalues of the Discrete Case Shown with Scaled Eigenvalues of the Continuous Case. 121

Figure 5.9. Example PRBS Input and Case 2 Process Output. 122

Figure 5.10. Bode Gain Magnitude Plot for First Order Process (Case 1) and PCR Models.. 123

Figure 5.11. Bode Gain Magnitude for Second Order Process (Case 2) and PCR Models. 124

Figure 5.12. Eigenvalues of Autocovariance Matrix for Original Signal and Filtered Signals as Cutoff Frequency is Lowered. 127

Figure 5.13. Coefficients of the First, Sixth and Tenth Eigenvectors of the ACM for Varying Filter Parameters. 127

Figure 5.14. Bode Gain Magnitude for 1 Latent Variable Models of Second Order Under-damped Process--No Noise. 129

Figure 5.15. Bode Gain Magnitude Plots for the Seven Test Processes. 131

Figure 5.16. Continuum Regression PRESS Surface From Models of First Order Process (Model 1) with 5% Process Noise. 132

Figure 5.17. Continuum Regression PRESS Surface From Models of First Order Process (Model 1) with 40% Process Noise.. 133

Figure 5.18. Effect of Process Noise Level on Number of Latent Variables in

Best Model for Each Continuum Parameter. 134

Figure 5.19. Effect of Process Dynamics on Number of Latent Variables in Best Model for Each Continuum Parameter. 135

Figure 5.20. Effect of Input Excitation on Number of Latent Variables in Best Model for Each Continuum Parameter. 136

Figure 5.21. Effect of Mis-Estimated Time Delay on Number of Latent Variables in Best Model for Each Continuum Parameter. 138

Figure 5.22. Model Errors for Correct and Incorrectly Estimated Time Delays. 139

Figure 5.23. FIR Coefficients Identified from First Order System with Noise = 20% of Process Gain.. 140

Figure 5.24. Bode Gain Plots of Process Model 7 and ARX Models Identified with PCR. 142

Figure 5.25. PRESS Surface for ARX Models Tested Against PRBS. 144

Figure 5.26. PRESS Surface for ARX Models Tested Against Random Input. 144

Figure 5.27. Number of Latent Variables in Best Models for Different Noise Levels, PRBS Test. 145

Figure 5.28. PRESS Surface for High Noise Low Input Excitation Case. 146

Figure 5.29. Bode Gain Plots for True Process and ARX Models Determined by CR and MLR. 147

Figure 5.30. Non-Linear Tank Calibration Data.. 150

Figure 5.31. Segment of Calibration Data Set Showing Model Fits. 152

Figure 5.32. Test Data Set Showing Actual and Predicted Process Output. 153

Figure A1.1. Loadings on First Two Principal Components in Melter Foaming Study. 171

Figure A1.2. Scores on First Two Principal Components in Melter Foaming Study. 172

Figure A1.3. Loadings for First versus Third Principal Component for Revised Melter Foaming Study. 175

Figure A1.4. Scores on First versus Third Principal Component for Revised Melter Foaming Study. 176

Figure A1.5. Loadings for First PC of Mean Centered SF-11 Data. 179

Figure A1.6. Loadings for Second PC of Mean Centered SF-11 Data.. 180

Figure A1.7. Loadings for Third PC of Mean Centered SF-11 Data.…………………180

Figure A1.8. Loadings for Fourth PC of Mean Centered SF-11 Data. ………………181

Figure A1.9. Loadings for First PC of Autoscaled SF-11 Data..……………………184

Figure A1.10. Loadings for Second PC of Autoscaled SF-11 Data. …………………184

Figure A1.11. Loadings for Third PC of Autoscaled SF-11 Data..……………………185

Figure A1.12. Loadings for Fourth PC of Autoscaled SF-11 Data.……………………185

Figure A3.1. Responses of t- and T^2-Tests to Biases on Specific Sensors..…………194

LIST OF TABLES

Table 2.1. Connection Between Variable Numbers and West Valley TAG ID……52

Table 3.1. Φ Matrix for Example System. ………………………………………61

Table 3.2. Γ Matrix for Example System.………………………………………61

Table 3.3 **C** Matrix for Example System. ………………………………………61

Table 3.4. Variance Captured by PCA Model of Example Process Output.………62

Table 3.5. PCA Loadings Vectors for Example System.…………………………62

Table 4.1. Matrix for Calculating Residuals of PLS Models with Number of Latent Variables in each Model in Square Brackets.……………………90

Table 4.2. Error Detection Simulation Results for Bias Errors. …………………95

Table 4.3. Error Detection Simulation Results for Noise Errors.…………………96

Table 5.1. Numerator [A(q-1)] and Denominator [B(q-1)] Polynomial Coefficients..……………………………………………………………122

Table 5.2. Numerator and Denominator Polynomial Coefficients for Test Models.………………………………………………………………131

Table 5.3. Model Fit Errors and Prediction Errors for Non-Linear Process ……152

Table A1.1. Percent Variance Captured by PCA Model of SF-11 Data using Mean Centering.………………………………………………………179

Table A1.2. Percent Variance Captured by PCA Model of SF-11 Data using Autoscaling.…………………………………………………………183

Table A3.1. Response of t- and T^2-Tests to Sensor or Random Biases.…………194

ACKNOWLEDGMENTS

It is doubtful that anyone makes it to the end of their dissertation without a good deal of help along the way. This author is certainly no exception. High on the list of those to thank is my advisor, Larry Ricker. I have enjoyed working with Larry a great deal. He is always very good at asking the tough question, but I have never known him to ask it with any intention other than to make me think very carefully about what I was doing. I'd like to thank Bruce Kowalski for introducing me to multivariate analysis. Bruce's enthusiasm certainly sparked my interest, the results of which you now hold in your hand. Through many of these developments Brad Holt has been my chief skeptic. I thank him for that because arguing with him has forced me to clarify my own thinking. Brad also helped me straighten out some mathematical details, for which I am equally thankful. Dave Veltkamp has made many helpful comments along the way. I have consulted him many times in order to tap into his fundamental understanding of the methods used here. Don Marshall of the Math department deserves special thanks for his help with Toeplitz operator theory. For a professor to spend as much time with someone who just walked in from "out of the blue" as Don did with me was definitely beyond the call of duty. I'd also like to thank Dan Haesloop for sharing his neural network data, making for some interesting comparisons.

For their financial support, I would like to thank Battelle Pacific Northwest Laboratories. Rick Brouns was particularly helpful in setting up the original contracts which led to my master's degree and eventually this. Also, I would like to thank West Valley Nuclear Services Co., Inc. who supported the bulk of this research. The National Science Foundation also contributed to this project and I am particularly grateful for the Macintosh IIx computer purchased with NSF funds.

On the non-technical side, I would like to thank Norman McCormick for his mentorship. Norm was always looking out for my best interests, even, or maybe especially, when I wasn't. I'd like to thank Neal Gallagher for the almost daily, always irrelevant E-mail messages, which always made me look forward to going into the office. I have also appreciated Dave Williams' friendship and our many conversations and outings. Special thanks goes to the Schirmers and the Buskes for general moral support when I was a single guy. I would like to thank my parents for both their financial and moral support, and for teaching me that effort was at least as important as accomplishment. My in-laws have also been very encouraging and helpful. I am grateful for this, since they must have wondered at times why their daughter would want to marry someone who was 30+ years old and still in school. Finally, and perhaps most emphatically, I would like to thank my wife Jill for putting up with me even when this project turned me into an ogre. Jill is certainly the best thing that happened to me in graduate school.

1.0 Introduction and Overview

In today's highly competitive industrial environment, better control of chemical processes is an important step towards increasing the efficiency of production facilities. Improved process control can have a positive impact on chemical processes in several ways: improved compliance with increasingly stringent environmental regulations, reduced generation of hazardous wastes and more consistent quality of the final product . This work concerns the chemical process monitoring and modeling problem, specifically process sensor fault detection, process upset detection and process model identification. These related problems are addressed with a closely related set of methods or tools. The methods considered are multivariate techniques based on eigenvector decompositions. The overall goal of this work is to test the applicability of these methods for process monitoring and modeling and to adapt them, when necessary, for use with dynamic systems. As such, this work should be considered an exercise in methods development.

1.1 Motivation

Process monitoring, with the goal of detecting failing sensors and process upsets, is important for reasons of safety and process efficiency. Process control action based on faulty sensors is at best inefficient and at worst dangerous. Process upsets or disturbances can also lead to operating inefficiencies, such as increased process down time or off-specification product. Timely identification of failed sensors and upsets is, therefore, an integral aspect of the overall process control problem.

Recent advances in process instrumentation and data collection techniques have resulted in a rapid increase in the amount of data coming from chemical processes. Today's processes are typically monitored at more locations and by instruments that provide more measurements than in the past. However, process operators (and many control systems) typically rely on a few key variables and largely ignore the bulk of the data. This is due, in part, to the largely redundant nature of the data produced: many of the variables measured are very highly correlated. While much of this data is often highly correlated, it can also be true that there is more information in the data than might be at first suspected. Information of this type is potentially useful, then, for two distinct reasons. If the significant information can be extracted from the data it can lead to a deeper understanding of the the process and can potentially be used for predicting some otherwise unmeasured output. The redundancy of the process variables, on the other hand, can be modeled. These models can then be used to check new process data for changes in the process or its sensors.

Extracting the significant information from the plethora of data produced by heavily instrumented processes can be difficult. The addition of process sensors has also increased the complexity of the fault detection and diagnosis aspects of the monitoring problem. Fortunately, largely because of the present availability of computers, this data is now beginning to be utilized much more fully. Much of the data processing, however, requires multivariate analysis methods. Multivariate methods are necessary due to the data "reduction" or "compression" that they provide. Variables which contribute essentially the same information must be combined in a logical fashion in order to reduce the effective number of variables. This should serve to keep the process personnel from suffering from "information overload". Secondly, the multivariate calibration aspects of the methods used must be considered. The methods should be "robust" to the redundancy in the data which can cause conventional methods to produce results that are highly sensitive to small perturbations or noise in the process data.

Historically, chemical process modeling has been done in order to increase the understanding of the process in the hopes of increasing process operating efficiency (optimization). This process modeling has generally been done from a theoretical standpoint. Models of processes were built up from the fundamental equations of heat and

mass transport, chemical reaction kinetics and material and energy balances. Modeling has become an increasingly important aspect of chemical process control.
This is evidenced by the progression of controller design methods. The modern era of control started with the advent of the proportion/integral/derivative (PID) controller. Even PID controllers, however, rely on a simple implicit model of the process that includes information such as process response time. Controller design advanced to using process models in the actual setting of the PID tuning parameters. Recently, advances in process control algorithms, particularly the model based controller design methods, have further increased the reliance on process models. There are several popular techniques, such as Internal Model Control (IMC), Model Predictive Control (MPC), and Quadratic Dynamic Matrix Control (QDMC), which rely on an explicit model of the process as part of the the controller. These techniques have increased the attention given to dynamic process model identification. In fact, when the model-based design schemes are used, the controller design process is largely complete as soon as the process model (and in some cases the model error bound) has been specified. Better process models, therefore, should result in better process control.

There are several advantages that theoretical models have over statistical/empirical models. For instance, it is more appropriate to extrapolate beyond experimentally verified data when using a theoretical model than when using an empirical model. A deeper understanding of the modelled process may also result from theoretical model development. However, theoretical models are often compromised because they require so many simplifying assumptions in order to be tractable that they are often biased. There is also the possibility that the modelers may build in some of their prejudice about how the process should work. Furthermore, developing a theoretical model of a complex process can be very time consuming.

Because of the increasing reliance on process models and the difficulties with obtaining them directly from theory, the task of identifying dynamic models from plant data has become an increasingly important one. Unfortunately, process model identification is complicated by the fact that actual plant data may suffer from any or all of the following problems: high noise levels, unmeasured process disturbances, short data records or correlation in the input parameters. Any of these difficulties may cause the identification problem to be poorly conditioned. In these cases the identified models are highly sensitive to small changes in the data and can vary widely depending upon which subset of the available data was used in their calibration.

It appears that these monitoring and modeling problems could be attacked successfully using several related multivariate statistical techniques. These are Principal Components Analysis (PCA), a pattern recognition and data modeling technique, and the regression methods that may be grouped under the unifying theme of Continuum Regression (CR). The biased regression methods Partial Least Squares (PLS) and Principal Components Regression (PCR), as well as conventional Multiple Linear Regression (MLR), can all be shown to be special cases of CR. While PCA, PCR, PLS and MLR have been used extensively in other fields, such as analytical chemistry calibration, this work represents a departure because of the dynamic nature of the monitoring and modeling problem. There are many things to be learned regarding the interpretation and applicability of these methods for use with systems where time is a variable. Furthermore, there has been very little work done in any discipline using CR. Thus this work provides some additional insight into the behavior of the CR method.

1.2 Scope, Goals and Approach

This work is limited to the monitoring and modeling of time invariant processes, with an emphasis on linear processes. A limited number of examples of non-linear model identification are considered. There are many model forms, however, only Finite Impulse Response (FIR) and Auto-Regressive eXtensive (ARX) models will be considered in the process model identification sections. For monitoring aspects of the the work, the state-space model format will be introduced.

In general, the goal of the research was to determine how PCA, PCR and PLS could be used to enhance process monitoring and control. This general goal was broken down into three areas of application: process monitoring, dynamic model identification and process analysis. In each of these three areas there are specific questions which were attacked. These are outlined below.

<u>Process monitoring:</u> Can PCA and PLS be used as effective process monitoring tools? Earlier studies have shown that PCA appears to model the "normal" process variation and have indicated that PCA may be useful for identifying process upsets and failed sensors. Other studies have shown that PLS can be a used as a monitoring tool in a fashion quite similar to PCA. Can statistical limits be developed around the methods so that they can be used in a straightforward fashion for fault detection? How does PCA relate to the process models normally used and what are the special considerations that must be (if any) made when dealing with time series data? These issues are considered in Chapter 3 which focuses on PCA and Chapter 4 which focuses on PLS.

<u>Dynamic model identification:</u> What are the special effects encountered when the biased regression techniques encompassed by CR are used to identify dynamic process models? How do the biases manifest themselves in the resulting models? How can the methods be interpreted in conventional control/modeling terms such as frequency response analysis. What affect do process noise level, dynamics, input excitation, data record length and process dead time have on the relative advantage of the CR method over conventional techniques? How do these factors affect the location of the optimum model in the CR parameter space? Chapter 5 addresses these issues.

<u>Process analysis:</u> What can be learned about dynamic multivariate processes using PCA? Studies from other fields have indicated that PCA can be useful as a pattern

recognition technique. Often, otherwise unrecognized relationships between variables and samples are made apparent when the data is subjected to PCA. Does this hold for dynamic process data? Results in this area are somewhat limited and rather subjective. This work is covered in the Appendix.

The approach to these questions was to derive as many of the answers from theory as possible. When this was not possible, certain techniques were demonstrated/tested with extensive simulations. Finally, examples of the use of techniques are given using actual process data.

1.3 Review: The State of Process Fault and Upset Detection

Much has been written about process fault detection, however, the basic approach can be summarized in a few short sentences. Generally, a model of a process is developed, either through theoretical or empirical means. The model is used to predict process outputs which are then compared to actual outputs. The differences between actual and predicted outputs, the residuals, are then subjected to some sort of statistical test to determine if they are significant. This general approach is outlined in the references by Himmblau (1978), Willsky (1976) and Iserman (1984). The major difference in the methods lies in the types of models used for output prediction and the type of statistical test applied to the residuals. For instance, a large body of work is available concerning statistical significance tests for use with the autocorrelated residuals that are often produced, such as the work by Harris (1989) and Alt et. al. (1977). The body of literature concerning identification of process models is also relevant to fault detection but this will be dealt with more fully in the next section.

Using PCA to develop a model which directly related process outputs to each other for process monitoring is a new approach. Some work has been done where PCA was used on process residuals as shown in Subba and Rao (1974 and 1976). It is possible that the approach taken here has not been used before because of the limited number of systems to which it has been applicable in the past. As will be shown, the PCA/PLS monitoring techniques are best suited to systems that are heavily instrumented.

1.4 Review: The State of Process Model Identification

Not surprisingly, the method used for identifying a process model is generally dependent upon the model form chosen. Thus, the many models forms used have given rise to a large number of identification techniques. The main interest here is the identification of FIR and ARX models, therefore, this discussion will be limited to methods typically applied for identification of these models.

There is a surprising dearth of information concerning the identification of FIR models, especially considering their use in several controller design schemes. For instance, in the definitive work by Ljung (1987), FIR models are scarcely mentioned, and no discussion of methods suited to their identification is made. It appears that, in general practice, most FIR models are generated by doing single pulse or step tests on the process inputs and getting the FIR coefficients directly. A notable exception to this is the work by Ricker (1988) where PLS and a method based on the Singular Value Decomposition (SVD) were used to obtain FIR models. In Ricker's work a Pseudo-Random Binary Sequence (PRBS) was used as an input to the process. While it is possible to use MLR to obtain FIR coefficients, the author has found that this approach generally does not produce satisfactory results: the correlation in the FIR coefficients makes the MLR solution somewhat unstable. The models obtained often have spurious variations in the coefficients. To a large extent, this work can be considered an extension of that done by Ricker.

Identification of ARX models is typically done either by MLR or with the method of Instrumental Variables (IV) (Ljung 1987, Soderstrom and Stoica 1983). MLR is more satisfactory for use with ARX models than with FIR models. This is because the smaller number of parameters, and reduced correlation in the parameters of ARX models makes the problem solution more stable.

Many more complex model forms, particularly those that include explicit noise models, are typically identified with prediction error methods (PEM) (Ljung 1987). In these algorithms various search techniques are employed to identify the model parameters that provide the best prediction of the process outputs. These search techniques are employed because these more complex models are inherently non-linear in the parameters. The conventional PEM is not necessary for identification of FIR and ARX models, and in some cases, (depending upon how it is employed), can be shown to reduce to the MLR solution.

2.0 Background

The ground work for this research comes from many areas. As much of this information as is practical is included here so that the reader will not be required to go to many other sources in order to understand the developments in this work. This chapter includes sections on the data modeling technique PCA, the multivariate calibration methods covered by CR (MLR, PLS and PCR) and statistical process control. In addition, some common process model forms and identification techniques will be reviewed. It is beyond the scope of this document to give a complete treatment of all of these fields but enough background information is included so that the reader may follow the developments here. In addition, descriptions of the processes that we have used to test the techniques developed here are included.

Most of the mathematics covered in this chapter are well known. However, in some cases minor extensions to known methods, developed by the author, are included for the sake of continuity. Such extensions are noted in the text. Major developments/extensions are covered in subsequent chapters.

2.1 Principal Components Analysis (PCA)

The mathematical ideas behind the PCA method have been known since the times of Gauss, although efficient means of calculating the eigenvectors and eigenvalues of matrices were not discovered until much later (Strang 1980). The technique was not used for data analysis until this century when it became a popular pattern recognition/factor analysis technique in the field of psychometrics (the application of statistical methods to psychology). The development of PCA and many other statistical techniques was driven, in part, by the need to analyze data sets with many correlated variables and large amounts of noise or uncertainty. Psychometricians face problems of this type quite often and this work owes much to them. The PCA technique has also been used in the field of econometrics, and more recently in chemometrics. PCA and many other multivariate techniques were slower to catch on in the hard sciences largely because the data is better in these fields, *i.e.*, there are smaller uncertainties and fewer confounding influences in the data. As problems with more variables and higher noise levels have become common in the chemical fields the need for chemometrics has increased.

2.1.1 The PCA Method

In PCA an m by n data matrix $\mathbf{X}$ is decomposed into the sum of the product of n pairs of vectors (Jackson (1976,1980A), Sharaf et. al. (1986), Geladi and Kowalski (1986)). Each pair consists of a vector in n called the loadings, $\mathbf{p}_i$, and a vector in m referred to as the scores, $\mathbf{t}_i$. Thus $\mathbf{X}$ can be written as

$$\mathbf{X} = \mathbf{t}_1\mathbf{p}_1^T + \mathbf{t}_2\mathbf{p}_2^T + ... + \mathbf{t}_n\mathbf{p}_n^T \tag{2.1}$$

The matrix of loadings vectors $\mathbf{P}$ forms a new orthogonal basis for the space spanned by $\mathbf{X}$ and the individual $\mathbf{p}_i$ are the eigenvectors of the scatter matrix of $\mathbf{X}$, defined as:

$$\text{scatter}\,(\mathbf{X}) = \frac{1}{\text{m - 1}}\left(\mathbf{X}^T\mathbf{X}\right) \tag{2.2}$$

Thus

$$\text{scatter}\,(\mathbf{X})\mathbf{p}_i = \lambda_i\mathbf{p}_i \tag{2.3}$$

where λ_i is the eigenvalue associated with the eigenvector $\mathbf{p}_i$. If the variables (columns) of $\mathbf{X}$ have been mean-centered, (mean subtracted off each variable to produce variables of mean zero), the scatter matrix defined by equation (2.2) becomes the covariance matrix of $\mathbf{X}$. If the variables have been autoscaled (mean-centered and divided by the standard deviation to produce variables of zero mean and unit variance) the scatter matrix becomes the correlation matrix. The loadings vectors $\mathbf{p}_i$ are often referred to as principal components, or as "latent variables" (particularly in PLS) because they are linear combinations of the original variables that together explain large fractions of the information in the original matrix. Each of the the $\mathbf{t}_i$ is simply the projection of $\mathbf{X}$ onto the new basis vector $\mathbf{p}_i$:

$$\mathbf{t}_i = \mathbf{X}\mathbf{p}_i \qquad (2.4)$$

The value of each λ_i is an indicator of the covariance in the data set in the direction $\mathbf{p}_i$. In fact

$$\text{fraction variance in direction } \mathbf{p}_i = \lambda_i/\Sigma\lambda_i \qquad (2.5)$$

In a data set that has been scaled to have variables of zero mean and unit standard deviations (autoscaled)

$$\Sigma\lambda_i = n \qquad (2.6)$$

where n is the number of variables in the data set. In this case, each of the scores vectors $\mathbf{t}_i$ will then have a mean of zero and a standard deviation equal to λ_i. The scores can be adjusted to unit variance (which is convenient for other statistical tests, as will be shown) by dividing through by the associated eigenvalues

$$\mathbf{t}_{i,adj} = \mathbf{t}_i/\lambda_i \qquad (2.7)$$

PCA is very closely related to the Singular Value Decomposition (SVD) (Strang 1980) where a data matrix $\mathbf{X}$ is decomposed as

$$\mathbf{X} = \mathbf{USV}^T \qquad (2.8)$$

where $\mathbf{V}$ contains the eigenvectors ($\mathbf{p}_i$) and $\mathbf{S}$ is a diagonal matrix containing the square roots of the eigenvalues (the singular values) of the covariance matrix of **X**.

Once the eigenvectors have been determined using PCA or SVD, projections of the data onto the eigenvectors can be made. These projections are commonly referred to as "scores plots" and are often useful for showing the relationships between the samples (rows) in the data set. Plots can be done as the projections of the samples onto a single eigenvector versus sample number (or time) or onto the plane formed by two eigenvectors. A projection of the samples onto the two eigenvectors associated with the largest eigenvalues depicts the largest amount of information about the relationship between the samples that can be shown in two (linear) dimensions. It is for this reason that PCA is often used as a pattern recognition and sample classification technique.

Plots of the coefficients of the eigenvectors themselves, known as "loadings plots", show the relationships between the original variables in the data set. Correlations between variables show up in the loadings plots.

2.1.2 The Q and T^2 Statistics

When PCA is done on a data set it is often found (and it is generally the objective) that only the first few eigenvectors are associated with systematic variation in the data and that the remaining eigenvectors are associated with noise. Noise in this case refers to uncontrolled experimental and instrumental variations arising from random processes. PCA models are formed by retaining only the eigenvectors which are descriptive of systematic variation in the data. Determination of the proper number of eigenvectors can be done by cross-validation or other techniques, as pointed out in the works of Malinowski (1977a, 1977b, 1987). Once the PCA model is formed new data can be viewed as projections onto single eigenvectors (scores plots) or the plane formed by pairs of eigenvectors. The scores can be used to obtain the "PCA estimate" of a given sample, *i.e.* the projection of the sample into the PCA model. For a reduced order model $\mathbf{P}_k$ (where only the first k of the n total eigenvectors were retained) and a new sample $\mathbf{x}_i$ this is obtained from:

$$\hat{\mathbf{x}}_i = \mathbf{T}_i\mathbf{P}_k^T = \mathbf{x}_i\mathbf{P}_k\mathbf{P}_k^T \tag{2.9}$$

where $\mathbf{T}_i$ is the vector of scores on the model $\mathbf{P}_k$ for sample $\mathbf{x}_i$.

The "goodness" of fit between new data and the model can be monitored by calculating the data residual. The residual $\mathbf{r}_i$ for sample $\mathbf{x}_i$ is given by

$$\mathbf{r}_i = \mathbf{x}_i - \hat{\mathbf{x}}_i = \mathbf{x}_i(\mathbf{I} - \mathbf{P}_k\mathbf{P}_k^T) \tag{2.10}$$

The magnitude of the residual for any sample $\mathbf{x}_i$ is

$$Q = \|\mathbf{r}_i\| = \mathbf{r}_i^T\mathbf{r}_i = \mathbf{x}_i^T(\mathbf{I} - \mathbf{P}_k\mathbf{P}_k^T)\mathbf{x}_i \tag{2.11}$$

and expresses the "goodness of fit" of the new sample to the model $\mathbf{P}_k$ as a scalar. It can be calculated by taking the sum of squares of the components of $\mathbf{r}_i$. Jackson (1976, 1979, 1980, 1981) used the results of Jensen and Solomon (1972) to show that approximate confidence limits can be calculated for the model residual Q provided that all the eigenvalues of the covariance matrix are known, as shown below:

$$Q_\alpha = \Theta_1 \left[\frac{c_\alpha\sqrt{2\Theta_2 h_0^2}}{\Theta_1} + 1 + \frac{\Theta_2 h_0(h_0 - 1)}{\Theta_1^2} \right]^{\frac{1}{h_0}} \tag{2.12}$$

where

$$\Theta_i = \sum_{j = k+1}^{n} \lambda_j^i \text{ for } i = 1,2,3 \tag{2.13}$$

and

$$h_0 = 1 - \frac{2\Theta_1\Theta_3}{3\Theta_2^2} \tag{2.14}$$

In (2.12) above c_α is the normal deviate corresponding to the upper $(1 - \alpha)$ percentile. Equation (2.13) simply states that the Θ_i are equal to the sum of the eigenvalues for the eigenvectors not used in the model taken to the $i^{\underline{th}}$ power. Note, however, that this result was derived assuming random errors of mean zero etc. It is not clear how autocorrelated data from a real process would affect this result. However, Jensen and Solomon (1972) point out that the Q statistic changes little even when the underlying distribution of the original data differ substantially from normal.

The variables responsible for large Q values can often be found through normal statistical process control methods which track single variables. However, there are instances when these methods fail to detect systematic changes in the process or its sensors because the values of the individual variables have not gone "out of bounds" but have instead just become uncorrelated (or changed their correlation) with the remaining variables.

There are several methods for determining the source of the large Q values in this case. The simplest method is to calculate the column norm (the sum of squares over the variables, instead of over the samples, as is done to calculate Q values) for the residuals matrix for the samples with large Q values. Generally, the perturbed variables will show up as having the largest residuals. In other cases the factors responsible for large values of Q can be found by subjecting the matrix of $\mathbf{r}_i$ vectors to PCA. This determines the major source of variation in the data not accounted for by the original PCA model. Typically the variable with the largest (absolute value) coefficient in the first eigenvector from the residuals matrix will be the variable responsible for the deviation of the PCA model. Statistical tests for identifying sources of large Q values are developed in the following sections.

While the Q statistics offer a way to test if the process data has shifted outside the normal operating space, there is a need for a statistic that provides an indication of unusual variability within the normal subspace. This is provided by Hotellings T^2 statistic (Hotelling 1947, Jackson 1980). The value of T^2 for one sample is equal to the sum of squares of the adjusted (unit variance) scores on each of the PCs in the model. That is:

$$T^2 = \sum_{i=1}^{k} \left(\frac{t_i}{\lambda_i}\right)^2 \tag{2.15}$$

Here k is the number of principal components retained in the model. In words, T^2 is the squared length of the projection of the current sample into the space spanned by the PCA model of the data. As such, it is an indication of how far the PCA estimate of the sample (as given by equation 2.9) is from the multivariate mean of the data. The statistical confidence limits for the values of T^2 can be calculated by taking advantage of the statistical F-distribution as follows

$$T^2_{k,m,\alpha} = \frac{k(m-1)}{m-k} F_{k,m-k,\alpha} \tag{2.16}$$

Here m is the number of samples in the data set used in the calculation of the PCA model, k is the number of principal component vectors retained and α corresponds to the standard normal deviate.

2.1.3 Statistics Associated with PCA Residuals

The sample variance of the residual for each variable can be calculated for the PCA models. It can be shown that for a given data set, **X**, with a full set of PCA vectors, **P**, of which k are retained, and eigenvalues λ, then the variance of the residual for the j^{th} variable is

$$s_j^2 = \sum_{i=k+1}^{n} p_{ij}^2 \lambda_i \qquad (2.17)$$

where p_{ij} is the loading of the $j^{\underline{th}}$ variable in the $i^{\underline{th}}$ PC. Equation (2.17) is a direct result of the fact that the variance in the direction of a particular eigenvector is equal to the associated eigenvalue as indicated by equations (2.5) and (2.6). For the data matrix from which the model was obtained equation (2.17) will be exact. If, however, it is assumed that the eigenvalues of all the PCs not retained in the model are equal (which is a common assumption when they are not used[1]) then the variance in the residual of the $j^{\underline{th}}$ variable can be estimated using only the PCs and eigenvalues retained in the model from

$$\hat{s}_j^2 = \left(\sum_{i=1}^{n} \lambda_i - \sum_{i=1}^{k} \lambda_i\right)\left(1 - \sum_{i=1}^{k} p_{ij}^2\right) \qquad (2.18)$$

The first term on the right hand side of equation (2.18) can be replaced with the total sum of squares, which is equal to the sum over all of the eigenvalues. If this is done there is no need to calculate any of the principal components which are not retained in the model.

In chemical processes where there are many types of sensors in use (*e.g.* temperature, pressure, pH etc.), the amount of noise for each sensor may vary widely. In this case it is probably best to use equation (2.17) to estimate the variance of the residuals for each variable. For applications where the sensors are all closely related equation (2.18) may be more appropriate. An example of this situation is spectroscopy, where the variables are absorbances at specific wavelengths which may be measured by a single sensor or by a series of identical sensors.

The statistical properties of the residuals calculated above can be used with hypothesis testing to check for failed sensors and changes to the process. This is the general method outlined in Mehra and Peschon (1971), and is the basis for many of the methods surveyed in Willsky (1976), Isermann (1984) and Basseville and Benveniste (1986). However, these methods require a complete dynamic model of the process. Also, it should be noted that all of the methods mentioned here assume that the residuals are "white", *i.e.* are uncorrelated in time. The conditions under which the PCA residuals may be considered "white" is addressed in the next chapter.

---

[1] See for example Malinowski (1977a, 1977b and 1987). In these works the number of principal components is determined by testing the hypothesis that the eigenvalues of the unretained principal components are equal.

It is possible to compare the observed andexpected (as calculated by (2.17) or (2.18)) variance of the residuals of individual variables in order to identify changes to the system and its sensors. The residuals on the individual variables are not independent of each other; they have only as many degrees of freedom as the number of unused PCs in the data model. However, if the PCA model has captured the deterministic variation in the data set and the remaining variation is due to white noise then the individual residuals will be normally distributed. (This will be discussed in more detail in the next chapter.) In this case, the standard F-test with the appropriate degrees of freedom may be used. The test will check to see if

$$\frac{s^2_{j_{new}}}{s^2_{j_{old}}} > F_{v_{new},\, v_{old},\, \alpha} \tag{2.19}$$

where

$$v_{new} = m_{new} - 1 \qquad (2.20)^{[2]}$$

$$v_{old} = m_{old} - k - 1 \tag{2.21}$$

Here m_{new} and m_{old} are the number of samples in the test data set and training data set, respectively, and k is the number of PCs retained in the model. When the inequality in equation (2.19) holds then a change in the variance of the residual has occurred to a confidence level of 1 - α. The F-test parameters can now be used to set upper and lower limits on the variance of the residuals.

The mean residual should be zero for all the variables. The t-test can be used to detect a shift in the mean away from zero. In this case the hypothesis that the means are equal is to be tested. Thus the t-test reduces to

---

[2] In Wise (1989) and Wise (1990) this was erroneously reported as $v_{new} = m_{new} - k - 1$.

$$t_{v\,t\alpha} = \frac{\left(\overline{x_{old}} - \overline{x_{new}}\right)\left(v_{old} + v_{new}\right)^{0.5}}{\left(1/v_{old} + 1/v_{new}\right)^{0.5}\left(v_{old}s_{old}^2 + v_{new}s_{new}^2\right)^{0.5}} \tag{2.22}$$

where the degrees of freedom are both one greater than for the case given above. For the purpose of setting limits, the variances can be assumed to be equal to the variance of the residuals of the calibration set as calculated by (2.17) or (2.18), as appropriate. Once the desired confidence level is chosen, it is possible to solve for the difference between the old and new means that is just significant.

Note that the variance test in (2.19) requires at least 2 samples, while the mean test in (2.22) could be done (in principle) with 1 sample. In practice, while it would be possible to base the tests on 2 or 1 samples, respectively, a “window” of samples would be used in order to increase the sensitivity of the methods. Better sensitivity to changes in the system is obtained by looking at a series of residuals from recent samples. The number of past samples in the series (the width of the window) would be based on the response time and sensitivity desired for the detection scheme. Wide windows which consider many samples would allow detection of smaller changes, but would not respond as quickly to large changes as narrow windows.

It is also possible to apply a T^2 test to all of the residuals collectively in order to detect a shift of the multivariate mean residual of the process. This test is outlined in Anderson (1984) and was employed by Ricker (1990). In this application for a window of N samples the T^2 statistic is:

$$T^2 = N\,\bar{\eta}^T \mathbf{S}^{-1}\bar{\eta} \tag{2.23}$$

In equation (2.23) $\bar{\eta}$ is the mean of the N scaled independent residuals η_i which can be calculated from each sample $\mathbf{x}_i$:

$$\eta_i = \mathbf{x}_i \mathbf{P}_r \Lambda_r^{-1} \qquad (2.24)$$

where $\mathbf{P}_r$ is the matrix of principal components not retained in the model and Λ_r is the diagonal matrix of their associated eigenvalues. Thus at time t:

$$\bar{\eta}(t) = \frac{1}{N}\sum_{i=1}^{N} \eta(t - i + 1) \qquad (2.25)$$

The covariance of the mean centered independent residuals $\mathbf{S}$ at time t can be calculated from

$$\mathbf{S}(t) = \frac{1}{N-1}\sum_{i=1}^{N} \left[\eta(t-i+1) - \bar{\eta}\right]^{\mathbf{T}}\left[\eta(t-i+1) - \bar{\eta}\right] \qquad (2.26)$$

The statistical limit on T^2 is then calculated as

$$T^2_{r,N,\alpha} = \frac{r(N-1)}{N-r} F_{r,N-r,\alpha} \qquad (2.27)$$

where r is the number of principal components not retained in the model.

Ricker (1990) has shown that for a given bias direction, $\mathbf{b}_u$, it is possible to estimate the minimum detectable bias magnitude, ν_{min}:

$$\nu_{min} = \pm\sqrt{\frac{T^2_{r,N,\alpha}}{N\mathbf{b}_u\mathbf{G}_r\mathbf{G}_r^T\mathbf{b}_u^T}} \qquad (2.28)$$

where

$$\mathbf{G}_r = \mathbf{P}_r\Lambda_r^{-1} \qquad (2.29)$$

Ricker notes that the estimate of minimum detectable bias given in (2.28) will be degraded when $\mathbf{S} \neq \mathbf{I}$ and when the PCA model is not accurate. In spite of this, it does offer a useful indication of the sensitivity of the method for particular bias directions.

In order to use the T^2 test for a change in the mean residual, the sample window N must be greater than the number of unused principal components r. This is evident from equation (2.27), where it is apparent that in order for the degrees of freedom in the F test to be a positive values N must be greater than r.

In order to determine the “detection power” of the PCA model, the control limits must be converted back to the original units of the data. This is done by first determining the vector of ratios, $\mathbf{h}$, of the change in a variable to the change in its residual (with all other variables remaining constant). It can be shown that this is equal to the inverse of the diagonal elements of $(\mathbf{I} - \mathbf{P}_k\mathbf{P}_k^T)$, thus

$$\mathbf{h} = (\mathrm{diag}(\mathbf{I} - \mathbf{P}_k\mathbf{P}_k^T))^{-1} \tag{2.30}$$

The PCA detection limits for changes in the residual mean (calculated from either equation 2.22 or 2.28) can then be scaled by the $\mathbf{h}$ vector to obtain the detection limits in terms of the original variables. In order to obtain the detection limits for changes in the residual variance (from equation 2.19), the PCA limits must first be converted to standard deviations, then scaled and converted back to variance limits. The result of these scaling operations is that detection limits will be established in terms of the measurement units of the original variables. These limits are an estimate the smallest amount of sensor bias (arising from sensor drift or a change in the process) and added noise (arising from added measurement noise) which may be detected by PCA at the given confidence level.

2.2 Principal Components Regression (PCR)

The first regression method that will be considered in this study is Principal Components Regression (PCR). It a natural progression to go from PCA, which uses principal components to model a single block of data, to PCR, which uses the principal components of one block of data to build up a correlation between that data and another data vector.

2.2.1 Motivation for PCR

The basic idea behind PCR, and the related biased regression method PLS, is that not all of the variation in the independent data block is predictive of the dependent variables. In PCR the underlying assumption is that changes in the independent variables that lie along the directions of greatest variation tend to be causally related to variation in the dependent variables. Independent variable variations in other directions, while possibly correlated with the dependent variable, are not as predictive due to the corrupting influence of noise. Another way of looking at PCR is from the point of view of the stability of the regression problem. PCR can be thought of as a way to get around the sensitivity that the regression problem may have to small changes in the data. In MLR, if the independent data block is nearly rank deficient (the covariance matrix has some eigenvalues near zero), then the solution to the normal regression equation can change drastically for just a small change in the data. Thus a small amount of corruption from noise can make a big difference in the regression vector obtained. (See, for instance, the example in Strang 1980.)

2.2.2 The PCR Method

The PCR method is outlined in several articles and texts such as Geladi and Kowalski (1986), Lorber et. al. (1987) and Naes and Martens (1988). However, the method will be reviewed briefly here. Let the block of independent variables be called $\mathbf{X}$, where the columns represent the different variables and the rows are the samples. Let the vector of measurements of the dependent variable be called $\mathbf{y}$. Assume that the PCA decomposition of the $\mathbf{X}$ block has already been performed. The $\mathbf{y}$ vector can now be regressed against the first k $\mathbf{X}$ block scores using the normal regression equation. The result is the vector of regression coefficients $\mathbf{b}$ which relate the $\mathbf{X}$ block scores to the $\mathbf{y}$ vector. Here $\mathbf{T}_k$ is the matrix of the first k scores vectors $\mathbf{t}_i$ which are retained in the model of the **X** block.

$$\mathbf{b} = (\mathbf{T}_k^T\mathbf{T}_k)^{-1}\mathbf{T}_k^T\mathbf{y} \tag{2.31}$$

The regression vector is obtained by multiplying the regression coefficients **b** by the **X** block loadings vectors $\mathbf{P}_k$

$$\mathbf{r} = \mathbf{P}_k\mathbf{b} \tag{2.32}$$

The estimates of the dependent variables are now obtained by multiplying the **X** block by the regression vector

$$\hat{\mathbf{y}} = \mathbf{X}\ \mathbf{r} \tag{2.33}$$

As mentioned above, the PCR problem is well conditioned. It replaces the calculation of the inverse of the covariance matrix, $(\mathbf{X}^T\mathbf{X})^{-1}$, with calculation of the inverse of the scores covariance, $(\mathbf{T}_k^T\mathbf{T}_k)^{-1}$. The advantage exists because, if the data is nearly rank deficient, $(\mathbf{X}^T\mathbf{X})$ is nearly singular and has an unstable inverse. Because of the orthogonality of the scores vectors in PCA $(\mathbf{T}_k^T\mathbf{T}_k)$ is perfectly conditioned. It is zero everywhere except on the diagonal; thus its inverse is simple to compute.

The critical decision in PCR involves the number of components k to retain when building the regression model. This is generally answered through application of a cross-validation procedure. Typically, the data set is split into s subsets, with s depending upon the number of data samples available. All of the data, less one of the subsets, is then used to calculate up to n regression models $\mathbf{r}_{ij}$ (where n is the number of variables in the data set). This procedure is repeated s times, leaving out a different subset of the data each time. Thus, each regression model $\mathbf{r}_{ij}$ is calculated using a different number of principal components i and without a particular subset of the data j:

$$\mathbf{r}_{ij} = \mathbf{P}_{i\bar{j}}\left(\mathbf{T}_{i\bar{j}}^{T}\mathbf{T}_{i\bar{j}}\right)^{-1}\mathbf{T}_{i\bar{j}}^{T}\mathbf{y}_{\bar{j}} \tag{2.34}$$

where $\mathbf{P}_{i\bar{j}}$ is the matrix containing the first i principal components of the **X** block without the $j^{\underline{th}}$ subset, $\mathbf{T}_{i\bar{j}}$ is the corresponding scores matrix and $\mathbf{y}_{\bar{j}}$ is the **y** block without the $j^{\underline{th}}$ subset. Each of these models is then tested for its ability to predict the data that was not used in the calculation of the regression vector. A Predictive Residual Error Sum of Squares (**PRESS**) can be calculated for each regression model. This procedure is repeated once for each subset of the data and the final **PRESS** is the sum over the subsets j for each number of principal components i:

$$\mathbf{PRESS}_i = \sum_{j=1}^{s} \left[\mathbf{y}_j - \mathbf{X}_j\mathbf{r}_{ij}\right]^T \left[\mathbf{y}_j - \mathbf{X}_j\mathbf{r}_{ij}\right] \tag{2.35}$$

where $\mathbf{y}_j$ and $\mathbf{X}_j$ are the $j^{\underline{th}}$ subsets of the **y** and **X** data blocks, respectively.

It is generally found that initially the PRESS declines as principal components are added to the regression (starting from one principal component). However, the PRESS usually goes through a minimum and starts to increase as more components are added. (If all of the principal components are retained the MLR solution is obtained.) The final regression model is then calculated using the entire data set and the number of principal components (or latent variables) determined from the cross-validation. A typical PRESS plot is shown in Figure 2.1. In the example plot the regression model calculated with 5 principal components has the minimum prediction error.

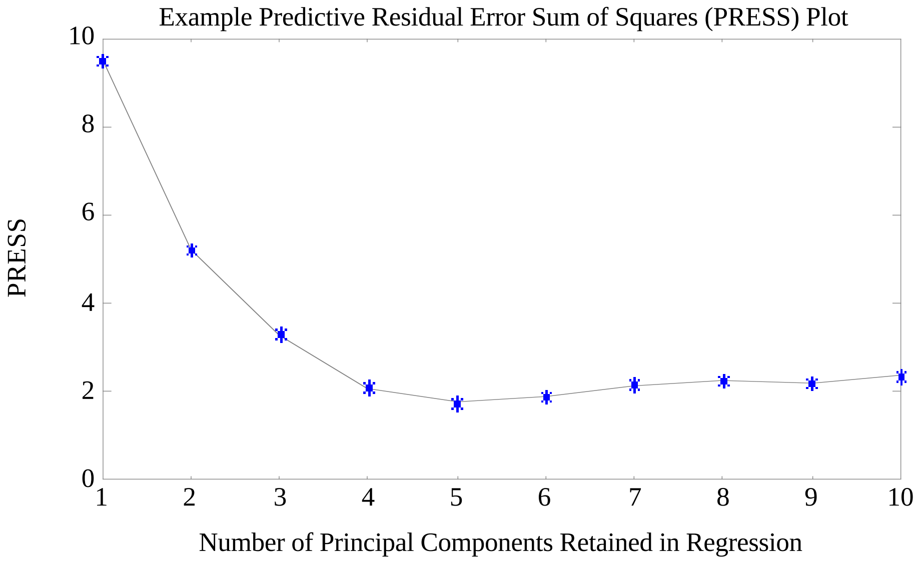

Figure 2.1. Example of Typical PRESS Plot.

2.3 Partial Least Squares (PLS) Regression

PLS regression methods are well described by Hoskuldsson (1988) and the history of PLS is covered quite well by Geladi (1988). A theoretical foundation for PLS is provided in the reference by Lorber et. al. (1987). In the paragraphs that follow a brief description of the method is presented and the computational steps are outlined.

2.3.1 Motivation for PLS

PLS is a multivariate calibration technique where a data matrix of inputs, known as the **X** or independent block, can be calibrated to a matrix of one or more outputs, the **Y** or dependent block (Lorber et. al 1987, Geladi 1988). PLS can be thought of as a simultaneous decomposition of the **X** and **Y** blocks using PCA. In PLS, however, the eigenvectors are rotated in each of the blocks so that the samples have similar highly correlated scores. To put it another way, the projections of the independent variables onto the first "rotated eigenvector" of the **X** block will be highly correlated to the projections of the dependent variables onto the first "rotated eigenvector" of the **Y** block and so on.

2.3.2 The PLS Method

Mathematically, the PLS algorithm exchanges the scores between the **X** and **Y** blocks as the matrix decomposition proceeds, resulting in highly correlated "eigenvectors" (latent variables). The algorithm is shown in equations (2.36) through (2.50) below.

$$\mathbf{u}_{\text{start}} = \text{some } \mathbf{y}_{\text{j}} \tag{2.36}$$

$$\mathbf{w}^{\mathrm{T}} = \mathbf{u}^{\mathrm{T}}\mathbf{X}/\mathbf{u}^{\mathrm{T}}\mathbf{u} \tag{2.37}$$

$$\mathbf{w}^{\mathrm{T}}_{\text{new}} = \mathbf{w}^{\mathrm{T}}_{\text{old}} / \|\mathbf{w}^{\mathrm{T}}_{\text{old}}\| \tag{2.38}$$

$$\mathbf{t} = \mathbf{X}\mathbf{w}/\mathbf{w}^{\mathrm{T}}\mathbf{w} \tag{2.39}$$

$$\mathbf{q}^{\mathrm{T}} = \mathbf{t}^{\mathrm{T}}\mathbf{Y}/\mathbf{t}^{\mathrm{T}}\mathbf{t} \tag{2.40}$$

$$\mathbf{q}^{\mathrm{T}}_{\text{new}} = \mathbf{q}^{\mathrm{T}}_{\text{old}} / \|\mathbf{q}^{\mathrm{T}}_{\text{old}}\| \tag{2.41}$$

$$\mathbf{u} = \mathbf{Y}\mathbf{q}/\mathbf{q}^{\mathrm{T}}\mathbf{q} \tag{2.42}$$

At this point the convergence of **t** can be checked. If it hasn't changed go to equation

(2.44), else go to equation (2.37). If the **Y** block has only one variable, then

$$\mathbf{q} = q = 1 \tag{2.43}$$

and the steps in equations (2.39) through (2.42) can be omitted and no more iteration is necessary, (the PLS method is non-iterative if there is only one **Y** block variable). Continue the procedure by performing the renormalizations:

$$\mathbf{p}^{T} = \mathbf{t}^{T}\mathbf{X}/\mathbf{t}^{T}\mathbf{t} \tag{2.44}$$

$$\mathbf{p}^{T}_{new} = \mathbf{p}^{T}_{old} / ||\mathbf{p}^{T}_{old}|| \tag{2.45}$$

$$\mathbf{t}_{new} = \mathbf{t}_{old}\, ||\mathbf{p}^{T}_{old}|| \tag{2.46}$$

$$\mathbf{w}^{T}_{new} = \mathbf{w}^{T}_{old}\, ||\mathbf{p}^{T}_{old}|| \tag{2.47}$$

After each latent variable is calculated then the corresponding value of b_i, (which relates the scores on the $i^{\underline{th}}$ **X** block latent variable $\mathbf{t}_i$ to the scores on the ith **Y** block latent variable $\mathbf{u}_i$) must be determined from

$$b_i = \mathbf{u}_i^{T}\mathbf{t}_i/\mathbf{t}_i^{T}\mathbf{t}_i \tag{2.48}$$

Once the b_i is determined then the residuals may be calculated in preparation for calculation of another latent variable as follows:

$$\mathbf{E}_i = \mathbf{E}_{i-1} - \mathbf{t}_i\mathbf{p}^{T}_i \text{ where } \mathbf{X} = \mathbf{E}_0 \tag{2.49}$$

$$\mathbf{F}_i = \mathbf{F}_{i-1} - \mathbf{b}_i\mathbf{t}_i\mathbf{q}^{T}_i \text{ where } \mathbf{Y} = \mathbf{F}_0 \tag{2.50}$$

At this point another latent variable can be calculated by returning to equation (2.36), and replacing **X** and **Y** by their residuals $\mathbf{E}_i$ and $\mathbf{F}_i$, respectively.

When PLS is used for prediction the independent block is decomposed while the dependent block is built up. The **X** block scores $\mathbf{t}_i$ are estimated by multiplying **X** by the weights ($\mathbf{w}_i$) as follows:

$$\mathbf{t}_i = \mathbf{E}_{i-1}\mathbf{w}_i \tag{2.51}$$

$$\mathbf{E}_i = \mathbf{E}_{i-1} - \mathbf{t}_i\mathbf{p}_i^T \tag{2.52}$$

where $\mathbf{E}_0 = \mathbf{X}$ (as above) and the estimated **Y** block is built up as

$$\hat{\mathbf{Y}} = \sum_{i=1}^{k} b_i\mathbf{t}_i\mathbf{q}_i^T \tag{2.53}$$

where k is the number of latent variables to be used in the prediction model.

PLS can be contrasted with Multiple Linear Regression (MLR) by noting that MLR is a special case of PLS, *i.e.*, MLR is equivalent to using all the latent variables in PLS. In MLR the vector of coefficients $\mathbf{b}_{i,mlr}$ is estimated for each of the $\mathbf{y}_i$ in **Y** as

$$\mathbf{b}_{i,mlr} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}_i \tag{2.54}$$

thus the estimated value of **Y** is

$$\hat{\mathbf{Y}} = \mathbf{XB} \tag{2.55}$$

where **B** is composed of the column vectors calculated in (2.54). While MLR generally gives a better fit to the calibration data, (because it uses all the variation in the **X** block and has a larger number of degrees of freedom), PLS often gives better prediction because it uses only the predictive information.

The parameters used in PLS prediction can also be reduced to a single linear equation, similar to that of (2.55):

$$\hat{\mathbf{Y}} = \mathbf{X}\mathbf{C} \tag{2.56}$$

where **C** is a matrix in the general case and a vector in the case of only one variable in the **Y** block. This is done by substituting equation (2.52) into (2.51) and substituting the result into (2.53). The result is equation (2.57) where k is again the number of latent variables to be used in the prediction (as above) and it is assumed that the value of the term in brackets is equal to **I** for the case of i = 1.

$$\hat{\mathbf{Y}} = \mathbf{X}\sum_{i=1}^{k} b_i \left[\prod_{j=1}^{i-1}\left(\mathbf{I} - \mathbf{w}_j\mathbf{p}_j^T\right)\right]\mathbf{w}_i\mathbf{q}_i^T \tag{2.57}$$

It is proposed that PLS can also be used to determine the general state of the process in a fashion similar to the use of the Q statistics associated with PCA models. This requires that PLS models be obtained that relate each variable to the remaining variables in the system. Thus for a system with n variables, n PLS models would be required. Fortunately, using the relationship given in equation (2.57) the n PLS models can be formed into a single matrix, with each model being a column vector. Because each of the variables does not contribute to its own prediction, the resulting prediction matrix, $\mathbf{M}_p$, has zeros on the diagonal. Thus the PLS prediction $\hat{\mathbf{X}}$ of a data matrix **X** can be obtained by simple matrix multiplication

$$\hat{\mathbf{X}} = \mathbf{X}\,\mathbf{M}_p \tag{2.58}$$

A residuals matrix, $\mathbf{R}_{pls}$, can be calculated from

$$\mathbf{R}_{pls} = \mathbf{X} - \hat{\mathbf{X}} = \mathbf{X} - \mathbf{X}\,\mathbf{M}_p = \mathbf{X}(\mathbf{I} - \mathbf{M}_p) \tag{2.59}$$

The similarity between equation (2.59) and the calculation of the PCA residuals in equation

(2.10) should be readily apparent.

The residuals matrix $\mathbf{R}_{\mathrm{pls}}$ can be used in much the same manner as the PCA residuals matrix $\mathbf{R}$ for determination of the overall state of the process (as in calculation of Q) or for determining the failure of specific sensors. In the latter case PLS predictions for each variable can then be compared with the actual values, and the off-normal variables can be identified. As noted previously, this approach is the generally accepted method for process fault detection. A model is produced which predicts the value of a process variable from other process variables (or process inputs) and the difference is monitored.

In contrast to least-squares methods, even if the calibration data consists of deterministic variation and white noise, the PLS model residuals are not expected to be mean-zero and white (see for instance Lorber (1987) or Hoskuldsson (1988)). PLS provides biased estimates. Therefore, the methods used to monitor the residuals for changes must be adjusted accordingly.

It should be noted that the idea of using PLS in this fashion is related to the general idea of prediction of process outputs using secondary measurements, such as the example provided by Mejdell and Skogestad (1989).

2.3.3 PLS and PCR with Non-Linear Inner Relationships

As presented in the previous sections, PLS and PCR are linear techniques. PLS and PCR can be used with nonlinear data, however, in one of two ways. First, it may be possible to transform the data to a linear (or approximately linear) form. Such transforms are generally arrived at through theoretical considerations[3]. On the other hand, both the PLS and PCR methods can be transformed into non-linear methods by changing the "inner relationship" between the $\mathbf{X}$ and $\mathbf{Y}$ blocks. This PLS this is done by substituting a non-

---

[3] A simple example of this would be modelling the relationship between the height of liquid in a tank and the outlet flow rate. Flow through a constriction is proportional to the square root of the pressure drop, which is proportional to liquid height. Therefore, a relationship would be proposed between the outlet flow and the square root of the surface height.

linear relationship, such as a polynomial fit, for equation (2.37), while in PCR this is done by proposing a non-linear relationship instead of equation (2.22) The exact form of the relationship can come either from theory or can be suggested by a plot of the **X** versus **Y** block scores (in PLS the $\mathbf{t}_i$ and $\mathbf{u}_i$, respectively).

2.4 Continuum Regression

The regression techniques of MLR, PCR and PLS can all be unified under one approach which will be referred to here as continuum regression. The basic idea behind continuum regression has been discussed among chemometricians for some time, as was pointed out in the article by Lorber et. al. (1987). The descriptive name "continuum" comes from a paper by Stone et. al. (1990). The algorithm used here is slightly different than either of the approaches of Lorber and Stone but the result is the same.

2.4.1 Motivation

It was the original intention of this work was to investigate 3 regression techniques for the identification of process models: Multiple Linear Regression (MLR), Partial Least Squares (PLS) and Principal Components Regression (PCR). The problem with this approach is that it presents a rather fragmented view of the identification picture: 3 isolated techniques. Continuum regression provides a way to unify the 3 methods: PCR, PLS and MLR can all be shown to be special cases of continuum regression. Furthermore, considering the methods collectively should provide additional insight into the how the methods relate to each other. An additional incentive was provided by the results of Lorber et. al. (1987), which showed that for some calibration problems techniques that lay in the continuum between PLS and MLR gave optimal regression models.

A central idea behind continuum regression is that the PLS method captures covariance between the input and output blocks. This can be thought of as an attempt to balance the two tasks of providing a reduced order description of the input data block and correlating the input data to the output data. On the extremes of this trade off are PCR, which starts with a model that describes variance in the input block and correlates it piece by piece to the output block, and MLR, which seeks only to correlate the input and output blocks without regard to the input block structure. The conventional PLS method tries to do both and thus occupies some middle ground between PCR and MLR. The relationship between CR and the individual techniques of MLR, PLS and PCR are shown graphically in Figure 2.2. Stone refers to MLR, PLS and PCR methods as "canonical correlation", "canonical covariance" and "canonical variance" methods, respectively. This balance between describing variance and capturing correlation can be changed continuously, however, by several algorithms.

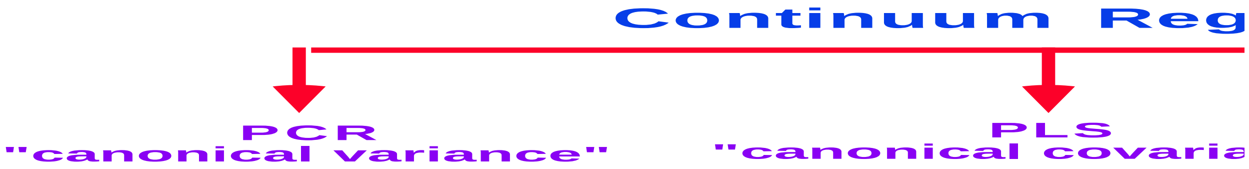

Figure 2.2. Relationship Between Continuum Regression and PCR, PLS and MLR.

In this work a continuum regression method is chosen that is simple to program but computationally inefficient. The mathematical details of this algorithm are given in the next section. In this routine a singular value decomposition (SVD) of the input data block is calculated and the singular values are taken to the desired power, with zero corresponding to MLR, infinity to PCR and 1.0 to conventional PLS. The input block is re-formed using the modified singular values and then a conventional PLS routine is used to obtain a regression model. The PLS regression vectors are formed then rescaled using the original SVD matrices and the modified singular values. This is something of a brute force approach but the algorithm converges to the PCR and MLR solutions for large and small powers of the singular values, respectively, as it should.

Now, instead of looking at the 3 regression techniques separately, it is possible to look at a continuum of techniques and see how the prediction properties change as the number of latent variables and continuum parameter (power to which the singular values are taken) are varied. Thus, a 3-dimensional predictive residual error sum of squares (PRESS) surface can be constructed for any given identification problem. A search can be made over the error surface to find the optimum model for prediction.

2.4.2 Continuum Regression Method

In the version of the CR method used in this work, the first step is to perform a singular value decomposition on the independent variable block

$$\mathbf{X} = \mathbf{U}\ \mathbf{S}\ \mathbf{V}^{\mathrm{T}} \tag{2.60}$$

The matrix $\mathbf{S}^{\mathrm{m}}$ is formed by taking each of the diagonal entries of **S** (the singular values) to the desired power, m. The new **X** block, defined as $\mathbf{X}^{\mathrm{m}}$, is then formed as follows:

$$\mathbf{X}^{\mathrm{m}} = \mathbf{U}\ \mathbf{S}^{\mathrm{m}}\ \mathbf{V}^{\mathrm{T}} \tag{2.61}$$

The PLS algorithm described earlier can now be used along with equation (2.57) for the regression vector calculation to produce a regression vector for any number of latent variables desired. The regression vector obtained is not properly scaled because of the rescaled **X** block. Any regression vector, **r**, can be rescaled, however, by projecting it onto the SVD basis set **V** and multiplying by the ratios of the singular values used in (2.61) to the original singular values calculated in (2.60). Thus:

$$\mathbf{r}_{\mathrm{scl}} = \mathbf{r}\ \mathbf{V}\ \mathbf{S}_{\mathrm{scl}}\ \mathbf{V}^{\mathrm{T}} \tag{2.62}$$

where

$$\mathbf{S}_{\mathrm{scl}} = \mathbf{S}^{\mathrm{m}}./\mathbf{S}; \tag{2.63}$$

where the symbol "./" indicates term by term division of the elements of the singular value matrices.

A convenient feature of this particular continuum regression algorithm is that it is just a new "shell" written around the PLS routine, and is therefore quite easy to program. It is probable, however, that other algorithms are more efficient computationally. Another nice feature of this CR algorithm is that it easy to understand how it works. Knowing that PLS captures covariance, (*i.e.*, tries to strike a balance between capturing **X** block variance and obtaining a correlation with the **Y** block), it is easy to see what the effect of CR would be. When the singular values are taken to large positive powers, the **X** block becomes progressively more directional. The PLS model gets progressively more biased towards the major eigenvectors (eigenvectors associated with large eigenvalues) because any rotation towards minor eigenvectors results in a rapid decrease in the PLS objective function. Thus the PLS latent variables begin to look more like the PCR loadings vectors. On the other hand, when the singular values are taken to very small powers, the **X** block becomes progressively less directional. Any rotation of the PLS latent vectors captures approximately the same amount of variance so the algorithm tends to find the best correlation. The result is that the solution begins to look very much like the MLR solution.

In PCR and PLS a search is performed which uses cross validation to determine the number of latent variables that minimizes the model error (PRESS). In CR a search is made over two variables: the number of latent variables and the continuum parameter. It is possible to view the PRESS as a function of the number of LVs and the continuum parameter as shown in Figure 2.3. In the figure the height of the surface represents the error for the model (PRESS) corresponding to the given number of latent variables and power of the singular values. The location of the PCR, PLS and MLR models is shown. In the figure the power of the singular values has been varied from 8 (next to PCR) to 1/8 (next to MLR) in logarithmically spaced intervals. In reality there is only one MLR model, not 15 as suggested by the figure. However, the error associated with this model has been repeated down the back right side of the figure for continuity.

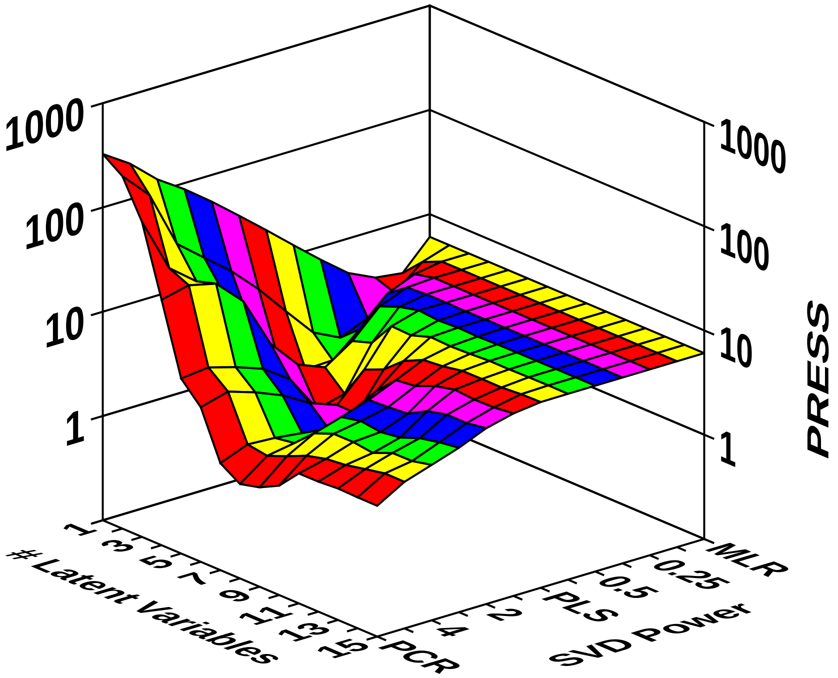

Figure 2.3 Continuum Regression PRESS Surface.

Figure 2.3 illustrates some of the features that are common to most CR PRESS surfaces. The level surface to the right in the figure, the "MLR plain", represents models identified with so many latent variables that they have converged to the MLR solution and, therefore, have the same model error. As mentioned previously, all PLS and PCR techniques converge to the MLR solution as latent variables are added. The more correlation (as opposed to variance) is factored in the regression, the faster the convergence. On the left of the figure are models with large error, the "PCR mountain", identified with too few latent variables adequately describe the "true" regression vector. In between is a "valley of best models" that have minimum PRESS. A search is performed to find is the model that represents the "bottom" of the "valley", *i.e.*, the model with the minimum prediction error.

2.5 Data Pretreatment and Scaling

Before completely leaving the topics of PCA and the CR methods, a word about preprocessing of data is in order. In particular, scaling of variables is very important to PCA, CR and other eigenvalue analysis type methods. Because the techniques discussed here are linear, (with the exception of PLS with non-linear inner relationship), linearization of the data can also be very important. Finally, eliminating outliers in the calibration set is also an important step in obtaining the optimally predictive model. Scaling, outliers analysis and linearization all need to be done before any modeling. Collectively these techniques are called preprocessing. Each of them will be discussed in turn in the following sections.

2.5.1 Types of Scaling

Scaling of variables is very important to methods such as PCA and CR because eigenvectors will tend to be biased towards variables with larger numerical values. This is because numerically larger variables appear to be associated with greater amounts of variance (larger eigenvalues). For this reason, it is customary to introduce some type of scaling that assures that the variables will have approximately equal weights in the regression.

The two most common types of scaling are mean centering and autoscaling. They are typically used in different situations. In mean centering, the mean of each data column is subtracted off, leaving a data matrix where the mean of each variable is zero. Autoscaling is mean centering plus variance scaling. After mean centering each variable is divided by the original standard deviation. The result is a matrix where all the columns have mean zero and unit variance. This leaves only information about the correlation between the variables. As mentioned previously, the covariance matrix of an autoscaled data set is known as the correlation matrix. There are ones on the diagonal (each variable is perfectly correlated with itself) and numbers between -1 and 1 everywhere else.

Mean centering is very common with data set where every variable is in the same units and/or have similar noise characteristics. An example of this is spectroscopy, where all the variables are in units of absorbance and the noise level for each variable would be expected to be the same. Variables (wavelengths) with a larger absolute variance will tend to be weighted more heavily in the PCA or CR models but these variables will also tend to contain relatively more information concerning any variation in the system. Thus the numerical bias from the scaling actually can serve a useful purpose.

Autoscaling is often used when the variables in a data set have different units and possibly widely different noise characteristics. Many chemical process systems are good examples of this. The variables can include a wide variety of measurements such as temperature (degrees C), power input (watts), fluid concentrations (moles/l) and tank levels (centimeters). The use of this technique implies that each of the variables are of equal importance over their range in the calibration data. Given no other information, this is generally a good starting point.

Other scalings, where the modeler chooses the scaling factors for each variable individually, are also used. As an example, if some variables are thought to be inherently more important based on physical reasons, they can be weighted more heavily. If the data set appears to be atypical, in that some variables would normally vary more or less than the observed variance in the sample set, scaling the variables to "percent of full scale" might also be a good choice.

In some situations it is not easy to decide what scaling is best, and it may come down to comparing the results of modeling with different scalings. The effect of scaling on PCA modeling results is considered further in Appendix I. The effect of scaling on the fault detection power of PCA models is considered in Chapters 3 and 4.

2.5.2 Outliers Analysis

Outliers, samples that are inconsistent with the majority of the data set, can also be a source of error when developing regression models. The modeler should try to remove any such bad data before building PCA or CR models. This is important because outliers have a great deal of "leverage" on the data or regression models and can change them significantly. This, of course, will result in misleading models.

There are several methods which may be used to screen data for outliers. In the simplest case, this can be done by simply "eyeballing" the data. Often, outliers are quite obvious. PCA can be used on data sets, before any final models are formed, to detect outliers. Typically, outliers will have either very large Q values or there will be principal components where only one or a few samples will have a very large score. Finally, there are also more formal statistical methods for detecting outliers, such as those given in the paper by Lorber (1989).

2.5.3 Linearization

As mentioned in section 2.3.3, the PCA and CR techniques are linear, and as such, will not provide good models of highly non-linear data. If the functionality of the non-linearities is known, it is best to pretreat these variables so that the resulting variables are linear. Sometimes a linearizing function can be determined from theoretical considerations. In other instances, a linearizing function may be arrived at by inspection. In other instances an approximately linearizing function may be found through an iterative search, such as the approach demonstrated by Parazoglu (1990). In any case, linearization may greatly improve calibration models. It is demonstrated in Chapter 5, however, that in some instances non-linear models will have better prediction abilities than linear models based on linearized data.

2.6 Statistical Process Control

An excellent review of the most widely used techniques in SPC is provided in the reference by Jackson (1976). Many texts are also available on the subject, including the recent one by Wadsworth et. al. (1986). This section gives the a brief introduction to the field of SPC. It is not intended to be comprehensive. For further information the references above should be consulted.

SPC got its start in the 1930s when Walter Shewart developed the first control chart (Shewart 1931). This chart was formed by plotting the deviation from the normal mean of a process variable. The chart also contained two control limit lines that were placed three standard deviations from the mean. Since the probability that any sample from the parent distribution would be outside the limits (given that the distribution is normal) is only .003, it was assumed that any sample falling outside the limits indicated a change in the process mean. The sensitivity of the chart to changes in the process mean could be increased by plotting the average of groups of samples. This also has the benefit of strengthening the normality assumption, since the distribution of averages is normal regardless of the parent distribution. This type of control chart, commonly called a Shewart Chart, is still in wide usage today. Shewart charts can also be constructed for the range and variance for the sample sets.

There are many variations on Shewart charts, but they are all time independent. By time independent it is meant that the samples are assumed independent of one another and no information from past samples is considered. There are many time dependent chart techniques, however, that do consider past information. Two of the most common are Cumulative Sum (Cusum) and Exponentially Weighted Moving Average (EWMA) charts.

A cusum chart is formed by calculating the cumulative sum of the deviations from the process mean and plotting this sum. If a process experiences a shift in mean the cusum will start to drift rapidly. This generally provides a more sensitive indicator of change in mean than a Shewart chart. Cusum type charts generally incorporate a V-mask to specify the operating limits, as shown in Figure 2.4. The control limits are specified by the lead distance d and the angle Θ. In the figure the lead distance is 5 and $\tan(\Theta)$ is .40. The actual values of the limits are generally determined from Average Run Length (ARL) specifications. The ARL is defined as the average number of observations required to detect a specified change in the process mean. If any observations are found outside the V-mask, such as observation 18 in the figure, the process is deemed out-of-control and in need of adjustment. The V-mask is a graphical approach, but if an algorithmic approach is desired the lead distance and mask angle can be used to calculate the parameters in a numerical algorithm which performs the same function as the V-mask.

In EWMA charts the current point is the weighted average Z_t of all previous samples in the process run:

$$Z_t = \gamma X_t + (1-\gamma)Z_{t-1} \qquad (2.64)$$

where γ is the weighting coefficient ($0<\gamma<1$) and X_t is the current value of the process variable being monitored. Anyone with experience in digital filtering will recognize this is very similar to a first order filter with time constant proportional to $\ln(1/\gamma)$. Control limits on EWMA charts are generally multiples of $(\gamma/(2-\gamma))^{1/2}$ times the standard deviation of the calibration data. Typically the value of γ is about .2. If the value of the weighting parameter was set to $\gamma = 1$ the resulting chart would be equivalent to a Shewart chart.

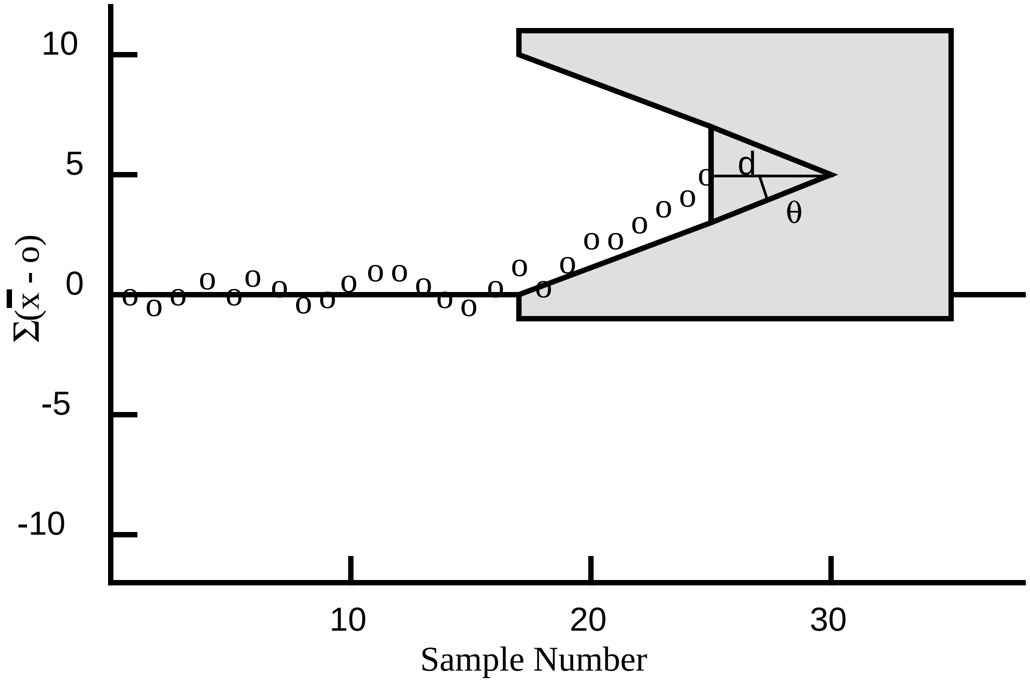

Figure 2.4. Cumulative Sum Chart.

Multivariate adaptations of SPC techniques, or MSPC, are still relatively new, with the work of Jackson (1979, 1980, 1981A, 1981B) being the major exception. In fact, a review article in the Journal of Quality Technology (Gibra, 1975) makes no mention of multivariate analysis in SPC with the exception of some work directly related to that of Jackson. Jackson used PCA to form multivariate versions Shewart charts. Single variables were replaced with PCA scores, residuals (Q) and sum of scores (T^2) values as indicated in section 2.1.

By 1980, more publications concerning multivariate adaptations of SPC were beginning to appear (Vance, 1983). The first article concerning a truly multivariate adaptation of cusum charts was published by Woodall and Ncube (1985) and has led to two more (Healy 1987 and Crosier 1988). Applications of PCA have appeared recently in Wise and McMakin (1988), Wise et. al. (1988), Wise and Ricker (1989), Wise, Ricker and Veltkamp (1989), MacGregor (1989), MacGregor (1990) and Kresta, MacGregor and Marlin (1990).

The problem of correlated observations is also beginning to be addressed by the SPC literature. The most common procedure is to propose a model of the Autorecursive Moving Average (ARMA) type described by Box and Jenkins (1970). Under appropriate conditions the residuals of the ARMA model can be modelled as white noise and control limits can then be set in a manner similar to that shown for PCA residuals in section 2.1.3. This is the approach in the paper by Alt et. al. (1977) and the approach of Harris (1989).

As a final note in this section, the question may arise concerning the interface between SPC and the type of process control generally studied by engineers. MacGregor (1988) has considered this topic extensively and has concluded that there is a very important area of overlap between these two disciplines. That area is on-line quality control. SPC and process control are simply two ways of approaching the same problem. The problem is that the personnel in each of the disciplines does not fully understand the techniques used by the other. MacGregor points out that there is much that the process control engineer can learn from SPC methods. "Aside from providing a procedure to decide on when to apply control actions to a process, SPC charts are invaluable as diagnostic tools. They highlight the periods where process upsets have occurred, and by analyzing the process data in these periods one can often pinpoint the cause of the disturbances and perhaps eliminate or minimize such disturbances in the future. Of course, this is where real process improvement is made."

2.7 Process Models

There are many types of process models in use today. In the following sections a few of these models that are pertinent to this work will be reviewed. This includes state-space, Finite Impulse Response (FIR) and Auto-Regressive eXtensive variable (ARX) models.

2.7.1 State-Space Models

State space models are well described in several texts on process control. This includes the works by Sage and White (1977) and Kwakernaak and Sivan (1972) which emphasize continuous time systems, and that of Åstrom and Wittenmark (1984) which emphasizes discrete systems. A brief review of the state space model format follows.

State-space models are characterized by a set of state variables which capture the state, or essential, information of the process system, and a set of measurement variables, which correspond to the actual measurements from the process. The discrete form of the state space model is

$$\mathbf{x}(k+1) = \Phi\mathbf{x}(k) + \Gamma\mathbf{u}(k) + \nu(k) \qquad (2.65)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) + \mathbf{e}(k) \qquad (2.66)$$

Equation (2.65) is commonly referred to as the state equation, while equation (2.66) is the measurement equation. Assuming that the process has n states, r inputs and p measurements, then in the state equation, $\mathbf{x}(k)$ is the (n by 1) vector of state variables at time k, $\mathbf{u}(k)$ is the (r by 1) vector of process inputs at time k, Φ is the (n by n) state transition matrix which determines the effect of the states at time k on the states at time k + 1, Γ is the (n by r) input matrix which determines the effect of the inputs at time k on the states at time k + 1 and $\nu(k)$ is the (n by 1) vector of state disturbances at time k. In the measurement equation, **y**(k) is the (p by 1) vector of process measurements at time k, **C** is the (p by n) measurement matrix which describes how the states relate to the measured outputs, **D** is the (p by r) feed through matrix, which describes the direct effect of the process inputs on the measured outputs and **e**(k) is the vector of measurement noise at time (k). For most processes **D** is zero; process inputs rarely have an instantaneous effect on the process outputs. The dimension of the state space can be greater than, less than or equal to the dimension of the measurement space.

The state space model format is convenient for several reasons which will be important in later sections. The model formulations allows for flexibility concerning the number of measurements and process states. For instance, the number of states can be greater or less than the number of measurements, provided that the matrices in equations (2.65) and (2.66) are sized correctly. The case where the process measurements are identical to the process states is also handled easily by making **C** equal to the identity matrix. The model form is also has the additional feature that it distinguishes between state disturbances and measurement noise.

State-space models grew out of the solutions to sets of differential equations which describe the way particular systems operate. In some state space models, for instance, some of the states will be derivatives of other states. As such, many state space models are derived directly from theory. The primary interest in state-space models from the point of view of this document involves the relationship between the state-space and PCA approach to process modeling. This issue will be considered further in Chapter 3.

2.7.2 Finite Impulse Response (FIR) Models

There is a renewed interest in FIR models of late, primarily because several popular model-based controller design methods depend on FIR process models (Prett et. al. 1989). In an FIR model the process output is considered to be a function of past values of the process input only:

$$y(k) = b_0u(k) + b_1u(k-1) + b_2u(k-2) + ... + b_n(k-n) \qquad (2.67)$$

The model of equation (2.67) includes a feed-through term (b_0), which is equivalent to a non-zero **D** matrix in the state-space format. If the model of the system included time delays, some additional initial terms would be zero, up to the number of sample periods of the delay. This relationship of (2.67) is perhaps more conveniently expressed in shift operator form. Starting from the definition of the forward shift operator q

$$q(x(k)) = x(k+1) \qquad (2.68)$$

then the backward shift operator q^{-1} is defined as

$$q^{-1}(x(k)) = x(k-1) \qquad (2.69)$$

The relationship of equation (2.67) can be rewritten as a polynomial in q^{-1}, B(q). (Note that the superscript -1 will be dropped from the notation for B polynomials.)

$$y(k) = B(q)u(k) \qquad (2.70)$$

where the polynomial B(q) is of the form

$$B(q) = b_0q^0 + b_1q^{-1} + b_2q^{-2} + ... + b_nq^{-n} \qquad (2.71)$$

As mentioned above, for a system with time delays, some of the initial terms in the B polynomial may be zero.

The FIR models shown above are for single input/single output (SISO) systems. The FIR model form can, however, include B polynomials for multiple inputs. This allows for modeling of multiple input/single output (MISO) systems. If multiple input/multiple output models are required it is common to use a collection of MISO models.

A typical FIR model might have 25-100 coefficients, *i.e.*, the order of the polynomial B(q) might be 25 - 100[4]. FIR models are often referred to as non-parametric because they make no assumption concerning the underlying mechanistic principles of the true process, with the exception that the FIR model form can fit only processes that are asymptotically stable[5]. An FIR model is thus not a fit of parameters to a predetermined form. In order for an FIR model to be accurate, however, enough terms must be included to cover the time response of the system in question. For instance, if the system to be modelled takes 50 sample periods to come to steady state, the order of B(q) must be ~50 in order to obtain an accurate model. It is also possible to change the sample rate in order to adjust the number of FIR terms needed to describe the process.

2.7.3 Auto-Regressive Extensive Variable (ARX) Models

In contrast to FIR models, in ARX models the current output is considered to be a function of both past values of the input and output. Thus the ARX model takes the form

$$y(k) = a_1y(k-1) + ... + a_my(k-m) + b_1u(k-1) + ... + b_n(k-n) \qquad (2.72)$$

This is more conveniently expressed in shift operator form as

$$A(q)y(k) = B(q)u(k - d) \qquad (2.73)$$

Here the d term has been added to indicate a possible pure delay of d time units.

Unlike FIR models, the ARX model is a parametric form. Once the orders of the A and B polynomials are fixed, there is a limited number of plant behaviors which can be modelled. For instance, if the order of A is taken to be 1, and the order of B is taken as 0,

---

[4] See, for instance, the application of FIR models to an anerobic wastewater treatment process in Ricker (1988).

[5] An exception to this is the FIR form proposed by Morari (1990) that can model processes with pure integrators.

the resulting (stable) processes must either rise exponentially to the steady state value or overshoot it at the first step and approach it with an exponentially decaying oscillation.

For a SISO or MISO process, the ARX form given in (2.73) can be shown to be equivalent to the state space form of equations (2.65) and (2.66). The transformation between model forms is not unique, however. There are, in fact, an infinite number of state-space models which have the same input/output behavior as a given ARX system.

2.8 Common Process Model Identification Methods

In this section some common methods for identifying FIR and ARX models will be considered. The methods outlined here will be used as the "yardstick" for assessing the relative success or failure of the CR method.

2.8.1 FIR Model Identification

As mentioned in the introductory chapter, there is little information available concerning the identification of FIR models. It appears that most FIR models are identified directly from step or pulse tests and smoothed based on engineering judgement. However, it is difficult to do a comparison with this as the reference method. Instead, MLR will be the standard for identification of FIR models.

Consider the FIR model of equation (2.56). Let Θ be the vector of b parameters that must be estimated (following the notation in Ljung, 1987), that is:

$$\Theta = [b_0 \ \ b_1 \ \ b_2 \ \ \ldots \ \ b_n] \tag{2.74}$$

Assume a data set has been obtained from the system to be modelled:

$$\mathbf{Z}_N = [y(1)\ u(1)\ y(2)\ u(2)\ \ldots\ y(N)\ u(N)] \tag{2.75}$$

Before MLR can be used to estimate Θ, the input/output data must be rewritten into a matrix which is consistent with the proposed model. In this case the matrix of inputs is:

$$\phi(t) = [u(t) \;\; u(t-1) \; \ldots \; u(t-n)] \qquad (2.76)$$

where $\phi(t)$ is the $t^{\underline{th}}$ (1 by n) row vector of the ϕ matrix. The MLR estimate of Θ can now be calculated from the normal equations:

$$\hat{\Theta} = (\phi^T\phi)^{-1}\phi^T\mathbf{y} \qquad (2.77)$$

Estimates of the output variable are obtained from:

$$\hat{\mathbf{y}} = \phi\,\hat{\Theta} \qquad (2.78)$$

Estimates of Θ obtained this way, however, can be very inaccurate, depending upon the condition of ϕ. If ϕ is nearly rank deficient, i. e. if the matrix $\phi^T\phi$ has some very small eigenvalues, the resulting Θ can be corrupted by noise. Use of this method often leads to "ringing" in the coefficients of Θ: the coefficients will be alternately high and low relative to the true coefficients.

When identifying systems with a delay, an estimate of the delay is usually obtained prior to forming the final model and the parameter vector and ϕ matrix are formed accordingly. If a delay exists and it is not properly accounted for, then some of the coefficients in Θ which are to be estimated are actually zero. In this case, it is likely that in the estimation procedure these coefficients will deviate substantially from zero because of spurious correlation with the process output. The resulting model will then be less accurate than it would have been had the coefficients been set to zero to begin with. In Chapter 5 it will be shown that incorrect estimation of the time delay has a greater impact on model accuracy when the CR method is used for identification instead of MLR.

Estimates of the the process time delay can be obtained by cross-correlation of the input/output data (see for instance Ljung, 1987). In this procedure the correlation between the process inputs and the output at later times is determined. The process time delay is estimated as the minimum time shift for which a significant correlation is obtained between input and output. It is also possible to obtain estimates of the system time delay by cross-validation of a series of FIR models with incremental changes in the assumed delay. In this method the model with the best predictive ability would be assumed to have the correct system delay.

2.8.2 MLR and Instrumental Variables for ARX Models

ARX models are typically identified either by MLR or the method of instrumental variables (IV method). Identification by MLR follows the general procedure outlined above with appropriate changes in the ϕ and Θ matrices.

Consider the ARX model of equation (2.72). From this it can be seen that the parameter vector will be of the form:

$$\Theta = [a_1 \ \dots \ a_m \ b_0 \ b_1 \ \dots \ b_n] \qquad (2.79)$$

The matrix of inputs and lagged outputs will be

$$\phi(t) = [y(t-1) \ \dots \ y(t-m) \ u(t) \ \ u(t-1) \ \dots \ u(t-n)] \qquad (2.80)$$

The MLR estimate of Q can now be obtained as in (2.72), and prediction will follow as shown in (2.73).

It should be noted that the normal MLR equation for estimation of the parameters in ARX models is typically much better conditioned than the corresponding regression for FIR models. This is, in part, due to the smaller number of parameters typically estimated and the lack of correlation in the parameters. A particular problem in the estimation of

ARX models, however, is the presence of unmodeled disturbances. It is shown in Ljung (1987) that in these cases that the estimate of Θ will not tend towards the true value even for very large numbers of samples due to the correlation between the disturbance and ϕ. It this author's experience that the models obtained using data with unmodeled disturbances are typically biased towards the lagged outputs over the input variables, i. e. the coefficients of the A polynomial are larger at the expense of the coefficients of the B polynomial. The IV method was developed, in part, to address this concern (See Ljung, 1987).

In the IV method the lagged outputs in the ϕ matrix are replaced with the instrumental variables, creating a new matrix

$$\varsigma(t) = [\psi(t-1) \; ... \; \psi(t-m) \; u(t) \; u(t-1) \; ... \; u(t-n)] \qquad (2.81)$$

The objective of this is to remove the correlation found between the disturbance and ϕ. Whereas the MLR solution to the identification problem could be written as

$$\Theta \;\text{ such that }\; \phi^T[\mathbf{y} - \phi\Theta] = 0 \qquad (2.82)$$

The problem that must be solved for the IV approach is

$$\Theta \;\text{ such that }\; \varsigma^T[\mathbf{y} - \phi\Theta] = 0 \qquad (2.83)$$

The solution to (2.83) is obtained from

$$\hat{\Theta} = (\varsigma^T\phi)^{-1}\varsigma^T\mathbf{y} \qquad (2.84)$$

provided, of course, that the inverse of $\varsigma^T\phi$ exists.

The remaining question here, of course, is how to generate the instrumental variables. There are many ways to do this, as pointed out in Ljung (1987) and in Soderstrom and

Stoica (1983). Generally, the instruments are generated by putting the inputs through a linear filter. There are many ways to choose this filter, but a common choice is to use an existing model of the process as the filter. This is the basis of the IV four step method (IV4). The first step in the IV4 method is to obtain an ARX process model using the conventional MLR method. In the second step this model is used to generate the instruments which are then used to obtain a model estimate as in equation (2.84). In step 3 an auto-regressive (AR) model is fit to the model residuals from the previous step. In the fourth step the AR model is used to filter the instruments from the second step and the model is estimated once more using equation (2.84). It can be shown (as in Ljung 1987) that as the number of calibration samples becomes infinite, this procedure results in an unbiased model. This is a consequence of the way the method uses instruments which, (unlike ϕ) are uncorrelated with the process noise.

In this work MLR will be the reference method for ARX model identification. Some comparison to the IV4 method for ARX identification will also be made.

2.9 The Liquid-Fed Ceramic Melter

The LFCM is the reference process in the United States for solidifying the liquid wastes produced during the reprocessing of nuclear fuels (Burkholder and Jarrett 1986). A simple schematic of the LFCM operated at West Valley Nuclear Services Co. Inc. is shown in Figure 2.5. A slurry, consisting of reprocessing wastes mixed with glass forming chemicals, is fed onto the surface of the molten glass pool which is heated by passing a current between pairs of the three electrodes. Volatiles, consisting primarily of water and acids, are driven off and treated in an off-gas system which is not shown. The dried feed which remains forms a "crust" or "cold cap" which melts continuously into the glass. Glass is poured periodically from the melter through a riser section which is also not shown. This results in periodic fluctuations of the glass level.

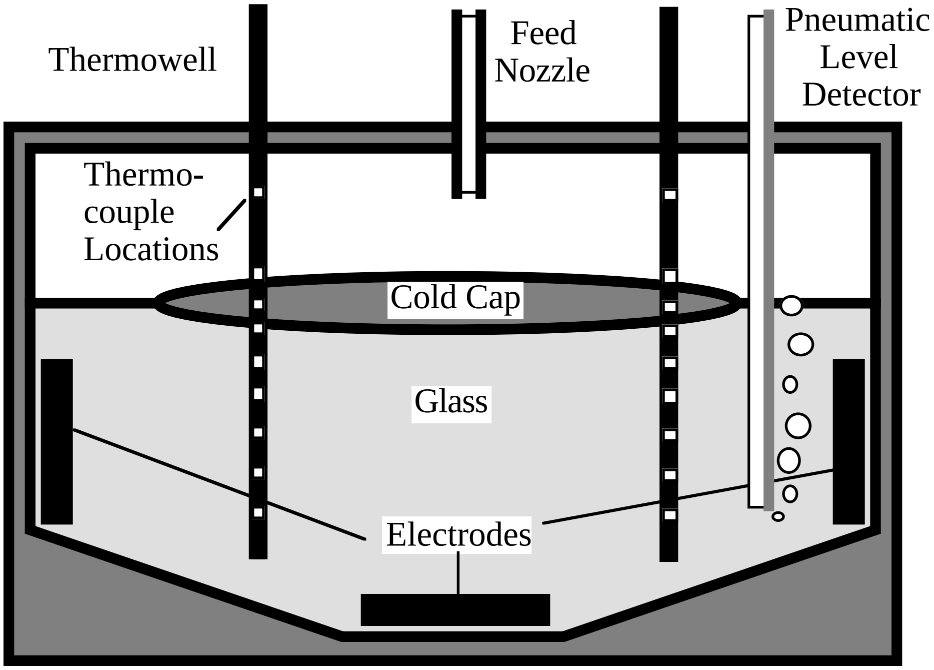

Figure 2.5. Schematic Drawing of Liquid-Fed Ceramic Melter

The melter is monitored extensively (Barnes et. al. 1985). Temperatures are monitored at 20 locations within the molten glass pool and the resistance and power dissipated between each of the electrode pairs is recorded. Data is also taken on feed flow rate and glass tank level. In all, 29 variables are recorded. The variables are keyed to their TAG ID number at West Valley in Table 2.1. As might be expected, many of the variables are highly correlated. PCA/PLS methods are more appropriate for data of this type than methods that assume statistical independence of the variables.

In Table 2.1, variables 1-10 correspond to the east side of the melter (as positioned at the West Valley site) while variables 11-20 correspond to the west side. The temperature variables are numbered sequentially starting with 1 and 11 near the bottom of the glass melt on the east and west side, respectively, and increasing up to 10 and 20 in the plenum. Variables 7-9 and 17-19 are in the cold cap region of the melt. Fluctuations in the glass level cause the cold cap to slide up and down the thermowells and can affect temperature measurements in the region. It is shown in Appendix I that glass level changes are a major source of variation in the melter data.

Table 2.1. Connection Between Variable Numbers and West Valley TAG ID

Variable Number	Description	TAG ID
1-9	Glass Temp (C)	TT--2011-TT--2019
10	Plenum Temp (C)	TT--2010
11-19	Glass Temp (C)	TT--2021-TT--2029
20	Plenum Temp (C)	TT--2020
21-23	Power (kW)	KT--2031-KT--2033
24-26	Resistance (Ω)	RT--2031-RT--2033
27-28	Feed Rate (l/h)	FT--1125/FT--2004
29	Glass Level (in.)	LIX-2001

The melter data is currently recorded at 5 minute intervals, though the process is actually sampled at a much faster rate. Though 5 minutes may seem long for some chemical processes, the LFCM is generally very slow to respond to setpoint changes or disturbances and so the 5 minute sample time is appropriate. The process time constant for temperature changes associated with power setpoint changes is on the order of hours. The glass tank residence time is on the order of days. Process data is stored on the WVNS VAX system and has been accessible at the University of Washington via modem.

3.0 Process Monitoring with PCA

In this chapter the role of PCA for process monitoring is considered in a fundamental light. Early studies, such as those in Wise and McMakin (1988), Wise et. al. (1988), Wise and Ricker (1989) and in Appendix 1[6] of this work, indicated that in heavily instrumented processes PCA appeared to capture the "essence" of the process variation. While the PCA loadings gave some information (sometimes quite ambiguous) about the correlation of variables in the process, the scores seemed to indicate the "state" of the process. This chapter answers the question of the nature of the relationship. Specifically, the relationship between PCA and the state-space model format is considered. The result is a theoretical basis for PCA monitoring.

3.1 A Theoretical Basis for PCA Monitoring

Consider once again a linear, time-invariant (LTI), discrete, state-space process model of the form:

$$\mathbf{x}(k+1) = \Phi\mathbf{x}(k) + \Gamma\mathbf{u}(k) + \nu(k) \tag{3.1}$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) + \mathbf{e}(k) \tag{3.2}$$

where $\mathbf{x}(k)$ is the (n by 1) state vector at sampling period k, $\mathbf{u}(k)$ is the (r by 1) input vector, and $\mathbf{y}(k)$ is the (p by 1) output measurement vector. The vector $\nu(k)$ represents the state noise or disturbance inputs; $\mathbf{e}(k)$ is measurement noise, which, for periods of “normal” operation is assumed to be random with zero mean. The Φ, Γ, $\mathbf{C}$, and $\mathbf{D}$ matrices are assumed to be constant. Note that equation (3.1) is a recursive relationship giving the state vector at sampling period k+1 in terms of the states and inputs at period k.

---

[6] Appendix 1 considers the usefulness of PCA for investigation of dynamic systems. In addition, the effect of scaling options is also considered. Some of the examples in this chapter employ models developed in Appendix 1, therefore, the reader may wish to review it before proceeding.

Equation (3.2) shows how the measurements, $\mathbf{y}(k)$, are related to the states, inputs, and measurement noise.

For the purposes of this chapter, we assume that $\mathbf{D} = 0$, which implies that there is no instantaneous effect of changes in the inputs, $\mathbf{u}(k)$, on the process measurements. Due to the delay caused by sampling, this is a realistic assumption for most chemical processes. If, however, $\mathbf{D}$ is known (and non-zero), the effect of the term $\mathbf{Du}(k)$ in equation (3.2) can be subtracted from $\mathbf{y}(k)$, and the methods described below can still be applied in a straightforward manner.

Note that although the *number* of state variables required to describe a system, n, is a fundamental property of the system. This is typically referred to as the system order. On the other hand, the *coordinate system* defining the numerical values of the states may be chosen arbitrarily. This property of the model is exploited in the next section.

3.1.1 Immediately Observable States

We define the “immediately observable” states as the subset of the n state variables that can be estimated from a *single* sample of the p outputs. From an examination of equation (3.2), it is easy to see that the number of immediately observable states is equal to the rank of $\mathbf{C}$. Let $\mathrm{rank}(\mathbf{C})=q$ and note that $q \le \min(n,p)$.

One can transform the state-space model given in equations (3.1) and (3.2) to a form in which the immediately observable states are easy to calculate. We first perform a singular-value decomposition on the $\mathbf{C}$ matrix:

$$\mathbf{C} = \mathbf{U}\Sigma\mathbf{V}^T \qquad (3.3)$$

In this case, $\mathbf{U}$ is p by p, $\mathbf{V}$ is n by n, and Σ has the form:

$$\Sigma = \begin{bmatrix} \mathbf{S} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad \text{for} \quad q < p \text{ and } q < n \qquad (3.4a)$$

$$= \begin{bmatrix} \mathbf{S} \\ \mathbf{0} \end{bmatrix} \qquad \text{for } q < p \text{ and } q = n \qquad (3.4b)$$

$$= [\ \mathbf{S} \quad \mathbf{0}\ ] \qquad \text{for } q = p \text{ and } q < n \qquad (3.4c)$$

where **S** is a *q* by *q* diagonal matrix of singular values. Define a new matrix, **Q**, as follows:

$$\mathbf{Q} = \mathbf{V}\begin{bmatrix} \mathbf{S}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \qquad \text{for } q < n \qquad (3.5a)$$

$$= \mathbf{V}\mathbf{S}^{-1} \qquad \text{for } q = n \qquad (3.5b)$$

where in equation (3.5a), the matrix in brackets is *n* by *n*, and **I** is an identity matrix of size *n*-*q*. Note that **Q** is defined such that $\mathbf{Q}^{-1}$ exists in all cases. We can now define a new coordinate system for the state variables, which is related to the original one according to: $\mathbf{x}_r(k) = \mathbf{Q}^{-1}\mathbf{x}(k)$, where $\mathbf{x}_r$ is the state vector in the new (rotated) coordinate system. Substituting $\mathbf{x}(k) = \mathbf{Q}\mathbf{x}_r(k)$ into equations (3.1) and (3.2), we obtain (for $\mathbf{D} = \mathbf{0}$):

$$\mathbf{x}_r(k+1) = \Phi_r\mathbf{x}_r(k) + \Gamma_r\mathbf{u}(k) + \mathbf{Q}^{-1}\nu(k) \qquad (3.6)$$

$$\mathbf{y}(k) = \mathbf{C}_r\mathbf{x}_r(k) + \mathbf{e}(k) \qquad (3.7)$$

where

$$\Phi_r = \mathbf{Q}^{-1}\Phi\mathbf{Q} \qquad (3.8)$$

$$\Gamma_r = \mathbf{Q}^{-1}\Gamma \qquad (3.9)$$

$$\mathbf{C}_r = \mathbf{U}\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \qquad \text{for } q < p \text{ and } q < n \qquad (3.10a)$$

$$= \mathbf{U}\begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} \qquad \text{for } q < p \text{ and } q = n \qquad (3.10b)$$

$$= \mathbf{U}\ [\ \mathbf{I} \quad \mathbf{0}\ ] \qquad \text{for } q = p \text{ and } q < n \qquad (3.10c)$$

In equation (3.10), the matrix in brackets is *p* by *n* in all cases. Note that the new state-space model has the same form as that in equations (3.1) and (3.2), and the input and output behavior is also identical. Thus, the only effect of the change of coordinates is in the numerical values of the states.

As is apparent from equation (3.7), only the first *q* states in $\mathbf{x}_r$ have a direct effect on the outputs. Let $\mathbf{x}_q(k)$ be the vector of the first *q* variables in $\mathbf{x}_r(k)$, *i.e.*, the immediately observable states. Then equation (3.7) can be re-written as:

$$\mathbf{y}(k) = \mathbf{P}_q\mathbf{x}_q(k) + \mathbf{e}(k) \tag{3.11}$$

where $\mathbf{P}_q$ is an orthonormal matrix defined as:

$$\mathbf{P}_q = \mathbf{U}\begin{bmatrix}\mathbf{I}\\ \mathbf{0}\end{bmatrix} \quad \text{for } q < p \tag{3.12a}$$

$$= \mathbf{U} \quad \text{for } q = p \tag{3.12b}$$

Given a sample of the outputs, $\mathbf{y}(k)$, we can use equation (3.11) to estimate $\mathbf{x}_q(k)$ as follows:

$$\hat{\mathbf{x}}_q(k) = \mathbf{P}_q^T\mathbf{y}(k) \tag{3.13}$$

The corresponding estimate of the measurement noise is:

$$\begin{aligned}\hat{\mathbf{e}}(k) &= \mathbf{y}(k) - \hat{\mathbf{y}}(k)\\ &= (\mathbf{I} - \mathbf{P}_q\mathbf{P}_q^T)\,\mathbf{y}(k)\end{aligned} \tag{3.14}$$

where

$$\hat{\mathbf{y}}(k) = \mathbf{P}_q\hat{\mathbf{x}}_q(k)$$

$$= \quad \mathbf{P}_q\mathbf{P}_q^T\mathbf{y}(k) \tag{3.15}$$

As discussed, *e.g.*, by Ricker (1990), equation (3.13) provides an optimal estimate of the states in the sense that it minimizes $\hat{\mathbf{e}}^T(k)\hat{\mathbf{e}}(k)$, *i.e.*, it is the solution of the linear-least-squares problem. Note that when $q = p$, $\mathbf{P}_q\mathbf{P}_q^T = \mathbf{I}$, and the estimated measurement noise will be zero, regardless of the values of the outputs. In this case, the estimated measurement noise cannot provide useful diagnostic information about the system. Otherwise, however, we can compare the current value of $\hat{\mathbf{e}}(k)$ to our expectations (based on a statistical analysis of past process behavior). As is shown in the next section, this forms the basis for PCA monitoring of the process. The most favorable situation occurs when $p >> q$, and, in particular, when $p >> q = n$. In this case, a fault in a single measurement is most likely to show up in the residuals, rather than in the estimated states.

The state-estimation approach described above is quite different from that used in the development of the Kalman Filter (see, *e.g.*, Ricker 1990 or Åstrom and Wittenmark 1984). The goal of the Kalman Filter is to obtain estimates of all n of the states. In the present case, this can only happen when $q = n$. Furthermore, the Kalman Filter is designed to minimize the squared error in the state estimates (not the estimated measurement noise). To do so, the filter compensates for the statistical characteristics of both the measurement noise, $\mathbf{e}(k)$, *and* the state disturbances, $\nu(k)$. Its main disadvantage is that it requires a complete dynamic model of the system (*i.e.*, the Φ and Γ matrices in addition to the $\mathbf{C}$ matrix), and one must specify the expectations of $\mathbf{e}(k)$ and $\nu(k)$. The large amount of required information is difficult to obtain for a chemical process of typical complexity. Equations (3.15) to (3.15), on the other hand, are essentially a simpler “filter”, which can be designed based on a knowledge of $\mathbf{C}$ only.

3.1.2 The Basis for PCA Monitoring

Imagine that we have an immediately-observable process that can be modeled by equations (3.1) and (3.2). Let n be the number of states needed to describe the process dynamics. Assume that we have collected m “calibration” samples of the p outputs, forming the m by p matrix, $\mathbf{Y}$ (with $m \geq p$). Then according to equation (3.2), the relationship of the sampled outputs to the states and measurement noise is:

$$\mathbf{Y} = \mathbf{X}\mathbf{C}^{\mathrm{T}} + \mathbf{E} \tag{3.16}$$

where $\mathbf{X}$ is an m by n matrix of state variables (in which each row is the vector of state variables at a particular sampling period), and $\mathbf{E}$ is an m by p matrix of measurement noise signals. Both $\mathbf{X}$ and $\mathbf{E}$ are unknown, and in the general case, $\mathrm{rank}(\mathbf{X})=\mathrm{n}$ and $\mathrm{rank}(\mathbf{E})=\mathrm{p}$.

If we perform a PCA decomposition on $\mathbf{Y}$, retaining q latent variables, we obtain:

$$\mathbf{Y} = \mathbf{T}_{\mathrm{q}}\mathbf{P}_{\mathrm{q}}^{\mathrm{T}} + \mathbf{E}_{\mathrm{p-q}} \tag{3.17}$$

Comparing this to equations (3.9) and (3.16), we see that PCA can be interpreted as a state-space model, with $\mathbf{T}_{\mathrm{q}}$ representing estimates of the q immediately observable state variables (in a particular coordinate system) at each sampling period, and $\mathbf{P}_{\mathrm{q}}$ taking the place of the $\mathbf{C}$ matrix. Since $\mathbf{P}_{\mathrm{q}}$ is orthonormal, PCA automatically gives us a state-space model in the form of equation (3.9). Note, however, that $\mathrm{rank}(\mathbf{E}_{\mathrm{p-q}})=\mathrm{p-q}$. In other words, unless the “true” measurement noise, $\mathbf{E}$, is also rank p-q, the PCA estimate will be biased. This may or may not be a problem in practice, depending on the nature of the experiments used to obtain $\mathbf{Y}$, as discussed in the next section.

Let us assume for the moment that $\mathbf{P}_{\mathrm{q}}$ is an accurate representation of the true $\mathbf{C}$ matrix (which implies that we know the number of immediately observable states, q). The PCA residual for a given output sample, $\mathbf{y}(\mathrm{k})$, is defined as:

$$\mathbf{r}(k) = (\mathbf{I} - \mathbf{P}_q\mathbf{P}_q^T)\mathbf{y}(k) \qquad (3.18)$$

Thus the PCA residuals are identical to the estimated measurement noise as defined in equation (3.14). In other words, if the model is correct, the residuals are a function of $\mathbf{e}(k)$ only, regardless of the system dynamics. Since we have stipulated that, under normal conditions, $\mathbf{e}(k)$ is random with zero mean (and uncorrelated with previous values of $\mathbf{e}$), we can use well-established statistical methods to monitor $\mathbf{r}(k)$ and verify that it indeed satisfies this condition.

3.1.3 Obtaining an Accurate Estimate of P_q

The success or failure of PCA monitoring depends, to a large extent, on the accuracy of the model, *i.e.*, the matrix $\mathbf{P}_q$. Normally, this matrix must be estimated from the calibration data $\mathbf{Y}$, (since it is rare that an accurate theoretical model is available), as outlined in the previous section. It is clear that the most favorable situation occurs when the term $\mathbf{XC}^T$ in equation (3.16) is large relative to the noise, $\mathbf{E}$. If the system is controllable (see, *e.g.*, Åstrom and Wittenmark 1984), then it is possible to carry out an experiment in which the input variables, $\mathbf{u}(k)$, are varied so as to achieve a good signal-to-noise ratio for all n state variables. In practice, however, good estimates of $\mathbf{P}_q$ are obtained even for very noisy systems provided that the noise is uncorrelated, *i.e.*, when the singular values of $\mathbf{XC}^T$ are large relative to the singular values of $\mathbf{E}$.

On the other hand, if $\mathbf{u}$ is allowed to vary “naturally”, or if the system includes states that are accessible only through the disturbance vector, ν, then it will not be possible to guarantee adequate excitation of all *n* states. For example, if calibration data were collected during a period when ν was relatively small, the resulting estimate of $\mathbf{P}_k$ might not reflect the influence of all the states. If this $\mathbf{P}_k$ were then used to filter new data, as in equation (3.14), and ν suddenly became large, the value of $\hat{\mathbf{e}}(k)$ might signal a fault. This may or may not be desirable depending upon the situation. If the purpose of the monitoring were to detect disturbances, such a model would be effective. If the objective were to detect sensor failures and fundamental process changes, however, the model would give false alarms when disturbances occurred.

3.1.4 Using the PCA Residuals

Once an estimate of $\mathbf{P}_k$ is obtained, it can be used to calculate residuals as shown in equation (3.18). As shown in Chapter 2, confidence limits can be calculated for the Q statistic, (the total magnitude of the residual), as well as for the residual on any individual variable. As pointed out by Jensen and Solomon (1972), in practice the underlying distribution of the residuals of individual variables can vary substantially from Gaussian without affecting the results.

3.2 Some Examples of PCA Monitoring

In this section we will consider two examples of the use of PCA for process monitoring. In the first case, a state-space model will be used to generate synthetic process data from which a PCA model will be identified. Examples using this model will demonstrate that when the fundamental assumption of PCA monitoring is correct, *i.e.*, when there are more measurements than states, then the PCA residuals are indeed uncorrelated. In the second case, a PCA model will be identified from West Valley LFCM data. Here the actual number of states is not known, but will be estimated from the data. In fact, using any number of states less than the number of measurements can regarded as an approximation. In spite of this, it will be demonstrated that the PCA monitoring method is still effective.

3.2.1 PCA Monitoring Example Using Synthetic Data

As our first example we will consider a process with r=5 inputs, n=5 states and p=10 measurements. The Φ, Γ and $\mathbf{C}$ matrices for the process are given below in Tables 3.1 to 3.3. These matrices were generated randomly, although care was taken to assure that the resulting system was asymptotically stable and the non-zero singular values of the matrices were all relatively large. The **D** matrix is zero. Note that $q=n$ in this case.

Table 3.1. Φ Matrix for Example System.

0.3696	-0.2761	-0.0582	-0.6364	0.1188
0.0216	-0.4511	-0.2586	0.3415	0.4932
0.6204	-0.0227	0.4012	0.2988	0.0633
-0.2987	-0.1517	0.5948	-0.1786	0.3078
0.0295	0.4772	-0.0921	-0.1311	0.5786

Table 3.2. Γ Matrix for Example System.

0.0955	-0.6535	-0.0114	0.3726	0.0330
-0.3014	-0.2935	0.5805	-0.2808	0.2099
-0.2041	-0.0979	-0.4576	-0.2390	0.4931
0.3669	0.1920	0.1957	0.2874	0.6066
0.5820	-0.2586	-0.0619	-0.4934	0.0089

Table 3.3. **C** Matrix for Example System.

0.4219	-0.1386	-0.0126	-0.0922	-0.0189
0.0998	-0.0273	-0.1266	-0.4567	0.1252
0.0052	0.1546	0.5789	0.0325	0.1544
0.3851	0.0766	-0.2299	0.0466	0.3672
0.0888	0.0554	0.2707	0.2040	0.4264
-0.1016	-0.3428	0.3047	-0.3706	-0.1408
-0.0620	-0.6174	0.0411	0.2417	0.1964
0.0498	-0.1887	0.0267	-0.2905	0.1761
-0.0243	-0.0041	0.0180	-0.2449	0.3680
0.4857	-0.0906	0.1995	-0.0216	-0.2322

The example process was driven by white noise of unit variance to produce a **Y** matrix consisting of 1000 samples of the 10 outputs. Uncorrelated measurement noise was added to each output sample. The variance of the noise was equal to the variance of the output for each output variable (*i.e.*, the resulting output was 50% deterministic variation and 50% measurement noise). The process outputs were scaled to zero mean and unit variance (autoscaled) and a PCA model was obtained according to equation (3.17) with $q=5$. The variance captured by the PCA model is given in Table 3.4, and the loadings vectors retained in the PCA model are given in Table 3.5.

Table 3.4. Variance Captured by PCA Model of Example Process Output.

```
PC#        Eigval  %Variance  %TotVar
1.0000    2.0645   20.6454   20.6454
  2.0000    1.6056   16.0561   36.7016
  3.0000    1.3760   13.7602   50.4618
  4.0000    1.3585   13.5847   64.0465
  5.0000    1.0537   10.5374   74.5838
  6.0000    0.5714    5.7135   80.2973
  7.0000    0.5308    5.3079   85.6053
  8.0000    0.5182    5.1817   90.7869
  9.0000    0.4777    4.7768   95.5637
 10.0000    0.4436    4.4363  100.0000
```

Here, the correct number of PCs to retain in the model is known, *i.e.*, 5. In practice this would have to be determined from cross-validation or comparison to the expected ratio of successive eigenvalues for noisy data, as discussed previously.

Table 3.5. PCA Loadings Vectors for Example System.

```
 0.3904   -0.2121    0.4830   -0.0758    0.0769
 0.4309    0.2533   -0.0102    0.2610   -0.3237
-0.0060   -0.3306   -0.2772   -0.5159   -0.3525
 0.3321   -0.4258    0.0398    0.3923    0.1404
 0.1261   -0.5176   -0.3749   -0.2228    0.0726
 0.1761    0.4600   -0.0287   -0.5113   -0.0541
 0.1557    0.0731   -0.0823   -0.2425    0.8346
 0.5165    0.2522   -0.0760   -0.0947    0.0129
 0.4166    0.0031   -0.4885    0.1319   -0.0968
 0.2002   -0.2276    0.5437   -0.3315   -0.1760
```

A new data set, **X**, of 1000 samples was generated using the same process model and a new input sequence, which in this case was a low-frequency psuedo-random binary sequence (PRBS). The PCA model was applied to the outputs as in equation (2.10), and a residuals matrix was generated. The autocorrelation function was calculated for the raw outputs and the residuals. These are plotted in Figure 3.1, which clearly shows that, while the outputs are correlated in time, the residuals are not, which is the expected result when the $\mathbf{P}_k$ matrix obtained by PCA is an accurate representation of the original **C** matrix.

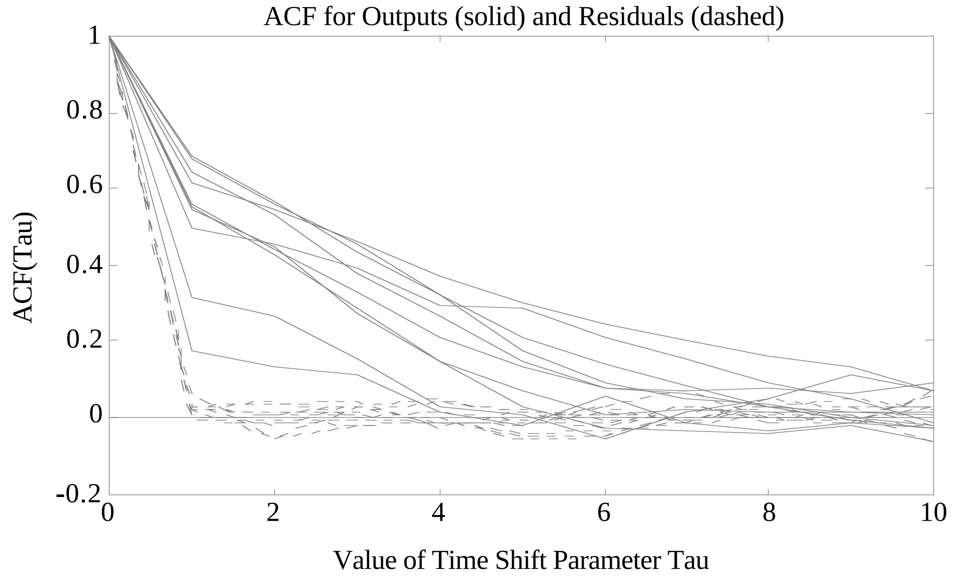

Figure 3.1. Autocorrelation Function for Outputs and Residuals for Test Data Set.

Because the residuals are not autocorrelated, the statistical tests in equations (2.19) and (2.22) can be used to test the residuals for faults and disturbances[7]. As mentioned previously, a sample "window width" must be chosen, and the desired confidence limits must be set before the test limits can be calculated.

For the example process we will choose, somewhat arbitrarily, a 20 sample window and 99% confidence limits. Thus the relevant statistic for detection of changes in the variance of the residuals is $F_{19,994,.01} = 1.91$ since we have 1000 samples in the original data, 20 in the new test sets, and desire 99% confidence limits. For changes in mean of the residuals $t_{1015} = 2.326$ and it is possible to calculate the change in mean that is just significant with (2.22) since the variances are known from (2.17). The detection limits can now be converted from the residual space back to the original variable space using (2.23). The results are shown in Figure 3.2, which gives the detection limits for changes in the

---

[7] It is also possible to develop detection limits based on the T^2 test for residuals given in equations (2.23) to (2.29). This is done for this example in Appendix 3. It has been found that the T^2 test for residuals is not as sensitive to failures as the simpler t- and F-based tests based tests used here. The reasons for this are discussed in Appendix 3, and some results from simulations are shown.

mean and standard deviation of the measurement noise for each process sensor (variable number) in the original measurement units.

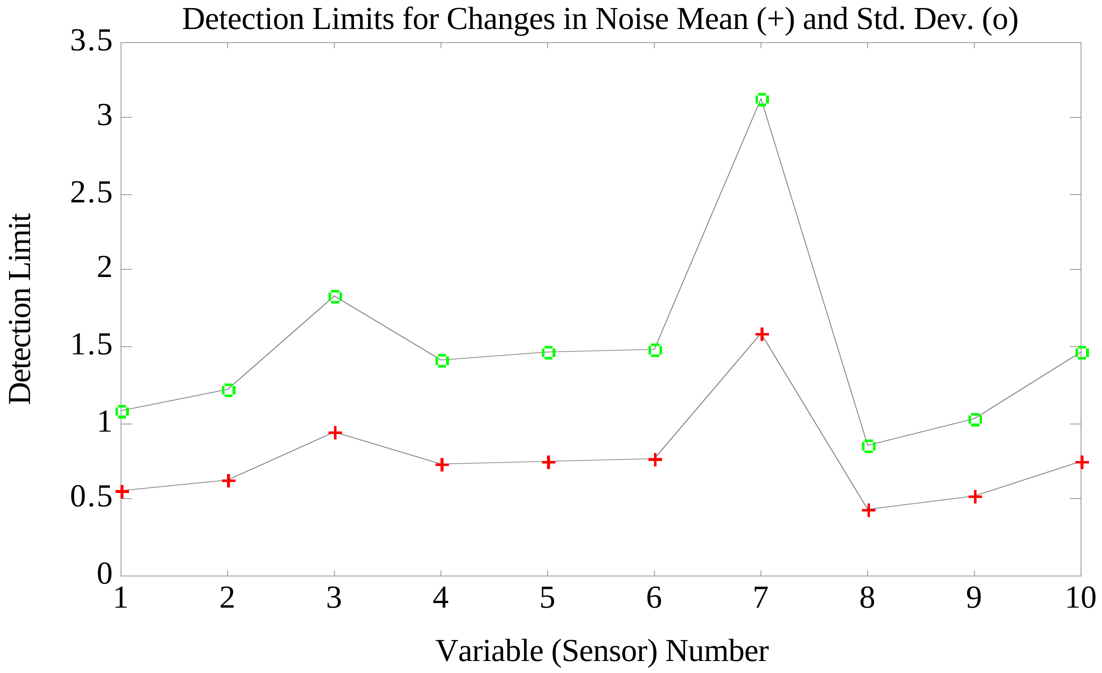

Figure 3.2. Detection Limits for Changes in Noise Mean and Standard Deviation in Original Units.

The detection limits shown in Figure 3.2 illustrate some important points. Note that variable number 7 has by far the worst detection limits for both mean and variance changes. A review of the loadings given in Table 3.5, however, shows that it is almost entirely included in the PCA model, *i.e.*, the sum of squared loadings for this variable in the retained PCs is very nearly 1. Variables that act nearly independently generally load strongly into only one PC. Thus, these variables tend to be either very strongly included in the model or very weekly included, but not intermediate. In any case, nearly independent variables have large detection limits because they are not highly correlated to other variables. Much of the error on such a variable is attributed to variations in the states (provided that the PC it is strongly loaded into is included in the model), and does not appear as a residual. This is easier to envision when one realizes that if a variable was entirely included in the model (*i.e.*, one of the states were defined as being equal to the measured value), its residual would always be zero.

On the other hand, measurements that have little influence on the state estimates also tend to have poor detection limits. The variables with the best detection limits are typically those that are highly correlated with other variables, which tends to make them be included in the model to an intermediate degree. In the situation where a variable does not load into the model at all, its residual variance is equal to its original variance. For a variable like this, standard SPC would work as well as MSPC.

3.2.2 PCA Monitoring Example Using LFCM Data

In this second example West Valley data from run SF-11 will be used. Specifically, the mean centered model indicated in Table A1.1 of the Appendix will be considered. In this case the first task is to determine the number of PCs to retain in the model. As mentioned previously, this can be done either through cross-validation or by considering the ratios of successive eigenvalues. Comparing the ratios it can be seen that it would be logical to keep either 4 or 7 PCs. In this case, we the example will be developed along parallel lines for both of these choices of PCs. This will allow observation of the effect of changing the number of factors in the model.

The autocorrelation function for the SF-11 data and the 4 and 7 PC residuals is shown below in Figure 3.3. The figure includes only variables 2, 4, 6, 8 and 10 for clarity; the other variables showed similar behavior. Note how the ACF of the raw data, shown as solid lines, has a very long correlation time. The amount of autocorrelation in the 4 PC residuals, shown as dotted lines, is much less but still significant. The ACF of the 7 PC residuals, shown as the plus marks, shows very little correlation, however. After the first sample instant it is essentially zero.

Based on the calculated ACF it would be expected that the residuals of the 4 PC model would not meet the criteria of random variables, and thus the statistical limits as calculated in the previous example would not apply. The 7 PC model residuals, however, should be much closer to the random variable assumption.

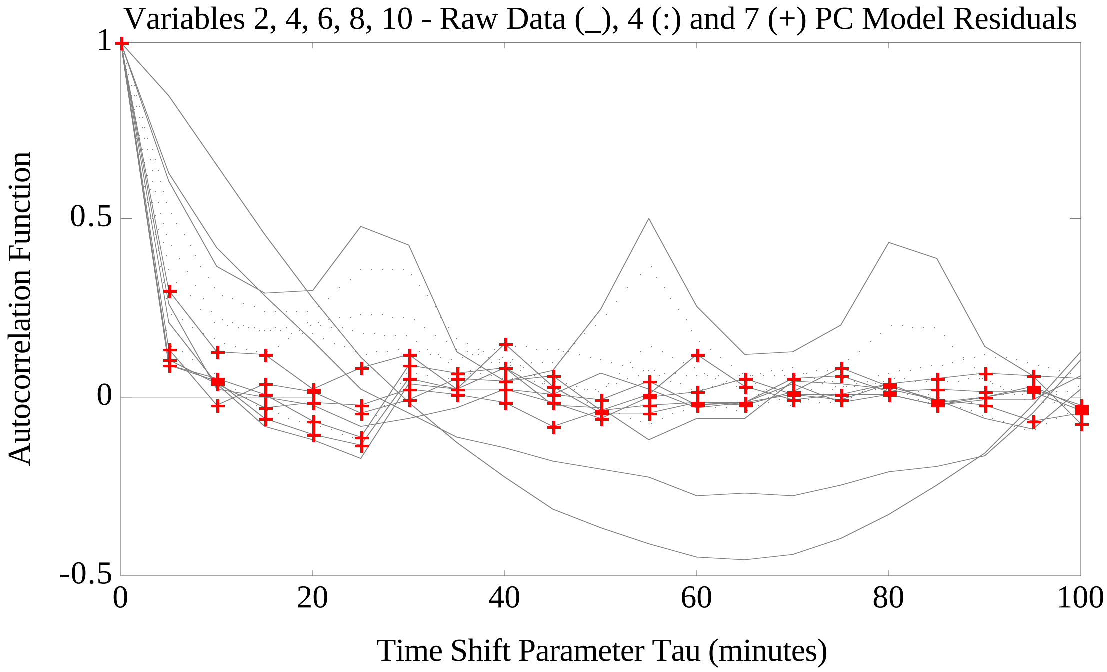

Figure 3.3. Autocorrelation Function of SF-11 Data and 4 and 7 PC Model Residuals.

It is generally found that the amount of autocorrelation in the residuals decreases as the number of PCs retained in the model is increased, up to a point where the residuals become essentially uncorrelated. This makes sense from a physical standpoint. Any persistent change in one variable must eventually be seen in the variables in the surrounding region, producing correlated outputs for neighboring variables. PCA, of course, seeks to build the correlation between variables into the model, leaving behind only random fluctuations. Large variations tend to be more persistent in time, affect many variables and thus get captured in the first few PCs. Smaller variations are less persistent, affect fewer variables and thus are captured in later PCs.

As in the previous section, it is now possible to calculate the detection limits in terms of the units of the original variables once the sample window size and the desired confidence limits have been chosen. Here again we will arbitrarily choose 20 samples (100 minutes) and a confidence level of 99%. The calculated detection limits for changes in the mean are shown in Figure 3.4 for both the 4 and 7 PC models. The corresponding limits for changes in the variance are shown in Figure 3.5, where the limits are actually shown in standard deviations rather than in units of variance.

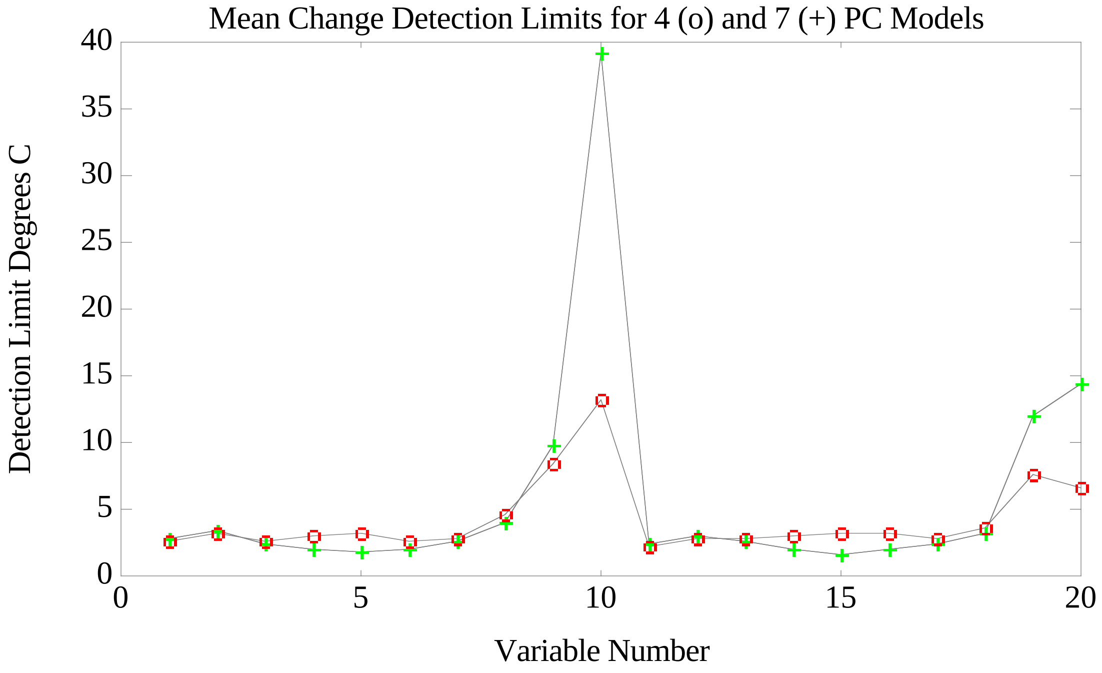

Figure 3.4. Detection Limits for Changes in Mean Residual From Mean Centered SF-11 Data Using 4 and 7 Principal Component Models.

In Figures 3.4 and 3.5 it is apparent that the detection limits for all the variables are not equal. This would be expected based on physical arguments. The detection limits for the bulk glass variables are relatively small because because there are many nearly redundant measurements made in close proximity. Furthermore, these sensors very little variance to begin with. The plenum temperatures, however, vary widely and are measured in only 2 locations, resulting in rather large detection limits.

The differences between the calculated limits for the 4 and 7 PC models is a result of the changes in the amount that particular variables are included in the model depending upon the number of PCs retained. In the case of variable 10, when the 7 PC model is employed the variable is so strongly included in the model (sum of squared loadings for this variable is 0.9941) that it essentially has no residual, and therefore it takes a huge change in such a variable to make a significant difference. In Figure 3.5, the limit for this variable is off the chart at 169. Some of the bulk glass temperature variable detection limits are better in the 7 PC model because more of the correlation between these variables shows up in PCs 5-7 than in the previous PCs, driving the detection limits down.

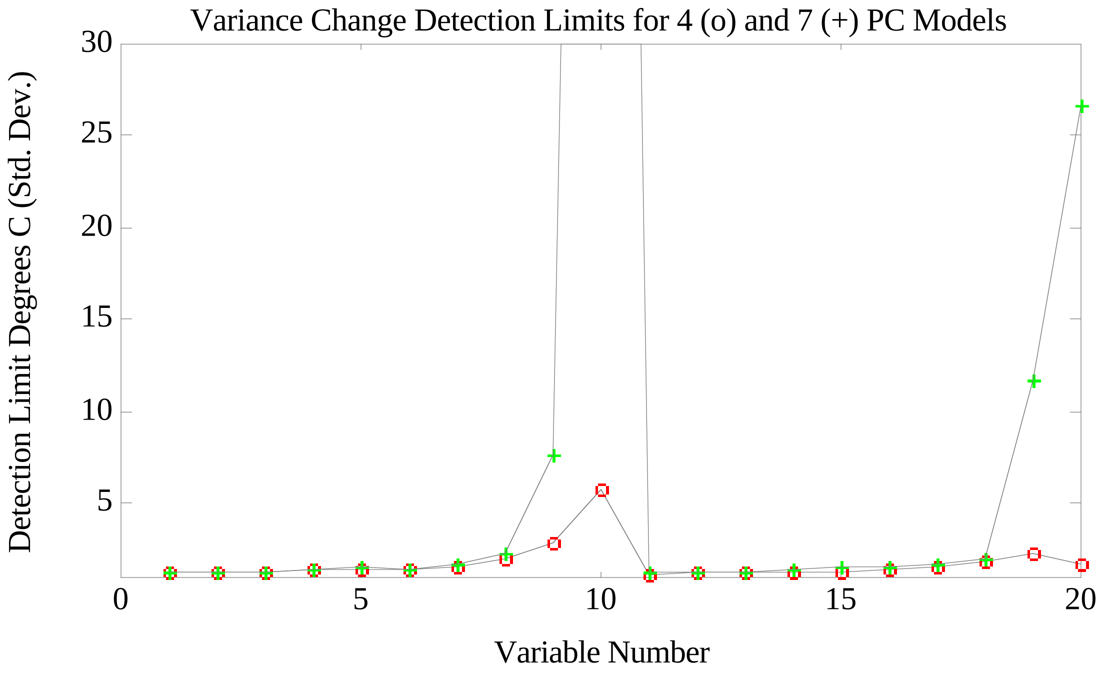

Figure 3.5. Detection Limits for Changes in Standard Deviation of Residual From SF-11 Data Using 4 and 7 Principal Component Models.

It should be noted that, while the 4 PC model looks much better in terms of detection limits, the autocorrelation function of the 4 PC residuals shows much more correlation. This will have the effect of invalidating the statistics that assume that the residuals are random variables. Thus we must strike a balance between the residual correlation and the apparent detection power of the models. In this case, one might expect that the 6 PC model may be a much better choice, as it does not include the plenum temperatures to the extent that the 7 PC model does, while still eliminating the bulk of the autocorrelation. Inspection of the ACF of the 6 PC model residuals, however, shows that there is still a very large amount of autocorrelation.

3.3 Effect of Scaling on PCA Monitoring

In order to better understand the effects of scaling on the PCA monitoring effectiveness, the previous example using LFCM data with mean centering is now repeated using autoscaling. The variance captured by the PCA model is shown in Table A1.2. The ratio of successive eigenvalues again suggest that 4 or 7 PCs would be logical for this model. The ACF of the residuals was calculated for PCA models using 4, 7 and 10 PCs. Some representative ACFs are shown below in Figure 3.6 for the 4 and 7 PC model case, and are compared to the ACF of the raw data. It is apparent from Figure 3.6 that the mean centering option is somewhat more effective at removing autocorrelation from the residuals than the autoscaling. Even the 10 PC residuals from autoscaling (not shown) were more correlated than the 7 PC residuals from the mean centering.

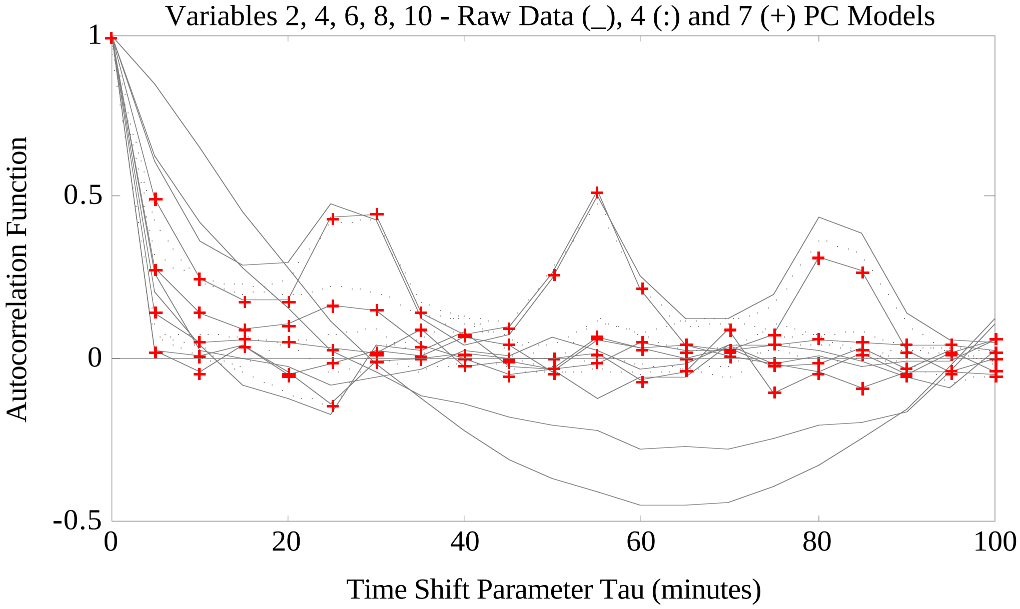

Figure 3.6. Autocorrelation Function of SF-11 Data and 4 and 7 PC Model Residuals from Autoscaled Data.

The reason that mean-centering removes the correlation from the PCA residuals more effectively than autoscaling in this situation becomes apparent after considering a small thought experiment. Imagine for a moment a data set where the variables have widely different deterministic variances, but have added measurement noise of identical magnitudes. Furthermore, assume that the deterministic variation in the data set is confined to a subspace of the data space, *i.e.*, the data is deterministically rank deficient. In this case the autoscaling option would have the effect of increasing the variance of variables that are mostly noise relative to those that are mostly deterministic variation. This increases the effective noise level of the data set. Any PCA model (with a specified number of PCs) formed from this data would, therefore, be more likely to capture noise in the model and, thus, less likely to include all deterministic variation. Any deterministic variation that is not captured by the model will lead to autocorrelated residuals in the dynamic model case.

The theoretical detection limits for bias errors (changes in noise mean) and noise errors (changes in noise variance) were calculated for the 4, 7 and 10 PC models and are shown in Figures 3.7 and 3.8, respectively. The calculated limits are very similar to those calculated from the mean centered data in the previous section. It is important to remember, however, that the mean centered data has the autocorrelation in the variables more effectively removed than in the autoscaled data. Thus it would be expected that the serial correlation in this data would have a tendency to enlarge the detection limits over the theoretical limits shown here.

3.4 Treating Autocorrelation in PCA Residuals

As shown in the previous examples, PCA can effectively remove most, but not all, of the autocorrelation in the model residuals. This is a direct result of the approximations made when using PCA, *i.e.*, that the process is linear and can be reduced to some number of states that is less than the number of measured variables. In reality, all processes are infinite dimensional. However, their dynamics are typically dominated by a few states which persist in time. Minor states tend to have more transitory effects on the outputs, but these states do cause some "leftover" autocorrelation in the PCA residuals.

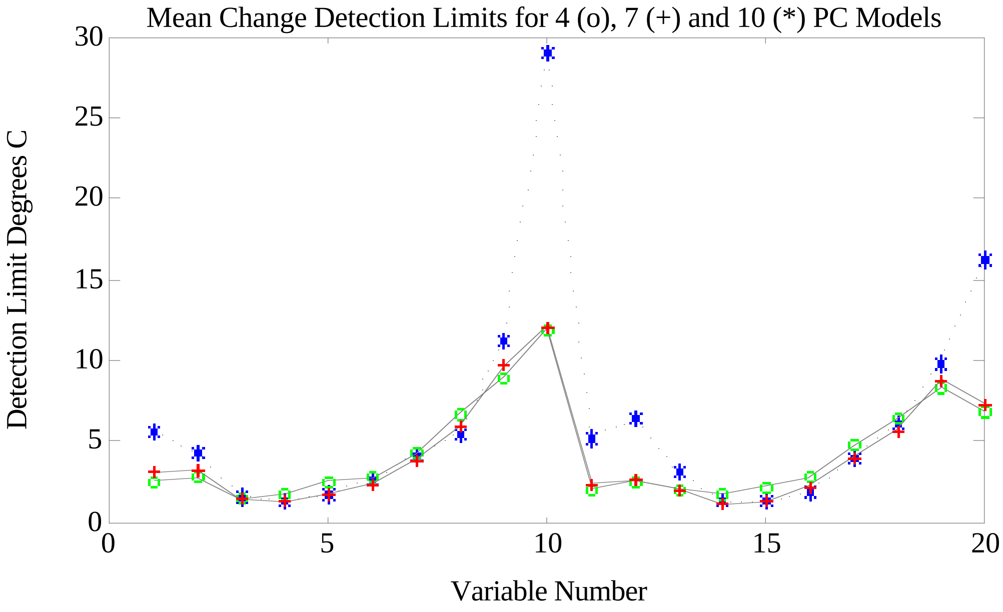

Figure 3.7. Detection Limits for Changes in Mean From SF-11 Data Using 4, 7 and 10 Principal Component Models from Autoscaled Data.

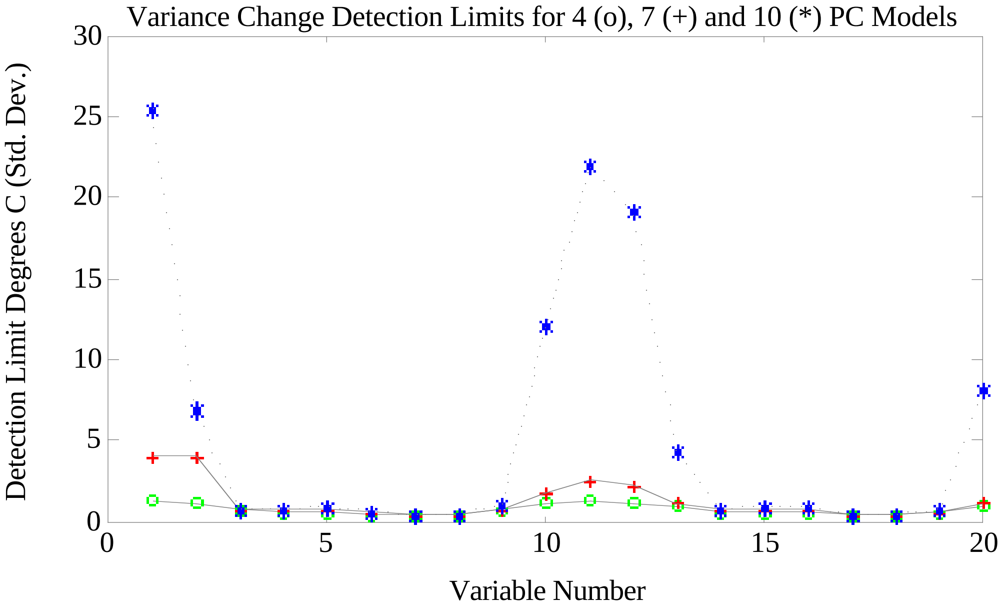

Figure 3.8. Detection Limits for Changes in Variance From SF-11 Data Using 4, 7 and 10 Principal Component Models from Autoscaled Data.

It is beyond the scope of this work to thoroughly treat all the possible ways for dealing with the remaining autocorrelation, but there are several methods that will be discussed briefly here. A brute force approach, which works well when there is an abundance of data, (such as in the LFCM examples given here), is to estimate the residual limits using subgroups of the data set used to calibrate the PCA model. In this method, the data is divided into subgroups of sequential samples with each subgroup containing as many samples as the number of samples in the "window" used for monitoring. The mean and variance of these subgroups is then calculated, and the distribution of the calculated mean and variance is determined. Limits are then chosen based on the confidence limit specified, *e.g.* if 99% limits were desired, the limits would be set so that only 1% of the subgroups had a calculated mean and variance outside of the limits. The major disadvantage of this approach is that it is entirely empirical. It makes no assumptions about the actual distribution of the residuals, but it does assume that the residuals used for calibration are typical and that the process will continue to produce similar data. Also, one would expect that, given enough data, the mean and variance limits calculated this way would approach the theoretical limits for the actual autocorrelated data.

In the example case, the number of totally separate subgroups of 20 samples for this data set is 25. Many of the data sets that have been used in this research have many thousands of samples available and thus would have many hundreds of independent subgroups available for calculating limits.

Another way of removing residual autocorrelation is to decrease the sample rate. It is easy to see how if the sample rate were increased enough, eventually all the residuals from the process would become autocorrelated. This is because variables in the process cannot change instantaneously for physical reasons. Going the other way, if the sample rate is decreased a sufficient amount, the residuals will become random; the effect of the minor states noted above dies out in a sufficiently long period of time.

Finally, there are more explicit methods of treating the residual correlation. These methods, such as those demonstrated by Harris (1990), start by fitting an auto-recursive model to the data which quantifies the degree of autocorrelation. These models are then used to set the limits on the mean and variance of sample subgroups. It would be expected that the limits calculated in this way would be similar to the limits calculated in the "brute force" empirical approach described above.

3.5 Robust PCA for Multiple Failures

In the preceding sections, a theoretical basis for using PCA with dynamic systems has been developed. This is an important step towards the development of PCA for process monitoring. However, other problems remain, including problems with the robustness of the monitoring method. A problem of particular interest concerns the continued use of the PCA model after a sensor has been identified as bad. It has been shown in the previous sections and in Wise and Ricker (1989b) that the method can identify a bad sensor, but can the PCA model still be used after that? If the data from the sensor continues to be included in the PCA monitoring it has the tendency to “mask” other events. For instance, because the Q residual is already large, a change in another sensor that would normally lead to a large residual tends to get “covered up” and may not be noticed.

It seems logical that there should be an optimal way of either replacing “bad” data or modifying the existing PCA model so that changes in other variables can be observed. One would also hope that it would be possible to do this in such a manner so that the same overall statistics that were developed around the PCA model could be used without modification. The solution to the problem of replacing data from single or multiple sensors follows.

Assume for the moment that the PCA model of the process of interest has been calculated, and that the matrix used for calculating the residuals has been obtained:

$$\mathbf{I} - \mathbf{P}_k\mathbf{P}_k^T = \mathbf{R}_m \qquad (3.19)$$

The Q residual for any sample **x** (a row vector) can then be calculated as:

$$Q = \mathbf{x}\mathbf{R}_m\mathbf{x}^T \qquad (3.20)$$

Suppose now that one or more sensors have failed and have been detected, using either the methods developed here or some other similar method. Further, suppose that it is convenient to partition **x** (possibly by rearranging the columns of **x**) into a group of “bad” sensors $\mathbf{x}_b$, and a group of good sensors $\mathbf{x}_g$. Thus:

$$\mathbf{x} = [\mathbf{x}_b \ \mathbf{x}_g] \qquad (3.21)$$

Furthermore, it is now possible to partition **R** into parts that act on each of the groups of good and bad sensors individually:

$$\mathbf{R_m} = \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{R}_{21} & \mathbf{R}_{22} \end{bmatrix} \qquad (3.22)$$

The Q residual is now calculated as:

$$Q = \begin{bmatrix} \mathbf{x}_b \ \mathbf{x}_g \end{bmatrix} \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{R}_{21} & \mathbf{R}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{x}_b^T \\ \mathbf{x}_g^T \end{bmatrix} \qquad (3.23)$$

By multiplying this through, and using the fact that $\mathbf{R}_{21} = \mathbf{R}_{12}^T$, the following expression for Q is obtained:

$$Q = \mathbf{x}_b\mathbf{R}_{11}\mathbf{x}_b^T + \mathbf{x}_g\mathbf{R}_{21}\mathbf{x}_b^T + \mathbf{x}_b\mathbf{R}_{21}^T\mathbf{x}_g^T + \mathbf{x}_g\mathbf{R}_{22}\mathbf{x}_g^T \qquad (3.24)$$

It would seem logical at this point to find values for the bad variables $\mathbf{x}_b$ which minimizes Q. This is equivalent to finding the values of the bad variables which are most consistent with the PCA model. Note that in equation (3.24) the last term is a function of $\mathbf{x}_g$ and $\mathbf{R}_{22}$ only, and is therefore fixed. The problem of minimizing Q then becomes the problem of minimizing the first three terms on the right hand side of (3.24). Thus the objective becomes to find:

$$\mathbf{x}_b \; \forall \; \{\mathbf{x}_b\mathbf{R}_{11}\mathbf{x}_b^T + \mathbf{x}_g\mathbf{R}_{21}\mathbf{x}_b^T + \mathbf{x}_b\mathbf{R}_{21}^T\mathbf{x}_g^T\} = \min \qquad (3.25)$$

Fortunately, this problem can be easily solved by “completing the squares” (see for instance Åstrom, p 258). The result is:

$$\mathbf{x}_b = -\mathbf{x}_g\mathbf{R}_{21}\mathbf{R}_{11}^{-1} \qquad (3.26)$$

Because $\mathbf{R}$ is fixed, the value of $\mathbf{x}_b$ for any new sample can be calculated from

$$\mathbf{x}_b = \mathbf{x}_g\mathbf{R}_R \qquad (3.27)$$

where $\mathbf{R}_R$ is the regression matrix formed from $\mathbf{R}_m$:

$$\mathbf{R}_R = -\mathbf{R}_{21}\mathbf{R}_{11}^{-1} \qquad (3.28)$$

which can be calculated once and retained. Calculation of $\mathbf{R}_R$ should be possible, in general, because $\mathbf{R}_{11}$ will be positive definite. Computationally, the biggest problem occurs when the bad variables are not in a “convenient” location and the matrices $\mathbf{R}_{11}$ and $\mathbf{R}_{21}$ must be extracted from $\mathbf{R}$. The process for doing this is shown in Figure 3.9. Here variables 2 and 5 are assumed bad. The figure shows how the parts of the original $\mathbf{R}_m$ matrix map into $\mathbf{R}_{11}$ and $\mathbf{R}_{21}$.

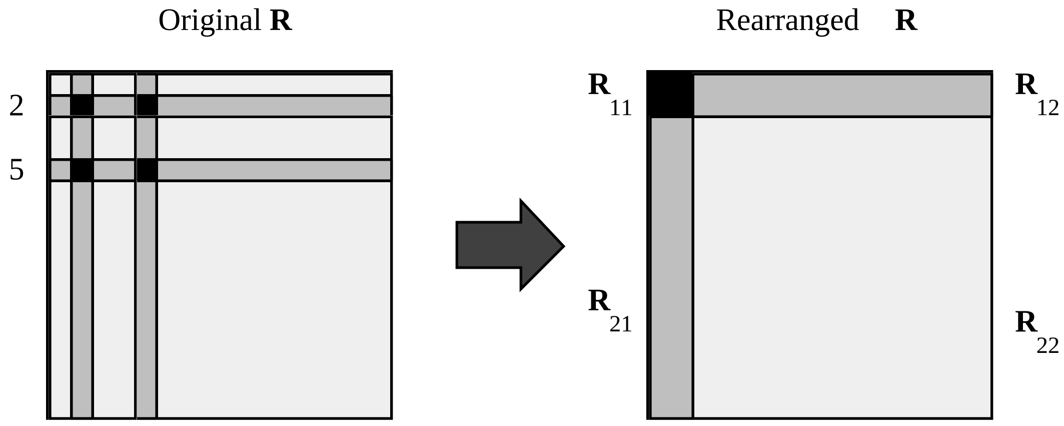

Figure 3.9. Obtaining $\mathbf{R}_{11}$ and $\mathbf{R}_{21}$ from Original $\mathbf{R}_m$ Matrix.

Once $\mathbf{R}_R$ has been obtained, it can be mapped into an identity matrix in such a way that multiplication of new samples by this matrix, the "replacement" matrix $\mathbf{R}_m$, results in the bad variables being replaced by their values that minimize Q. The mapping is shown in Figure 3.10. The solid black squares in the figure represent "ones", the white portions represent "zeros" and shaded portions are occupied by the regression matrix.

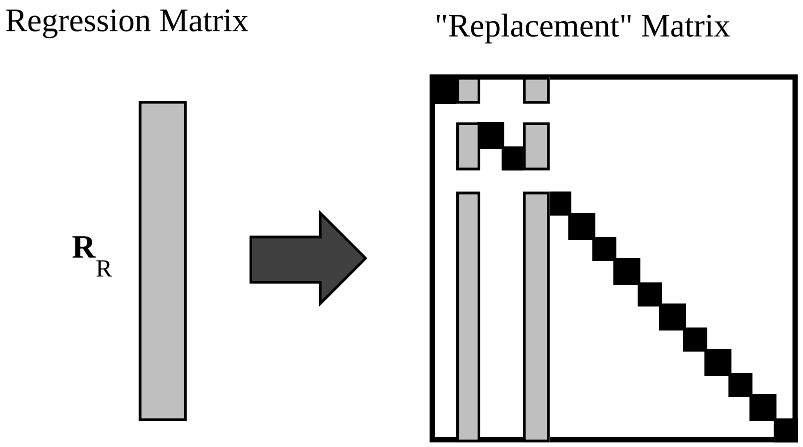

Figure 3.10. Mapping of Regression Matrix into "Replacement" Matrix.

It is found that when new samples are multiplied by $\mathbf{R}_M$ the PCA residuals of the replaced variables are identically zero. This can be seen by substituting the calculated values for $\mathbf{x}_b$ back into equation (3.24). This also leads to an expression for the minimum value of Q.

$$Q_{min} = -\mathbf{x}_g\mathbf{R}_{21}\mathbf{R}_{11}^{-1}\mathbf{R}_{21}^T\mathbf{x}_g^T + \mathbf{x}_g\mathbf{R}_{22}\mathbf{x}_g^T \qquad (3.29)$$

The problem that we are faced with now concerns the fact that the resulting Q values will tend to be artificially low. In order to remedy this, a “white” noise signal of zero mean and appropriate variance should be added to the calculated value of the bad variables. Ideally, we would like the variation in the residual of the bad variables to mimic the normal behavior. Therefore, we must determine the proper noise variance required for this.

It has been shown previously in equation (2.17) that the variance of the residual for any particular variable can be calculated using the PC loadings and corresponding eigenvectors not retained in the PCA monitoring model. Furthermore, equation (2.23) relates the change in a variable to the change in its residual (with all other variables remaining constant). Using these two relationships it is easily seen that the variance of the white noise s_{nj}^2 to add to a replaced variable is:

$$s_{nj}^2 = s_j^2(\mathbf{R}_{jj})^{-2} \qquad (3.30)$$

where $\mathbf{R}_{jj}$ is the j<u>th</u> diagonal element of $\mathbf{R}_m$. The result of this will be that the residual of a replaced bad variable will have zero mean and a variance approximately equal to the expected variance based on (2.15). The “normal” statistics generated for this system will then still hold, *i.e.*, the null hypothesis that the residuals have zero mean and variance predicted by (2.15) will remain unchanged. Thus, any new changes in variables can be detected in the normal fashion, since the null hypothesis will no longer hold in the event of change.

It may be that for most practical monitoring problems the preceding scheme for replacing variables, (an approximation itself), is unnecessary. Using the original calculated limits may be close enough, particularly in cases where there are many variables relative to the number of PCs retained in the model. In this case the loss of the variance due to one variable has only a small affect on the total residual. The variance of other residuals will be slightly decreased, making the original limits at an effectively higher confidence level. In many applications the difference would be insignificant.

The alternative to replacing bad variables in an existing PCA model is to entirely rebuild the model. This approach was compared to the method of replacing variables. The somewhat surprising result was that the two methods are equivalent. In the noise free case, where the data is truly rank deficient, the two methods produce identical results. The proof of this is shown shown in Appendix 2. In the presence of noise, the solutions are approximately equal, though usually very close provided a sufficient number of samples are available.

To demonstrate the equivalence of the replaced variables versus new model approaches, a numerical example is provided here along with some figures to help the reader visualize the mathematics involved. Suppose that PCA is used to model a process with 3 variables, but analysis shows that the data is essentially one dimensional. The PCA model of the process would then be a single vector **p** which is arbitrarily chosen for this example to be

$$\mathbf{p} = [0.5579 \;\; 0.7748 \;\; 0.2974]^T$$

Now suppose there is a new sample **x**

$$\mathbf{x} = [0.7 \;\; 0.6 \;\; 0.4]$$

The residual **r** of this sample on the model will be

$$\mathbf{r} = [0.1564 \;\; -0.1550 \;\; 0.1103]$$

The process of projecting the sample **x** onto the model **p** is shown in Figure 3.11.

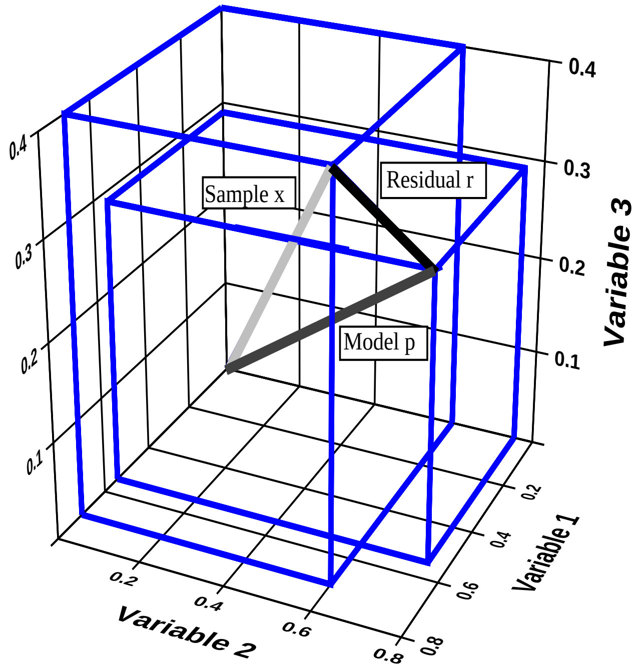

Figure 3.11. Projection of Sample Onto Model.

Now suppose that we have reason to believe that the sensor that which is represented by variable 1 has drifted. We can now use the method outlined previously to estimate the actual value of variable 1 using the original model and the information from variables 2 and 3. Upon doing this we obtain a new estimate of the sample **x**, $\mathbf{x}_C$

$$\mathbf{x}_C = [0.4729 \;\; 0.6 \;\; 0.4]$$

Note how the values of variables 2 and 3 have remained unchanged, however the value of variable 1 has been replaced. The residuals of this sample, $\mathbf{r}_C$ can now be calculated using the original model.

$$\mathbf{r}_C = [0.0000 \;\; -0.0568 \;\; 0.1479]$$

Note how the residual on the first variable is zero, as expected. The projection of this "corrected" sample is shown in Figure 3.12.

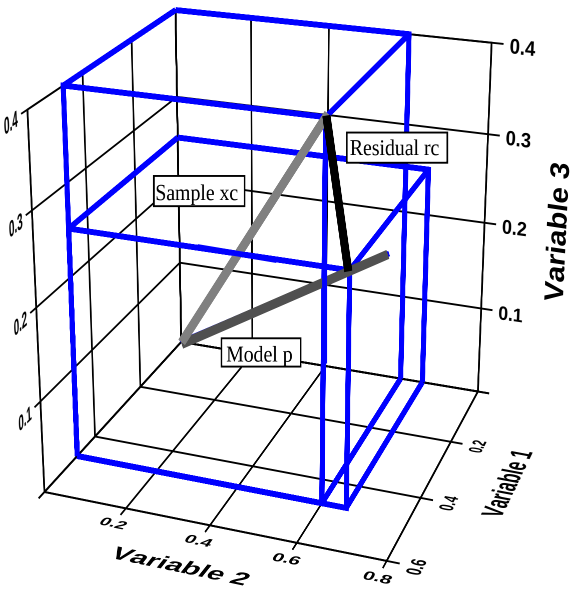

Figure 3.12. Projection of Corrected Sample $\mathbf{x}_C$ onto Model $\mathbf{p}$.

Now, imagine that instead of correcting samples for the failed sensor that a new model using only variables 2 and 3 has been developed. This new model is simply equal to the old model with variable 1 projected out of it. This is equivalent to projecting the model onto the plane formed by variables 2 and 3, then renormalizing it to unit length. When this is done the corrected model $\mathbf{p}_C$ is obtained

$$\mathbf{p}_C = [0 \;\; 0.9336 \;\; 0.3583]^T$$

When the residuals of the sample $\mathbf{x}$ with variable 1 set to zero, $\mathbf{x}_r$ are calculated, it is found that the residuals are identical to those given above for $\mathbf{r}_C$. The new model and residuals are added to Figure 3.12 and are shown in Figure 3.13. (Note that the projection does not appear to be orthogonal to the model due to a difference in scaling of the axes.)

3.6 Conclusions Concerning PCA Monitoring

In this chapter it has been shown when it is appropriate to use PCA to detect faults and upsets in dynamic processes. A particular advantage of the PCA approach (relative, *e.g.*, to the Kalman Filter) is that it provides a convenient way to estimate the **C** matrix in a standard discrete-time, LTI, state-space model (see for instance Ricker, 1990). Also, a complete dynamic model of the process is not required.

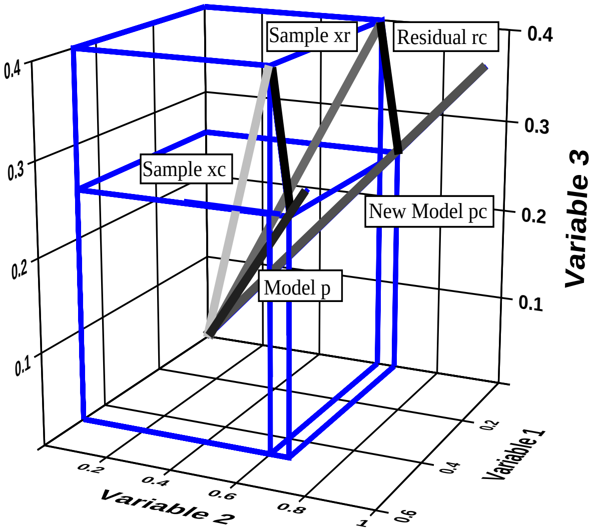

Figure 3.13. Projection of Corrected Sample $\mathbf{x}_C$ onto Model $\mathbf{p}$ Shown with Projection of Reduced Sample $\mathbf{x}_r$ onto New Model $\mathbf{p}_C$.

Proper application of PCA results in residuals that are non-autocorrelated. This greatly simplifies the statistics in fault-detection applications. Standard F- and t-tests can be used to monitor the residuals for changes in variance (arising from added sensor noise) and mean (arising from sensor bias or process changes), respectively. Other statistical tests, such as the maximum likelihood ratio used by Willsky (1976), could also be used (although these are not limited to processes with redundant measurements). In practice, while the sample autocorrelation is generally substantially reduced, some autocorrelation may remain. However, this can be treated by any of the approaches outlined above. It has also been shown that detection limits derived for the (scaled) PCA residuals can be used to calculate detection limits in terms of the measurement units of the original variables.

A point which should be emphasized is that MSPC is most effective when the process has significantly more measurements than imediately observable states. One could argue that all real systems have an infinite number of states and, therefore, could never have more measurements than states. There are many systems, however, in which a small number of states dominate the dynamic response. The remaining states are associated with transients that decay very quickly relative to the sampling period. For a given process, either an increase in the number of measurements or an increase in the sampling period will often be enough to make MSPC applicable. It is the contention here, therefore, that although MSPC cannot be used in all cases, the number of problems it can address is significant and is increasing as modern instrumentation systems become more common.

It should be pointed out that process faults which shift the output measurements within the subspace spanned by PCA model will not be detected by monitoring of the PCA residuals, though there would be some hope of detecting this type of error through observation of the PCA scores. Thus, there will be some errors to which the PCA method will be quite insensitive. This is discussed further in Ricker (1990) and Ricker and Douglas (1990).

An alternative method to completely rebuilding the PCA process model following the failure of a process sensor has also been demonstrated. There are several advantages to using the replacement method, including: the replacement method provides an estimate of the output of the failed sensor, computation time is reduced over rebuilding the model, and plant data displays do not have to be reset to account for changes in the model since the model remains the same. While the resulting confidence limits for error detection will be approximate when using the replaced variable, they will generally be quite close, particularly in applications with many variables but low intrinsic dimensionality of the data.

4.0 Process Monitoring with PLS

In chapter 2 (equations 2.55 to 2.59) it was proposed that PCA-like residuals could be generated with PLS. This approach was also demonstrated previously in Wise and Ricker (1989b). This chapter provides a basis for the PLS monitoring approach, gives some examples of its use and compares the effectiveness of the PLS monitoring method to PCA monitoring using both synthetic and LFCM data. The chapter closes with the solution to the "bad variable" problem for PLS monitoring that was solved for analogous PCA problem in the previous chapter.

4.1 A Basis for PLS Monitoring

In this section it will be shown that PLS monitoring is a logical extension of the PCA monitoring developed in the previous chapter. The argument proceeds as follows: using the results of Chapter 3 it is shown that a PCA model can be converted to another form that has detection power identical to the original model. It is shown that this transformed model has the form of a collection of regression models, arranged in a manner similar to the collection of PLS models described in Chapter 2. A collection of PLS models, therefore, is simply an optimization of these regression models to improve their predictive ability. Furthermore, the PLS models make no assumptions concerning the intrinsic dimensionality of the output variable space.

Suppose that we have a discrete LTI process with noise and with fewer states than measurements in the state-space form as defined by equations (3.1) and (3.2). Furthermore, suppose that an accurate PCA model $\mathbf{P}$ has been determined for the system, *i.e.*, $\mathbf{P}$ spans the same space as $\mathbf{C}$ from the "true" state-space model of the process. The PCA residuals for a new sample $\mathbf{y}(k)$ are given by

$$\mathbf{r}(k) = (\mathbf{I} - \mathbf{P}\mathbf{P}^T)\mathbf{y}(k) \tag{4.1}$$

Using the notation of Chapter 3, with $(\mathbf{I} - \mathbf{P}\mathbf{P}^T) = \mathbf{R}$ and with $\mathbf{R}$ partitioned as in equation (3.23), it is possible to write the residual on the first variable $r_1(k)$ as

$$r_1(k) = \mathbf{R}_{11}\mathbf{C}_1\mathbf{x}(k) + \mathbf{R}_{12}\mathbf{C}_2\mathbf{x}(k) + \mathbf{R}_{11}e_1(k) + \mathbf{R}_{12}\mathbf{e}_2(k) \qquad (4.2)$$

where $\mathbf{C}_1$ corresponds to the first row of the state-space $\mathbf{C}$ matrix and $\mathbf{C}_2$ is equal to the remaining rows 2 to m. It is known from the results of Chapter 3 that if the PCA model is accurate, the first two terms on the right hand side of equation (4.2) sum to zero, *i.e.* the states make no contribution to the residuals. Thus

$$r_1(k) = \mathbf{R}_{11}e_1(k) + \mathbf{R}_{12}\mathbf{e}_2(k) \qquad (4.3)$$

Suppose now that the PCA model is transformed so that the residuals on the transformed model $\mathbf{r}_t$ are defined as

$$\mathbf{r}_t(k) = \mathbf{y}(k) - \hat{\mathbf{y}}(k) \qquad (4.4)$$

where each of the $\hat{y}_i$ is estimated based on the variable replacement method in chapter 3, *e.g.*, for the first variable in the system

$$\hat{y}_1(k) = -\mathbf{y}_2\mathbf{R}_{12}\mathbf{R}_{11}^{-1} \qquad (4.5)$$

The first transformed model residual $r_{t1}(k)$ can now be written in terms of the state-space model parameters as

$$r_{t1}(k) = \mathbf{C}_1\mathbf{x}(k) + e_1(k) + \mathbf{C}_2\mathbf{x}(k)\mathbf{R}_{21}\mathbf{R}_{11}^{-1} + \mathbf{e}_2(k)\mathbf{R}_{21}\mathbf{R}_{11}^{-1} \qquad (4.6)$$

Note the correspondence between equation (4.6) and equation (4.2) above. From this it is easily seen that the residual $r_1(k)$ is different from $r_{t1}(k)$ by a factor of $\mathbf{R}_{11}^{-1}$. Furthermore, based on equation (2.21) it can be seen that the transformed model will have fault detection "power" identical to the original model. The normal residuals on the transformed model are larger by a factor of $\mathbf{R}_{11}^{-1}$, but when noise or bias is added the residuals on the new model also get larger by the same factor of $\mathbf{R}_{11}^{-1}$. In other words, the ratio of the expected size of the residuals to the size when a fault has occurred is the same for both the original and the transformed models. Furthermore, when the model is transformed in this manner, if a single variable changes, its residual changes by the same amount. The entire PCA model for calculating residuals $\mathbf{R}$ can be transformed to $\mathbf{R}_t$ as follows:

$$\mathbf{R}_t = \mathbf{R}(\mathrm{diag}(\mathbf{R}))^{-1} \tag{4.7}$$

where diag($\mathbf{R}$) is the matrix containing the diagonal elements of $\mathbf{R}$.

It is clear that the transformed model $\mathbf{R}_t$ has the form of a collection of regression models. Now, because of the "normalization" of the residuals, the predictive ability of $\mathbf{R}_t$ can be compared directly to a collection of PLS models $\mathbf{R}_{pls}$ as proposed in equations (2.55) to (2.59). The system used in Chapter 3, (Tables 3.1 to 3.3), will be used as an example of how the predictive ability of the models compare. For the test, data was generated with the example system exactly as described in section 3.2.1, except that the noise level was varied from 0 to 1.0 times the noise level specified previously in increments of 0.1. In each case 1000 samples were generated and PCA and PLS models, $\mathbf{R}_t$ and $\mathbf{R}_{pls}$, were formed. In each case the PCA model retained 5 PCs. The number of latent variables in each of the PLS models was optimized based on prediction error. This was determined from a cross validation where the calibration data set was randomly split into calibration and test sets of 500 samples 5 times. The predictive ability of the models was then tested on a new data set with the same noise level as the calibration set. Note that the only difference in the each of the calibration and test data sets was the noise level multiplication factor. Identical input and noise sequences were used.

The results of a subset of the prediction error tests are shown in Figure 4.1. The results are shown in terms of the percentage decrease in total sum of squares prediction error for the PLS model as compared to the PCA model, *e.g.*, for variable 7 and a noise level of 0.9 times the base noise level, the PLS model sum of squared error was almost 60% less than for the PCA model. Note how the difference in predictive ability of the models increases as the noise level is increased. It is also evident that there is a larger difference for some variables than for others.

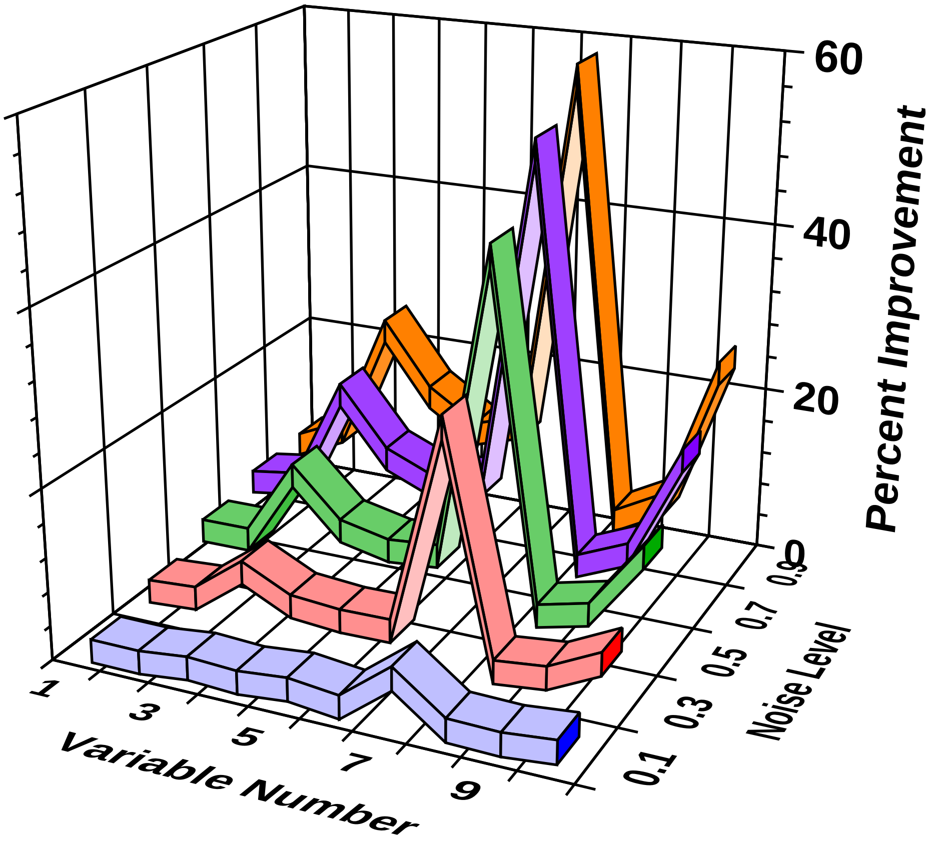

Figure 4.1. Percent Improvement in Sum of Squared Prediction Error for PLS Models Over PCA Models.

The results of this experiment show why PLS based residuals might be superior to those based on PCA. With the PCA model transformed in this manner, the change in a residual given a change in a variable due to error is identical for both the PCA and PLS models. However, the PLS model residuals under normal conditions are substantially smaller than the PCA residuals. Therefore, for the PLS model the change due to error is relatively larger, and should be more easily detected as unusual.

Comparison of the models $\mathbf{R}_{\mathrm{t}}$ and $\mathbf{R}_{\mathrm{pls}}$ themselves showed some interesting trends. When the noise level is zero the models are identical, as might be expected. As the noise level is increased, $\mathbf{R}_{\mathrm{t}}$ changes very little. This is consistent with experience from the previous chapter where it was shown that it is possible to identify an accurate PCA model even when the noise level is quite high. $\mathbf{R}_{\mathrm{pls}}$, on the other hand, changes a great deal as the noise becomes very high. For the optimum predictive ability, the model requires fewer latent variables, as would be expected. In general, the coefficients of the model tend to get smaller, though for some variables they may get larger. This trend is indicative of the shift towards fewer latent variables, which tends to spread the predictive ability of the model over more variables rather than concentrating it on a few.

While it can be expected that the PLS model $\mathbf{R}_{\mathrm{pls}}$ will be more sensitive to changes in the process data, it cannot be expected that the residuals will behave as in the PCA case. In general, PLS models do not produce zero-mean residuals. Furthermore, to the extent that $\mathbf{R}_{\mathrm{pls}}$ lies outside of the subspace spanned by $\mathbf{R}$, it can be expected that some state information will be mapped into the residuals. Therefore, because the states are usually autocorrelated, the PLS residuals will usually contain some autocorrelation also. Furthermore, if the autocorrelation in the states changes (perhaps due to a change in input behavior or a disturbance), then the autocorrelation of the residuals would also be affected. Any change in the autocorrelation of the residuals would have the affect of making the control limits invalid. Thus, any PLS detection scheme based on a particular correlation structure in the process states would become invalid given a change in the state behavior.

It was pointed out in the preceding chapter that the assumption that any process has a finite number of states is clearly an approximation, although generally a useful one. Thus, it is an approximation to say that the data from any process (where measurements are made at different locations) is intrinsically rank deficient. The PLS monitoring method, on the other hand, makes no assumptions about the intrinsic rank of the process data and, therefore, about the order of the process producing it. Instead, the PLS models are built up individually and the criteria for the models is the predictive residual error. Thus there is no "cutoff" approximation as in the PCA models. It could be said, however, that in each PLS model the number of factors used for prediction is an estimate of the number of process states that are relevant in the prediction of each output.

4.2 Generating and Using PCA-Like Residuals with PLS

The PLS residuals can be used in the same fashion as the PCA residuals, but the calculation of the detection limits must be modified. Because PLS is a biased regression method, there is no reason to expect that the residuals will be mean zero and normally distributed, even when the process consists of deterministic variation and Gaussian noise (see, for example, Geladi 1986 or Hoskuldsson 1988). Thus, in this work the limits on the mean and variance of the residuals will be set in the same manner that limits are set on PCA residuals with autocorrelation as described in section 3.4.

Statistical limits for sum of squared residuals, Q, based on the PLS models can be calculated in a similar fashion to the Q limit for PCA models. In this case, however, the observed variance of the residuals for all of the variables is substituted for the eigenvalues in equations (2.12) to (2.14). This substitution becomes obvious when one realizes that the eigenvalues of a covariance matrix are equal to the variance in each of the directions of the eigenvalues. When a PCA model is used, the original degrees of freedom of the problem is reduced: the residuals can only go in the directions of the unused eigenvectors and the degrees of freedom is reduced from the original number of variables by the order of the PCA model used. In the PLS problem, however, there is no loss of degrees of freedom. The residuals are free to be in any direction in the original vector space. Thus the total PLS Q residual is simply the sum of n independent squared variables, and the limits as obtained by Jensen and Solomon (1972) can be used. It would be expected that the deviation of the residuals from normal could degrade the accuracy of the calculated Q limits. However, Jensen and Solomon note that the behavior of Q approximately normal even when the distribution of the underlying variables (in this case the residuals) vary significantly from normal. This is a result of Q being the sum of many variables.

4.3 Comparison of PCA and PLS Monitoring

In the following sections PCA and PLS monitoring are compared using both synthetic and actual process data. The examples given here are representative of the results observed over many similar tests. For the synthetic example, the process given in section 3.2.1 will be used. Data from the West Valley SF-11 LFCM test will be used for the second example.

4.3.1 Examples with Synthetic Data

In this example the process with 10 variables but a true rank of 5 used in section 3.2.1 will be considered. The same output data used to calibrate the PCA model is used to generate a matrix of PLS models as noted in Chapter 2. The number of latent variables in each model of the matrix was determined through cross validation. The 1000 sample data set was split randomly into two 500 sample data sets five times for each variable. One of the 500 point segments was used to generate a PLS model while the other segment was used as the test set. The number of latent variables at the minimum PRESS for the sum of the 5 trials was chosen for each variable. The (10 by 10) matrix used for calculating the residuals of the PLS models, $\mathbf{I} - \mathbf{M}_p$, is given in Table 4.1, along with the number of latent variables retained in each of the models.

The effectiveness of the PLS models for removing autocorrelation from the system output was tested by checking the autocorrelation of the calibration set residuals, as shown in Figure 4.2. The ACF of the outputs is shown as the solid lines; the ACF of the PLS model residuals is shown as the dashed lines. Note that while there is less autocorrelation in the residuals than in the outputs themselves, there is still a significant amount. This is more apparent in Figure 4.3, which shows the ACF of the outputs and residuals from the 1000 sample test data set generated from a PRBS which used to test the PCA model in section 3.2.1. Unlike the PCA model residuals in Figure 3.1, the PLS residuals are correlated in time. A test was also performed to see if the a PLS model designed specifically with the PRBS generated data set would successfully remove the autocorrelation of the set. The results of this test are shown in Figure 4.4, where it is apparent that the residuals are much less autocorrelated than in Figure 4.2. It is apparent from these tests that, because of the mapping of some state information into the residuals, PLS models are more effective on data sets with structure similar to the calibration data.

Table 4.1. Matrix for Calculating Residuals of PLS Models with Number of Latent Variables in each Model in Square Brackets.

Columns 1 through 5

[2]	[4]	[2]	[3]	[5]
1.0000	-0.0673	0.0353	-0.2572	-0.0041
-0.0711	1.0000	0.1004	-0.1207	0.1587
0.0300	0.1113	1.0000	0.0803	-0.3397
-0.2559	-0.1180	0.0978	1.0000	-0.1968
-0.0067	0.1570	-0.3556	-0.1978	1.0000
-0.0122	-0.0935	-0.1214	0.3445	0.0944
-0.0654	0.1245	0.1002	-0.0151	-0.0864
-0.1399	-0.2474	0.0761	-0.0470	0.0065
0.0552	-0.2468	-0.1041	-0.1578	-0.1871
-0.3800	0.0001	-0.1479	-0.0372	-0.0250

Columns 6 through 10

[2]	[3]	[3]	[2]	[2]
-0.0272	-0.0843	-0.1319	0.0558	-0.4007
-0.0961	0.1636	-0.2374	-0.2495	-0.0007
-0.1191	0.1054	0.0518	-0.0876	-0.1379
0.3595	-0.0325	-0.0449	-0.1569	-0.0382
0.0774	-0.1063	0.0049	-0.1863	-0.0272
1.0000	-0.1639	-0.2674	-0.0162	-0.0584
-0.1399	1.0000	-0.1115	-0.0272	0.0441
-0.3047	-0.1367	1.0000	-0.2725	-0.0483
-0.0075	-0.0280	-0.2561	1.0000	0.1899
-0.0564	0.0584	-0.0441	0.1793	1.0000

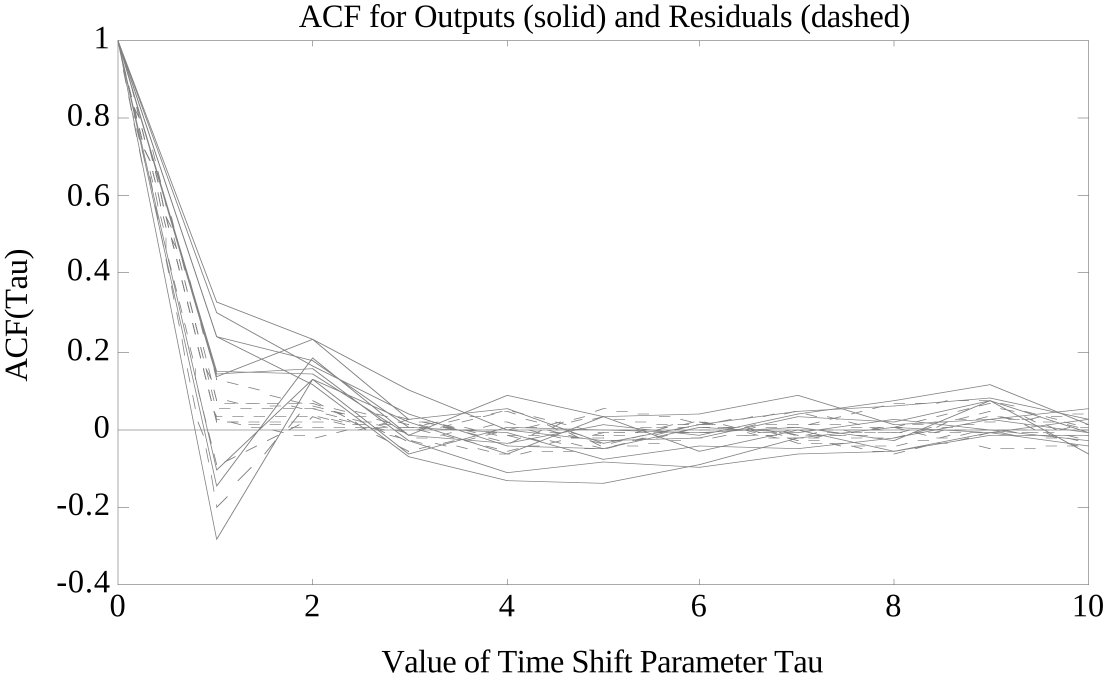

Figure 4.2. Autocorrelation Function of Process Outputs and PLS Residuals from Data Set with White Noise Input.

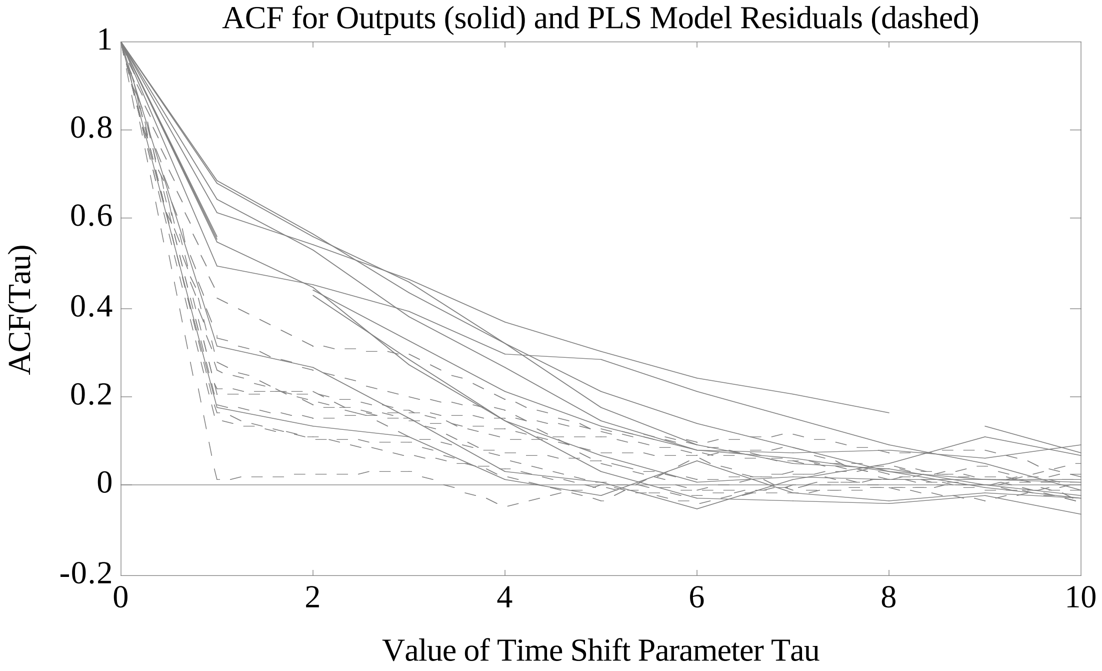

Figure 4.3. Autocorrelation Function of Process Outputs and PLS Residuals from Data Set with PRBS Input.

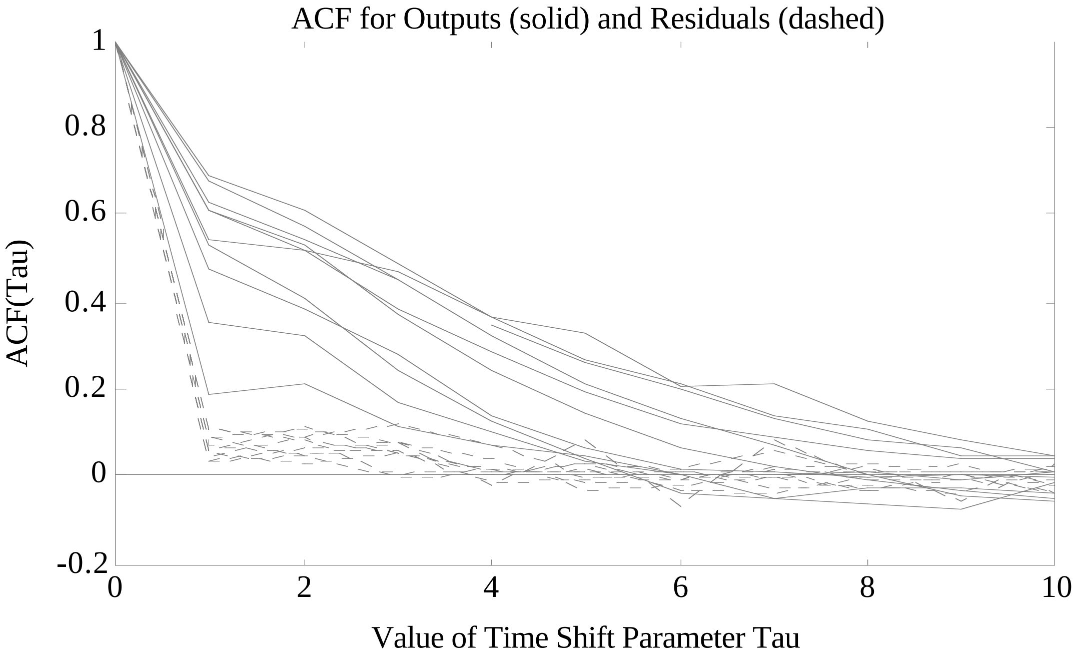

Figure 4.4. Autocorrelation Function of Process Outputs and PLS Residuals Based on PRBS Data from Data Set with PRBS Input.

The detection limits for the PLS models were calculated based on the observed residuals in the calibration data set. As with the PCA limits, the PLS limits were calculated assuming a window of 20 samples would be used. The limits were chosen[8] such that 99% of the 20 sample groups selected from the calibration data set would have mean and standard deviation within the limits. The resulting detection limits are shown in Figure 4.5 below as the dashed lines, and are compared with the corresponding limits for the PCA model shown as solid lines. Note that in all cases the PLS model detection limits are as good as or better than the PCA model limits. The largest differences are on variables 3 and 7, the variables with the worst detection limits in the PCA model.

In practice, with both PCA and PLS models, when a sensor fails it often causes other variables besides itself to go outside the calculated confidence limits. This is demonstrated

---

[8] As a test, the same procedure for calculating detection limits for the PLS models was used to calculate detection limits for the PCA model. These limits were found to agree very well with the PCA model limits calculated directly from theory.

in Figure 4.6, which shows the limits for a PCA residual mean change (to detect a bias error) and the observed PCA residual means. In this case the bias was added to variable 2, but both variables 2 and 7 have gone outside the defined limits. Variable 7 is actually further outside the limit in an absolute sense. The biased variable can be detected by determining the ratio of the observed residual mean to the limits, as shown in Figure 4.7. Here it is clear that variable 2 has the larger relative bias and is therefore correctly identified as the biased variable.

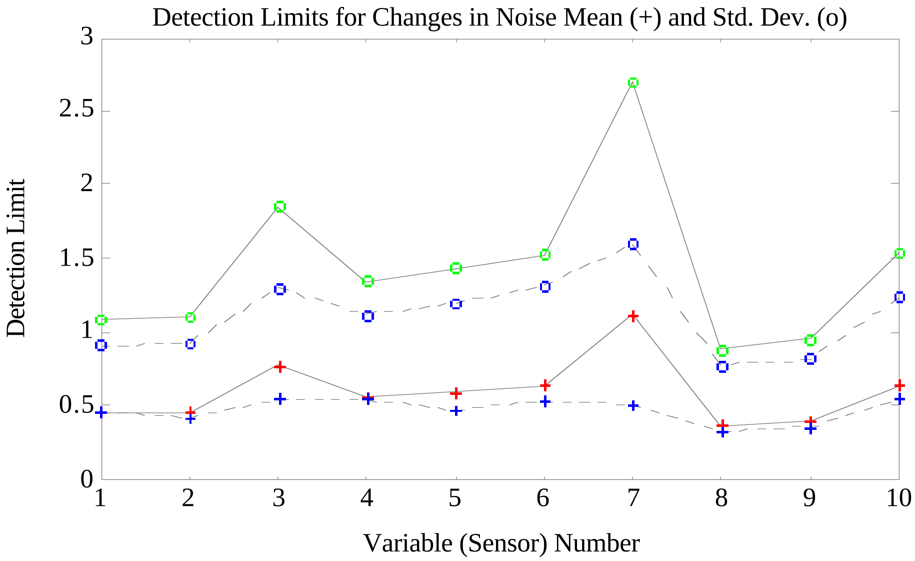

Figure 4.5. Detection Limits for Changes in Measurement Noise Mean and Standard Deviation for PCA (solid lines) and PLS Models (dashed lines).

The ability of the PCA and PLS models to accurately identify sensor failures (measurement bias and additional measurement noise) was tested through simulation. Two new 1000 sample data sets were generated using the model from section 3.2.1. In the first case the model was driven by white noise (as in the calibration set) and in the second case the model was driven by a PRBS, which has considerably more power at low frequencies. The data sets were broken into 50 segments of 20 points each Noise or bias was then added to each variable in the segment in turn, and the residuals were calculated and tested for significance. The "failed sensors" were identified based on the ratio of the residuals mean or standard deviation to the appropriate limits, as specified above[9].

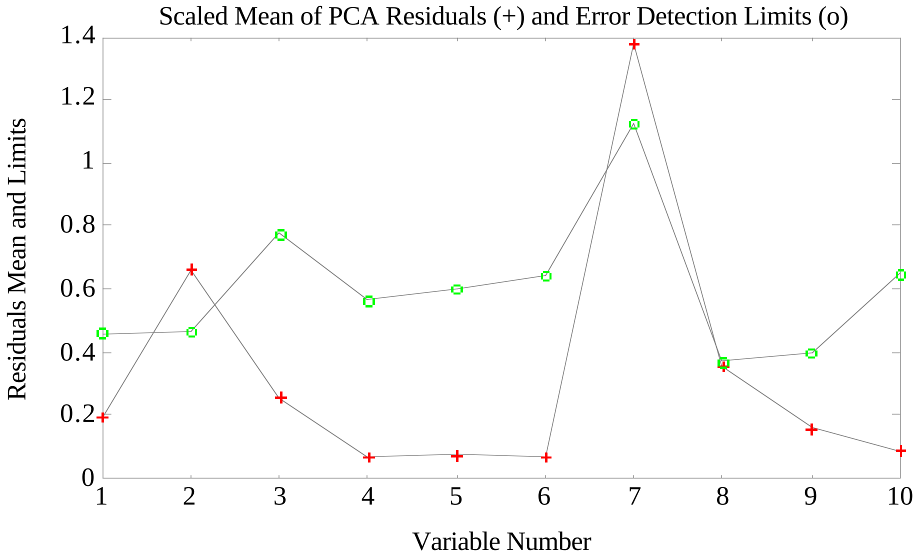

Figure 4.6. Scaled Mean of PCA Residuals and Bias Error Detection Limits Showing Apparent Bias on Variables 2 and 7.

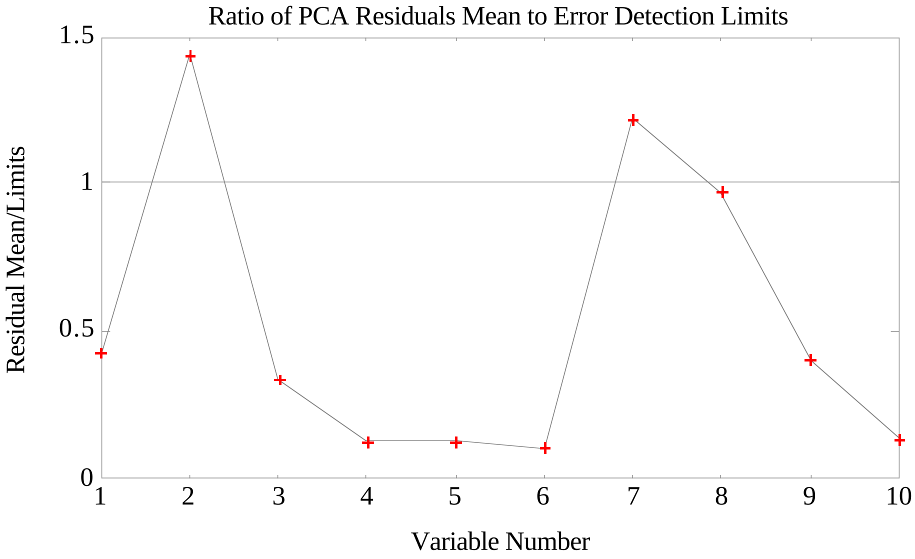

Figure 4.7. Ratio of Mean of PCA Residuals to Bias Error Detection Limits Showing Largest Bias on Variable 2.

---

[9] The results of the T^2 test in these simulations are given in Appendix 3. In general, the T^2 test did not perform as well at the t- and F-tests used here.

The results of the error detection simulation are shown in Table 4.2 for bias error detection and Table 4.3 for noise error detection. In each case the table is divided into two sections corresponding to the case of white noise input to the model or the PRBS sequence which is autocorrelated. The results for the PCA model are the four columns on the left while the PLS model results are the four columns on the right. Each of the four columns corresponds to a different level of added bias or noise. The basis for the bias and noise levels is the scaled outputs, *i.e.*, a bias of 0.5 indicates that a bias of 0.5 units was added to the scaled variable. The original scaling, which resulted in a mean zero unit variance calibration set, was used in the test. A noise error of 1.0 corresponds to adding white noise of unit standard deviation to the scaled outputs. The row labeled "Correct" indicates the number of times the method correctly identified the proper variable has having added bias or noise. "No Response" indicates the number of times there were no variables over the limits. "Wrong" indicates the number of times an out of bounds variables was detected but the wrong variable was indicated as faulty.

Table 4.2. Error Detection Simulation Results for Bias Errors.

Model Input White Noise

	PCA Model				PLS Model			
Bias Size	0.5	1.0	1.5	2.0	0.5	1.0	1.5	2.0
Correct Response	141	423	485	499	229	485	499	500
No Response	286	29	1	0	187	4	0	0
Wrong Response	73	48	14	1	84	11	1	0
WR + NR	359	77	15	1	271	15	1	0

Model Input Correlated PRBS

	PCA Model				PLS Model			
Bias Size	0.5	1.0	1.5	2.0	0.5	1.0	1.5	2.0
Correct Response	153	438	492	500	186	412	473	492
No Response	298	24	1	0	89	2	0	0
Wrong Response	49	38	7	0	225	86	27	8
WR + NR	347	62	8	0	314	88	27	8

There are several trends in the simulation results which should be noted. In general, the PLS model has considerably more correct responses than the PCA model, particularly in the case of white noise input to the process. Furthermore, even when the PCA model has more "no response" indications than the PLS model, it often has more "wrong responses." This can be taken as a indication that it is not just that the PCA limits are too large. If this were the only difference, it would be expected that the number of wrong PCA responses would be smaller than for PLS models when the number of no responses is larger.

Table 4.3. Error Detection Simulation Results for Noise Errors.

Model Input White Noise

	PCA Model				PLS Model			
Noise Std. Dev.	1.0	2.0	3.0	4.0	1.0	2.0	3.0	4.0
Correct Response	189	482	499	500	340	500	500	500
No Response	277	11	0	0	146	0	0	0
Wrong Response	34	7	1	0	14	0	0	0
WR + NR	311	18	1	0	160	0	0	0

Model Input Correlated PRBS

	PCA Model				PLS Model			
Noise Std. Dev.	1.0	2.0	3.0	4.0	1.0	2.0	3.0	4.0
Correct Response	204	485	498	500	348	499	500	500
No Response	257	7	1	0	97	0	0	0
Wrong Response	39	8	1	0	55	1	0	0
WR + NR	296	15	2	0	152	1	0	0

In Table 4.2 it is also apparent that the autocorrelation in the inputs has a large negative effect on the PLS model, but little effect on the PCA model. It is not surprising that there is little effect on the PCA model because the residuals are known to be uncorrelated regardless of the process input. The PLS residuals, on the other hand, are known to autocorrelated when the autocorrelated input is used, as shown in Figure 4.1. It is also not surprising that this autocorrelation in the PLS residuals affects the ability to properly detect bias error more than it affects the ability to detect noise errors. Autocorrelation has a greater effect on attempts to measure a change in the mean than on attempts to detect additional noise.

The number of false alarms (indications that the mean or variance had gone over the designated limit when no bias or noise had been added) was also considered in these tests, though it is not included in Tables 4.2 and 4.3. When the input signal was white noise, the number of false alarms was as expected for both PCA and PLS models, *i.e.*, each variable alarmed approximately 1% of the time. When the PRBS sequence was used, the PCA model false alarm rate stayed approximately constant. The PLS model false alarm rate, however, changed substantially, as might be expected. Increased autocorrelation in the residuals caused the bias test false alarm rate to go up to ~10%.

4.3.2 Examples with LFCM Data

In this section PCA and PLS monitoring techniques are tested using West Valley LFCM data from run SF-11. For these examples a 1500 sample segment of the run was selected. After some editing of the data set to replace outliers, the first 750 samples were used for calibration and the last 750 samples were used for the test set. A matrix of tests were performed in order to cover the effects of model order and noise or bias level. PCA generated from both mean centering and autoscaling were tested, the PLS models were all from autoscaled data. The order of the PCA models was varied from 1 to 10 PCs retained. PLS models matrices were formed where the number of latent variables in each of the individual models was fixed at the same value from 1 to 10. In addition, several PLS models were formed using different numbers of latent variables for each model, where the criteria for the number of latent variables was based on prediction error.

Because the models did not, in general, produce non-correlated residuals, the model limits were set based on the observed 99% limits of the calibration set. All of the models were tested for their ability to detect both biases and added noise in the test data set and for the number of false alarms produced. The number of "no response" indications (no variables detect as "bad"), "correct response" indications ("bad" variable correctly identified) and "incorrect response" indications (variables identified as bad but incorrectly specified) were recorded.

A major trend in the simulations is that as model order is increased, its sensitivity goes up but its specificity goes down. This is indicated in Figures 4.8 to 4.10, where the number of correct model responses to a bias error is plotted for each of the sets of autoscaled PCA models, mean centered PCA models and PLS models. Bias errors equal to 0.5 to 2.0 standard deviations of the calibration data set were tested. Note that the bias errors as a fraction of the detection limits are different for each variable, so it would not be expected that all variables would have a similar number of correct detections. The total possible number of correct responses is 740 (20 variables times 37 independent data sets).

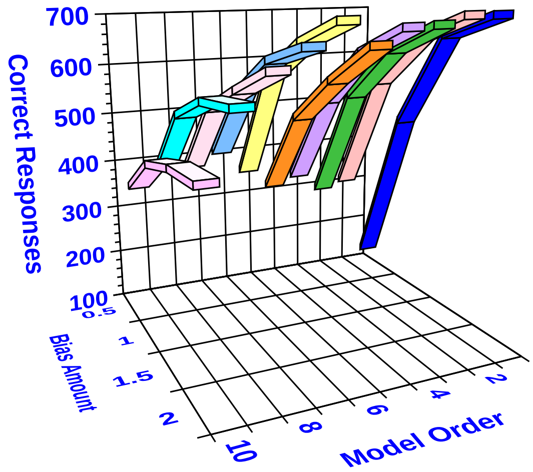

Figure 4.8. Number of Correct Responses to Bias Test for Autoscaled PCA Model.

In each of the figures it is apparent that increasing the model order has the effect of increasing the number of correct responses to small biases while decreasing the number of correct responses to large biases. The trend towards decreased specificity with higher order is greatest for the mean centered PCA model. This is expected because the variables with the largest variance in the mean centered model will tend to have very large loadings. Residuals of variables with very large loadings will tend to be more heavily influenced by other variables than by the variable itself, *i.e.* the diagonal elements in the $\mathbf{I} - \mathbf{PP}^T$ matrix will get very small for variables with high loadings thus decreasing sensitivity to changes in that variable.

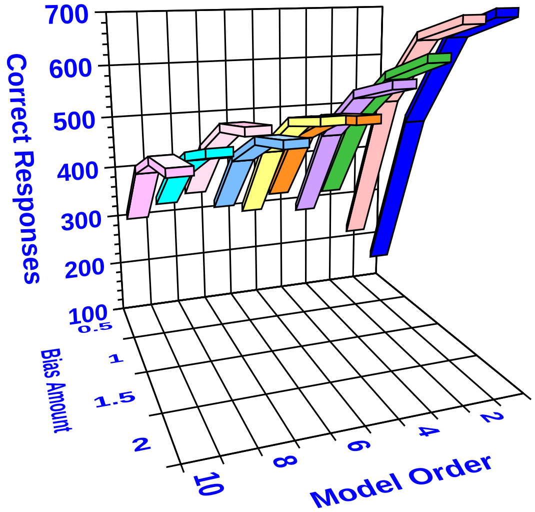

Figure 4.9. Number of Correct Responses to Bias Test for Mean Centered Model.

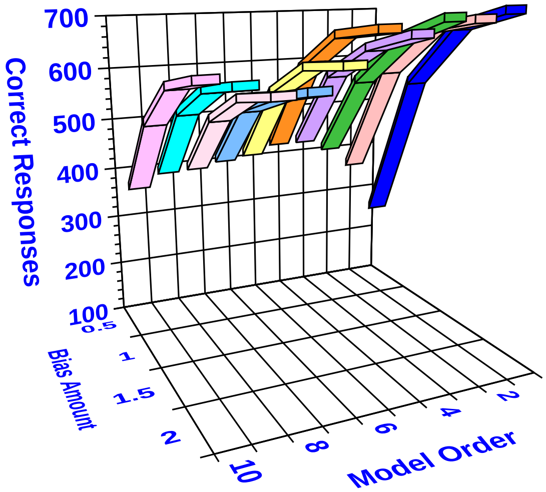

Figure 4.10. Number of Correct Responses to Bias Test for PLS Model.

The number of "no response" indications for the autoscaled PCA model, mean centered PCA model and PLS model are shown in Figures 4.11 to 4.13, respectively. (Note that the responses are shown as a function of model order rather than bias amount as in the previous figures.) It is apparent in all three of these figures that there are very few "no response" indications to large errors, as would be expected. It is also evident that, for the PCA models, the total number of no responses goes through a minimum for an intermediate model order. This is expected because as the model order increases, the residual space is made progressively smaller.

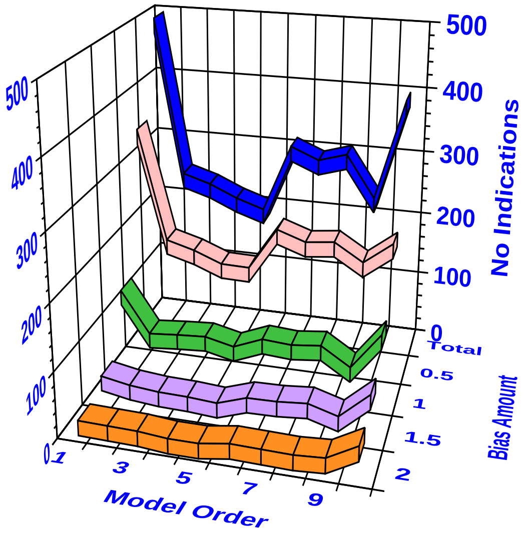

Figure 4.11. Number of No Indications to Bias Test for Autoscaled PCA Model.

The fact that the number of no indications goes through multiple minima is also expected. The largest effect of an additional PC may be to increase the sensitivity of the model for a particular variable which is not loaded strongly into the model, or it may decrease the sensitivity of the model to a variable which is already strongly loaded. Eventually, of course, the sensitivity of the model to changes in all variables will decline as more PCs are added. This is not true for the PLS models, as indicated by Figure 4.13. There is no reason to expect a large increase in the number of "no response" indications. Some small increases should occur, however, due to the effect that added latent variables have on the prediction error. As the prediction error for a variable increases, the sensitivity of the model to changes in that variable decreases.

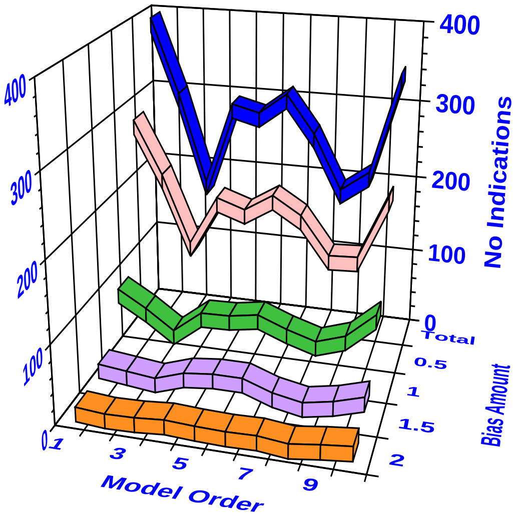

Figure 4.12. Number of No Indications to Bias Test for Mean Centered PCA Model.

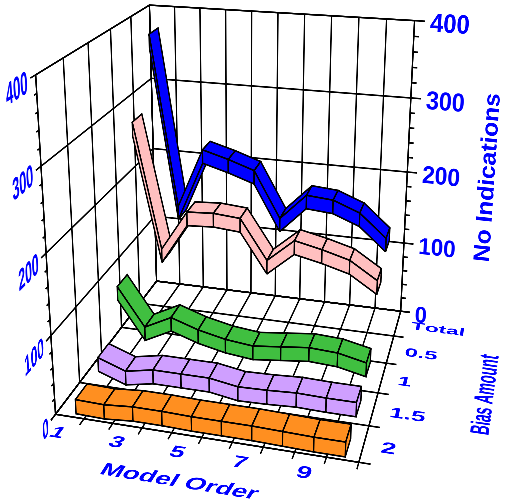

Figure 4.13. Number of No Indications to Bias Test for PLS Model.

The number of incorrect responses of the PCA and PLS models are shown in Figures 4.14 to 4.16. As expected, the number of incorrect responses tends to increase in all models as the model order increases. Note that there is little change in the number of incorrect responses to small errors but the number of incorrect responses to large errors increases substantially as the model order increases.

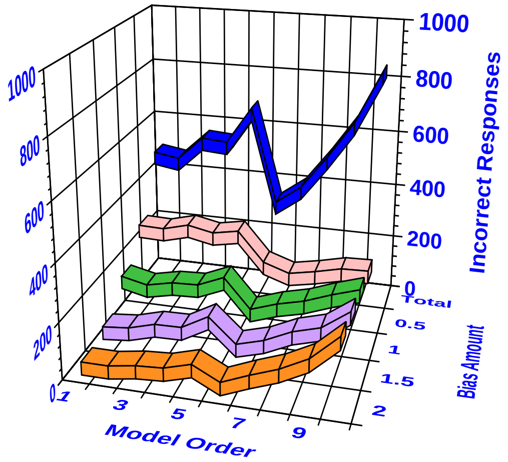

Figure 4.14. Number of Incorrect Responses to Bias Test for Autoscaled PCA Model.

The number of false alarms remaining approximately constant for the tests, at about 1% for each variable. However, the false alarm rate tended to increase with increase in model order. The apparent reason for this is that, as the model order is increased, the model is trying to approximate the true process to progressively smaller tolerances. Therefore, slight differences in the test data relative to the calibration data result in more false alarms.

The real advantage of PLS monitoring over PCA lies in the fact that models of "mixed order" can be constructed, *i.e.* models that have different numbers of latent variables for the prediction of each variable. The predictive ability of the model can then be optimized for each variable. Experiments have shown that there are some practical difficulties with this approach. The major one is that, for some cases, as the prediction errors of the variables decrease and the sensitivity of the model increases, its specificity goes down. Small errors become more easily detected, but the assignment of the errors to the correct variable becomes hampered.

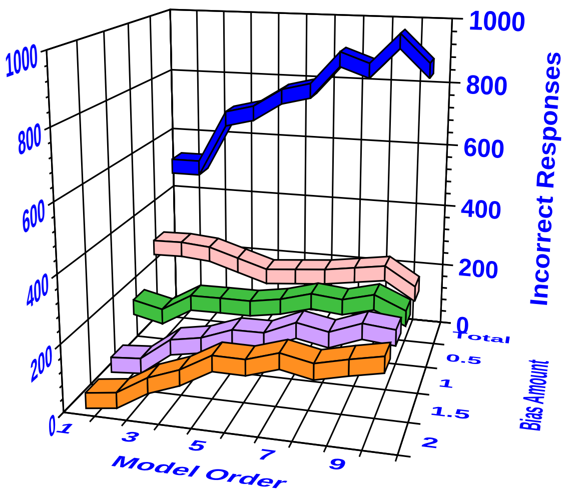

Figure 4.15. Number of Incorrect Responses to Bias Test for Mean Centered PCA Model.

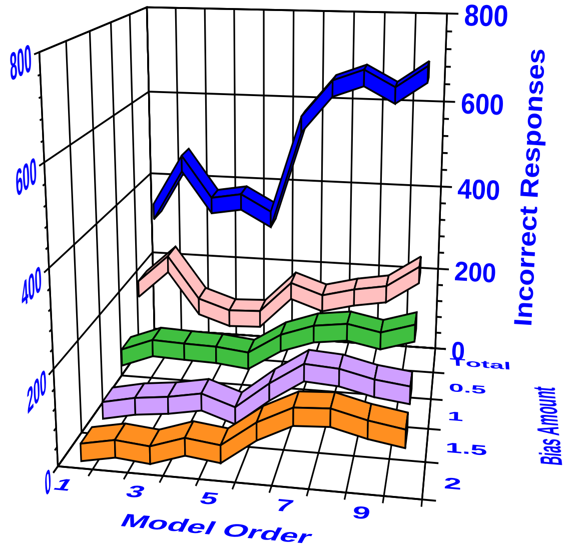

Figure 4.16. Number of Incorrect Responses to Bias Test for PLS Model.

The cause of the trade off between sensitivity and specificity becomes apparent when one considers how the coefficients in PLS models change as more latent variables are added to the model. PLS models with few latent variables tend to have smaller coefficients but more variables with significant coefficients, *i.e.* the prediction is spread out over more variables. As latent variables are added to the model the coefficients on the variables that are the most correlated with the variable to be predicted tend to get larger at the expense of the coefficients on the remaining variables. Because of this, low order PLS models are more robust to changes in single sensors, and especially to changes in the sensors which are most correlated with the predicted variable. Thus, it is found that when PLS models with many LVs are used in the error detection problem, when a sensor fails it is not only the "bad" variable that goes out of bounds but also all highly correlated variables. Often several variables go over the limits and by similar amounts. This is because the correlated variables rely heavily on the bad variable for prediction.

In the experiments performed here it was found that the collection of PLS models where each model was optimized for prediction was not a particularly effective monitoring tool. While the detection limits of this model are indeed optimum in some sense, the specificity of the model is not good due to the heavy reliance of some of the models on too few variables. By reducing the number of latent variables in the PLS models it was possible to construct a model with very good sensitivity and selectivity. In this work this was done in a rather unsystematic fashion, but this need not be the case. It would certainly be possible to use an optimization technique, such as non-linear programming, to solve for the optimum collection of models for error detection.

As a final comparison here, the best of the PLS and PCA models were compared to each other for their ability to detect biased variables. In this case "optimum" was defined as the model having the highest ratio of correct responses to the sum of the wrong and no responses. This is a subjective criteria, and depending upon the application the weights given to the wrong and no response categories might be different. The optimum model from the autoscaled PCA models is the 6 PC model, while for the mean centered data it is the 1 PC model. The optimum PLS model of fixed order is the 5 latent variable model. In addition, a "near optimum" PLS model of mixed order was also tested. The results of these tests are shown in Figures 4.17 to 4.19.

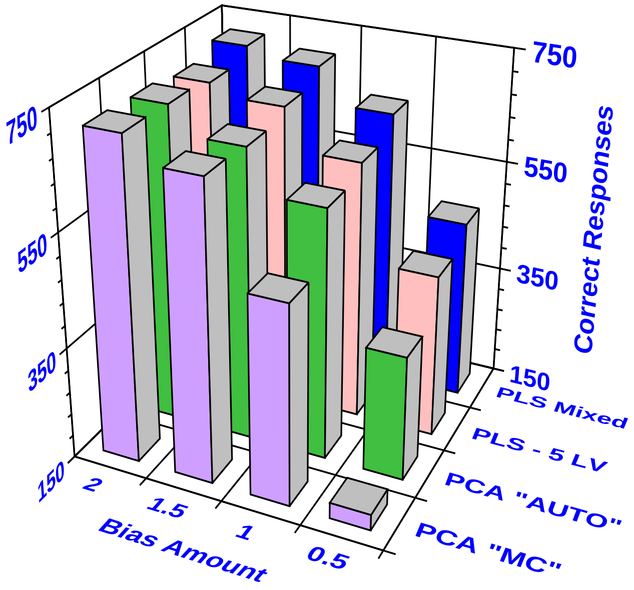

Figure 4.17. Number of Correct Responses to Bias Test for "Best" Models.

Figure 4.17 shows that, while all the models have a similar number of correct responses to large errors, the mixed PLS and 5 LV PLS models are considerably better at detecting the smaller errors. In absolute number of correct responses, the mixed PLS model was better at all bias levels tested.

Figure 4.18 shows that the mixed PLS model also had by far the fewest number of incorrect responses. Here the differences are relatively large between the mixed model and the others, particularly for small biases. Finally, Figure 4.19 shows that the mixed model is not quite as sensitive as the 5 LV model, but more sensitive than either of the PCA models.

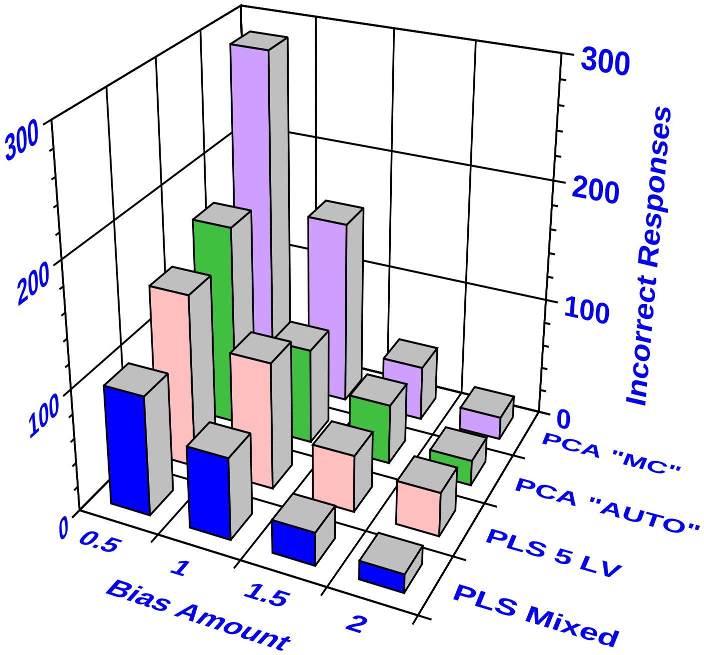

Figure 4.18. Number of Incorrect Responses to Bias Test for "Best" Models.

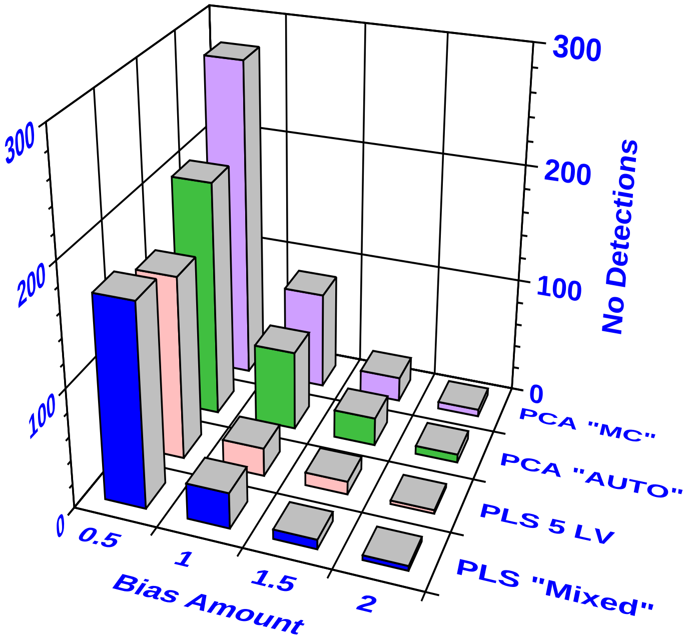

Figure 4.19. Number of No Responses to Bias Test for "Best" Models.

The calculated detection limits based on the observed residuals of the calibration set are shown for all the models in Figures 4.20 (for bias errors) and 4.21 (for noise errors). As expected, the detection limits are not the same for all variables. Thermocouples deep within the glass melt (variables 1-6 and 11-16) have much smaller limits than than those in the cold cap (variables 7-9 and 17-19) and plenum regions (variables 10 and 20).

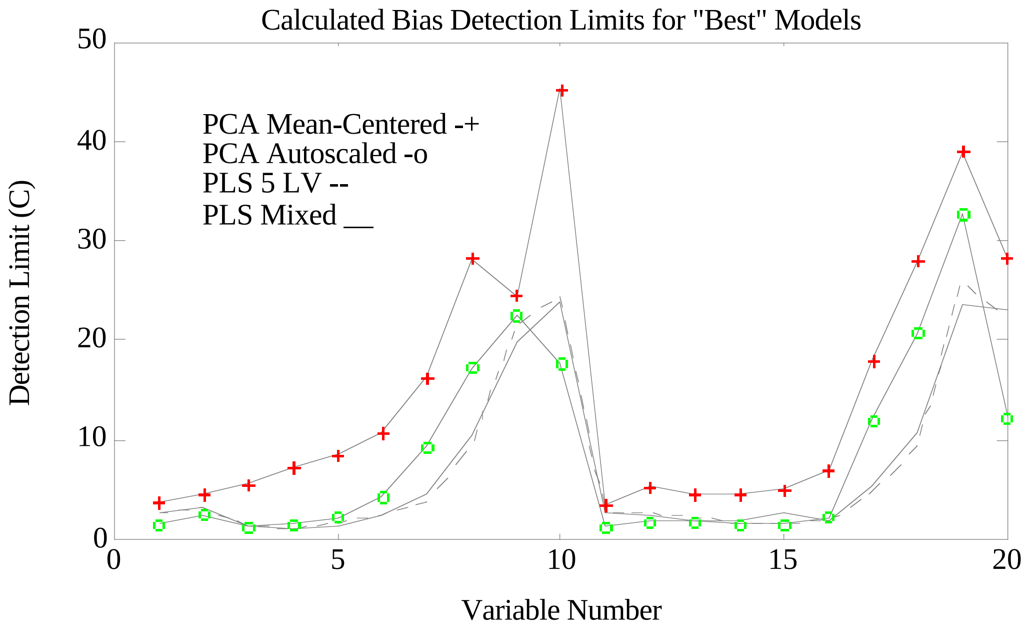

Figure 4.20. Calculated Bias Detection Limits for "Best" Models Based on Observed Residuals in Calibration Set.

The detection limits for the autoscaled PCA model are clearly better than for the mean-centered model. The detection limits for the PLS models are very similar to each other. For some variables the PLS models have significantly better limits than the autoscaled PCA model, while for other the limits are similar or slightly worse. It is known from the simulations, however, that the PCA model suffers from a lack of specificity. It would be expected that, while the PCA model might detect the presence of a small error that the PLS models would not detect, it is unlikely that the PCA model could properly assign the error to the correct variable.

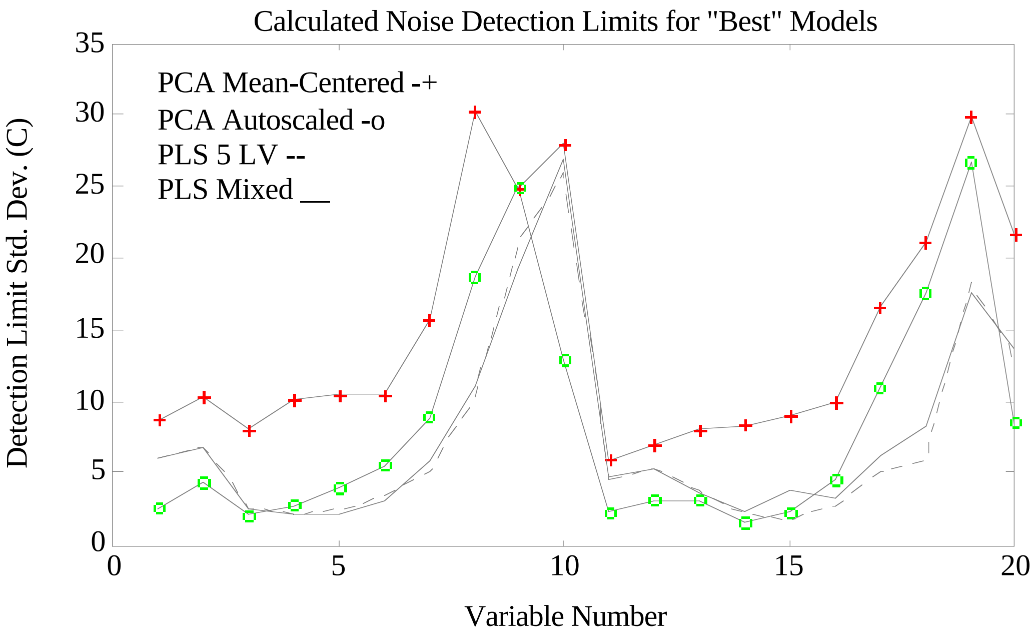

Figure 4.21. Calculated Noise Detection Limits for "Best" Models Based on Observed Residuals in Calibration Set.

4.4 Robust PLS for Multiple Failures

In the previous chapter, a method for predicting outputs for known “bad” sensors was derived. The idea was to determine the value of the bad output that minimized Q. The solution to this problem was easy partly due the symmetry of the PCA matrix for calculating residuals. In the case of generating PLS-based residuals, however, the symmetry does not exist and the problem becomes just slightly more difficult.

Assume that the matrix for calculating the residuals of the collection of PLS models defined in equation (2.59), $\mathbf{R}_{pls}$, can be partitioned in a manner similar to the $\mathbf{R}_m$ matrix shown in equation (3.22). The value of the Q statistic for the PLS models can then be written:

$$Q_{pls} = \mathbf{x}_b\mathbf{R}_{11}\mathbf{x}_b^T + \mathbf{x}_g\mathbf{R}_{21}\mathbf{x}_b^T + \mathbf{x}_b\mathbf{R}_{21}^T\mathbf{x}_g^T + \mathbf{x}_g\mathbf{R}_{22}\mathbf{x}_g^T \qquad (4.8)$$

Once again, it is possible to solve for the values of $\mathbf{x}_b$ which minimizes Q_{pls}. The lack of symmetry causes a minor complication, however, the result is

$$\mathbf{x}_b = -\mathbf{x}_g((\mathbf{R}_{21} + \mathbf{R}_{12}^T)/2)\mathbf{R}_{11}^{-1} \qquad (4.9)$$

It is interesting to note that, while the residuals on the bad variables becomes zero in the PCA case, it does not in the PLS case. Instead, the new residuals on the bad variables are:

$$\mathbf{r}_b = -\mathbf{x}_g\mathbf{R}_{ave} + \mathbf{x}_g\mathbf{R}_{21} \qquad (4.10)$$

where

$$\mathbf{R}_{ave} = (\mathbf{R}_{21} + \mathbf{R}_{12}^T)/2 \qquad (4.11)$$

So to the extent that $\mathbf{R}_{21}$ is not equal to $\mathbf{R}_{12}^T$ the residuals on the fixed variables will not be zero. The overall process residual Q_{PLS} can be calculated by substituting the solution for $\mathbf{x}_b$ in equation (4.9) into equation (4.8), however, this expression does not appear to simplify very much, so this exercise will not be repeated here.

4.5 Conclusions Concerning PLS and PCA Monitoring

In this chapter it has been shown that PLS can be used in a manner similar to PCA for monitoring multivariate processes. It is interesting to note that, unlike PCA, when PLS is applied in this manner uncorrelated residuals do not result, even in the ideal case of a linear process with more outputs than states. It is clear that to the extent that the collection of PLS regression vectors lies outside the subspace spanned by $\mathbf{I} - \mathbf{PP}^T$ the resulting model will produce residuals that are autocorrelated. This is because state information will be mapped into the residuals from the PLS models under these circumstances. In spite of this, the better prediction error of the PLS models allow for improved error detection under most circumstances. However, if the correlation in the model states changes significantly the

PLS model performance is expected to suffer.

The effect of model order was also investigated. It was found that higher order models, while more sensitive, tended to be less accurate in identifying the perturbed variable. In the PLS models, this trade off can be optimized for each variable. This typically results in models with fewer latent variables than would be expected based on a prediction error criteria alone. These mixed order PLS models achieve the best error detection performance of any of the methods tested.

5.0 Dynamic Process Model Identification with CR

This chapter concerns the effects of using continuum regression for identification of dynamic process models. There are many questions to be answered concerning this application of CR. The most important, of course, is whether it is better than existing methods. This chapter will demonstrate that for the case of FIR models CR is clearly more effective than existing methods. For ARX models, CR offers real advantages only in a limited number of situations. There are, however, other issues to consider and some potential pitfalls of the method that should be addressed. Specifically, the following issues are considered: What are the biases that the method introduces and how serious are they? Are there particular process dynamics that are difficult to capture using the CR method? What are the effects of data set size, noise level and degree of input excitation on the optimum model location in the CR model space? All of these questions will be considered in the sections that follow.

5.1 General Methodology

The questions posed above are attacked from several angles. The problem is first broken down into the types of models to be identified. Linear FIR models are considered first in section 5.2, then linear ARX models in section 5.3. A final section, 5.4, will deal with non-linear FIR models.

Within the model types a number of issues are considered. For FIR models, the problem of the frequency domain effects of PCR is considered from a theoretical viewpoint. Because PCR is a limiting case of CR, an understanding of it is useful for interpreting CR results. Other questions concerning linear FIR and ARX model identification are attacked using numerical simulations. A set of representative process models is selected, then these models are used to generate calibration data sets of various sizes with different amounts of noise and levels of input excitation. Models are then identified using the simulated outputs and compared to the original models. Finally, the potential for using PLS and PCR for non-linear FIR model identification is explored with an example employing data from a real process.

5.2 Continuum Regression for FIR Model Identification

The goal of this chapter is to develop an understanding of how the CR method performs when used for the identification of FIR models and identify and any potential problems concerning its application. One concern in particular involves the creation of "artifacts" in the identified models. In the following section the behavior of PCR, one extreme of the CR method, will be considered. Specifically, the PCA decomposition of the input block will be explored, and this decomposition will be related to the frequency domain behavior of the resulting PCR models. The reason for studying PCR is that, once the effects of PCR are understood, the effects of the CR method will be much clearer. The effects of process noise, input excitation and process dynamics on the FIR models obtained by CR will then be considered in later sections.

5.2.1 Eigenvector Decomposition of the ACM

The first step in understanding how PCR works when used for FIR identification is to review the form of the input matrix that will be decomposed by PCA. When the data are arranged for identification of an FIR model, the resulting matrices resemble those shown in Figure 5.1 below. Here the first 4 samples are shown for identification of a 5 coefficient FIR model. Note how the values of the input u(t) in the **X** matrix are repeated along diagonals. As pointed out in Box and Jenkins (p. 53), the correlation matrix of this **X**, [**X'X**/(n - 1)], approaches the Autocorrelation (or Autocovariance depending on the scaling of the original input signal) Matrix (ACM) as more data are collected. Each entry in the Autocorrelation (Autocovariance) Matrix, a_{ij}, is equal to the correlation coefficient (covariance) between u(t+j) and u(t+i). Note, however, that the correlation between any u(t+j) and u(t+i) depends only on the difference between i and j. This can be compared to the Autocovariance Function (ACF), which is a vector of the covariance between a (mean centered) signal u at time t and time t-τ. Thus:

$$ACF(\tau) = E\{u(t)u(t-\tau)\} \tag{5.1}$$

where E{ } denotes the expectation operator. If u is scaled to unit variance, the ACF becomes the Autocorrelation Function. Because every value in the ACM depends only upon the difference of the indices, every entry a_{ij} in the ACM can be replaced by ACF(i-j). This is shown in Figure 5.2, where each value in the ACM is replaced by the corresponding value in the autocorrelation function (ACF). Furthermore,

		X			**y**
u(5)	u(4)	u(3)	u(2)	u(1)	y(6)
u(6)	u(5)	u(4)	u(3)	u(2)	y(7)
u(7)	u(6)	u(5)	u(4)	u(3)	y(8)
u(8)	u(7)	u(6)	u(5)	u(4)	y(9)
:	:	:	:	:	:

Figure 5.1. Arrangement of Data for FIR Model Identification.

		X'X		
ACF(0)	ACF(1)	ACF(2)	ACF(3)	ACF(4)
ACF(-1)	ACF(0)	ACF(1)	ACF(2)	ACF(3)
ACF(-2)	ACF(-1)	ACF(0)	ACF(1)	ACF(2)
ACF(-3)	ACF(-2)	ACF(-1)	ACF(0)	ACF(1)
ACF(-4)	ACF(-3)	ACF(-2)	ACF(-1)	ACF(0)

Figure 5.2. Arrangement of Autocorrelation Matrix for FIR Model Identification.

The ACM is special type of matrix known as a Toeplitz matrix, in which values are repeated along diagonals. The ACM is also symmetric because ACF(τ) = ACF(-τ). Furthermore, if the characteristics of the input signal u are known, the expected value of the ACM can be calculated. For instance, if u(k) is generated by passing a white noise signal c(k) through a first order filter

$$u(k) = \alpha\, u(k-1) + (1 - \alpha)\, c(k) \tag{5.2}$$

then the expected values of the entries in the autocorrelation matrix are

$$\text{ACM}\,(i,j) = \mathbf{a}_{ij} = (\alpha)^{|i-j|} \tag{5.3}$$

As mentioned above, the PCR method relies on an eigenvector decomposition of the covariance or correlation matrix, [$\mathbf{X'X}/(n - 1)$], which for FIR models is equal to the Autocovariance or Autocorrelation Matrix of the input signal. Typically, this leads to eigenvectors (PCs) that have coefficients that look like sine and cosine curves. Such a case is plotted below in Figure 5.3, which shows the coefficients of the first five eigenvectors of an ideal ACM (where the entries in the ACM are equal to the expected values for large data sets). This ACM was calculated for |i-j| up to ±100 resulting in an ACM matrix that is 101 by 101. Because of the equality of principal component vectors and eigenvectors of the corresponding covariance matrix, the plots are identical to plots of the entries in the **p** vectors from PCA. This particular ACM was calculated assuming that the input signal was generated by passing white noise through a first order filter as in (5.2) where $\alpha = 0.8$. There would be 101 coefficients in the FIR model corresponding to this ACM.

For the continuous (as the sample rate goes to zero) but finite case (where there is a maximum correlation time to be considered) it can be shown that the eigenvector coefficients of an ACM resulting from white noise through a first order filter are identically sine and cosine functions. In this case the autocorrelation matrix becomes a continuous function of two variables, which we will call x and y, over a finite domain. In the discrete case it is possible to multiply the ACM (a function of two indices) by a vector (a function of one index) to obtain a vector (also a function of one index). In an analogous manner it is possible to multiply the continuous ACM (a function of two variables x and y) by a function (in one variable, say x) and then integrate (over x) to obtain a function in one variable (y). Let us define the continuous ACM over the interval -a to a in both the x and y directions. It will be shown that the width of the interval is proportional to the FIR window width that has been chosen. To further simply things, assume that the continuous

ACM can be defined as

$$\mathrm{ACM}(x,y) = e^{-|x+y|} \tag{5.4}$$

Here e is used because the resulting mathematics are simplified. It will be shown that this does not affect the generality of the solution.

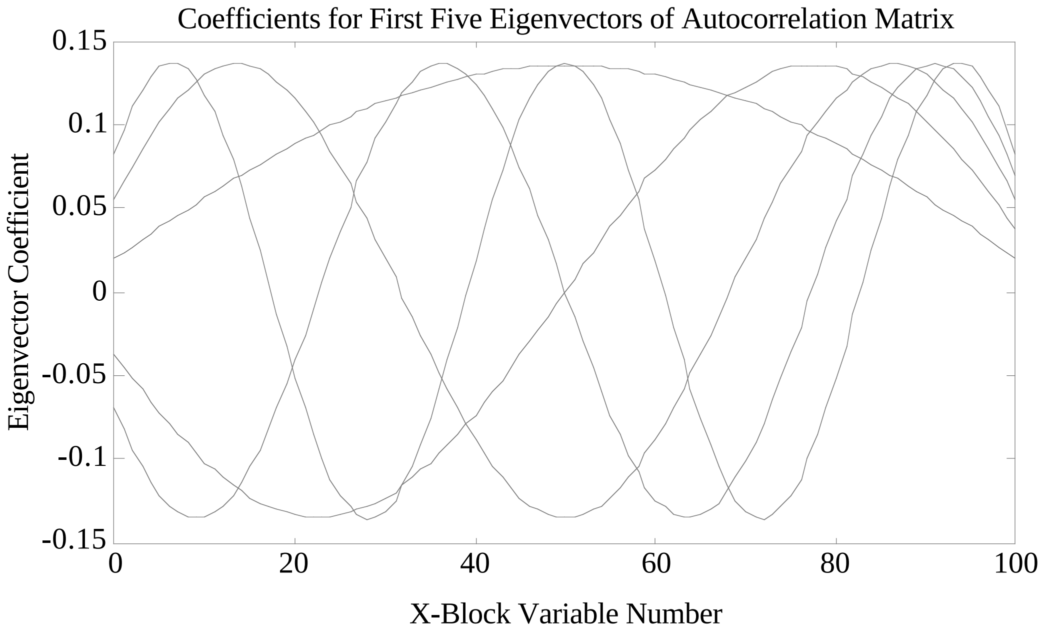

Figure 5.3. Coefficients in First Five Eigenvectors of Autocorrelation Matrix of White Noise Process Through First Order Filter.

It is known from numerical experience that the coefficients of the eigenvectors of ACMs look like sine and cosine functions. Therefore, it is proposed that a cosine function is an eigenfunction of the analogous continuous time problem. Thus, it is proposed that

$$\int_{x=-a}^{x=a} e^{-|x+y|} \cos\left(\frac{x}{n}\right) dx = \lambda \cos\left(\frac{y}{n}\right) \tag{5.5}$$

for some choices of n and λ. In order to see if any values of n and λ can be found to make (5.5) true, the integral on the left hand side must be evaluated. In order to remove the absolute value function from the exponential it is convenient to break the above expression apart as follows:

$$\int_{x=-a}^{x=a} e^{-|x+y|} \cos\left(\frac{x}{n}\right) dx = \left\{ \begin{array}{ll} \displaystyle\int_{-y}^{a} e^{-(x+y)} \cos\left(\frac{x}{n}\right) dx & \text{for } x+y > 0 \\ \displaystyle\int_{-a}^{-y} e^{(x+y)} \cos\left(\frac{x}{n}\right) dx & \text{for } x+y < 0 \end{array} \right\} \tag{5.6}$$

The solution to 5.6 is:

$$\int_{x=-a}^{x=a} e^{-|x+y|} \cos\left(\frac{x}{n}\right) dx = \left\{ \begin{array}{l} \left[\dfrac{-e^{-(x+y)} \cos\left(\frac{x}{n}\right)}{1+n^{-2}} + \dfrac{e^{-(x+y)} \sin\left(\frac{x}{n}\right)}{\left(1+n^{-2}\right) n} \right]_{x=-y}^{x=a} \\ \left[\dfrac{e^{(x+y)} \cos\left(\frac{x}{n}\right)}{1+n^{-2}} + \dfrac{e^{(x+y)} \sin\left(\frac{x}{n}\right)}{\left(1+n^{-2}\right) n} \right]_{x=-a}^{x=-y} \end{array} \right\} \tag{5.7}$$

which reduces further to

$$\int_{x=-a}^{x=a} e^{-|x+y|} \cos\left(\frac{x}{n}\right) dx = \left(e^{-(\pi+y)} + e^{-\pi+y}\right) \left(\frac{-\cos\left(\frac{a}{n}\right)}{1+n^{-2}} + \frac{\sin\left(\frac{a}{n}\right)}{\left(1+n^{-2}\right) n} \right) + \frac{2\cos\left(\frac{-y}{n}\right)}{1+n^{-2}} \tag{5.8}$$

For particular values of n the second term on the right hand side of equation (5.8) is identically zero. Some algebraic manipulations show that this occurs when

$$n = \frac{\sin\left(\frac{a}{n}\right)}{\cos\left(\frac{a}{n}\right)} = \tan\left(\frac{a}{n}\right) \tag{5.9}$$

This gives the allowed "periods" of the cosine eigenfunctions. (The true period would be equal to 2πn.) The eigenvalues associated with each period n, λ_n are given by

$$\lambda_n = \frac{2}{1 + n^{-2}} \qquad (5.10)$$

In a similar manner one can rewrite equation (5.5), with sine replacing cosine, and obtain a new solution. The result is that the allowed periods of the sine eigenfunction are given by

$$n = \frac{-1}{\tan\left(\frac{a}{n}\right)} \qquad (5.11)$$

The associated eigenvalues are as given in equation (5.10).

The results from the continuous case can now be compared to calculated results from the discrete case. As an example, let us define an ACM using equation 5.2 with $\alpha = 0.9$ and 101 coefficients. In order to set up the analogous continuous time problem the integration limits must be specified. For the problem to be completely analogous the parameter a must be set so that the values in the continuous function map onto the same values in the discrete ACM. The easiest way to assure this is to specify that

$$\mathrm{ACM}(1,m) = \alpha^{m-1} = e^{-2a} \qquad (5.12)$$

which when rearranged yields

$$a = \frac{1}{2} \log\left(\frac{1}{\alpha^{m-1}}\right) \qquad (5.13)$$

For our example this gives a = 5.268. Equations (5.9) and (5.11) can now be used to determine the periods of the cosine and sine eigenvectors, respectively. This is shown graphically in Figures 5.4 and 5.5. In Figure 5.4 the intersections of the n = n line with the tan(a/n) curves give the values of n that are solutions to equation (5.5). Figure 5.5 shows the intersections of the n = n line with the -1/tan(a/n) curves giving the values of n which solve the sine analog of equation (5.5). Notice that in both cases, while there are infinitely many intersections of the curves, there are no more intersections of the curves to the right of the last intersection shown. The rightmost intersections in Figures 5.4 and 5.5 correspond to the first (associated with the largest eigenvalue) cosine eigenfunction and sine eigenfunction, respectively.

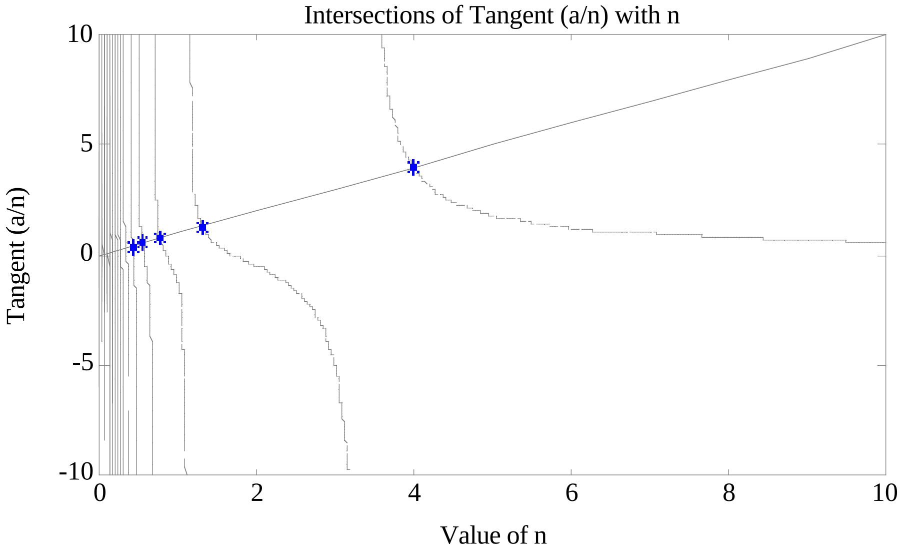

Figure 5.4. Points where n = Tangent (a/n) Showing the Frequencies of the Cosine Eigenvectors.

Figures 5.6 and 5.7 demonstrate the agreement between the continuous and discrete cases. In Figure 5.6 the coefficients of the first three cosine eigenvectors of the discrete problem (eigenvectors 1, 3 and 5) are plotted (solid lines) along with the predicted values based on the continuous problem (+'s). The coefficients are plotted against the position in the x-domain. The coefficients of the the continuous problem were scaled to yield unit vectors. In Figure 5.7 the first three sine eigenvectors (numbers 2, 4 and 6) are shown along with the continuous problem solution. Note how the agreement is nearly perfect in both figures. Small discrepancies can be seen in the higher frequency terms.

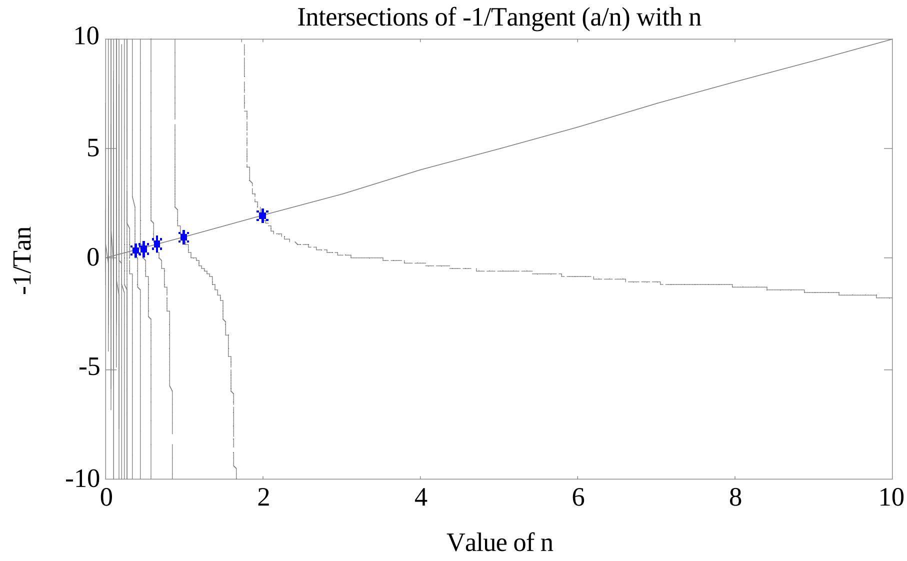

Figure 5.5. Points where n = -1/Tangent (a/n) Showing the Frequencies of the Sine Eigenvectors.

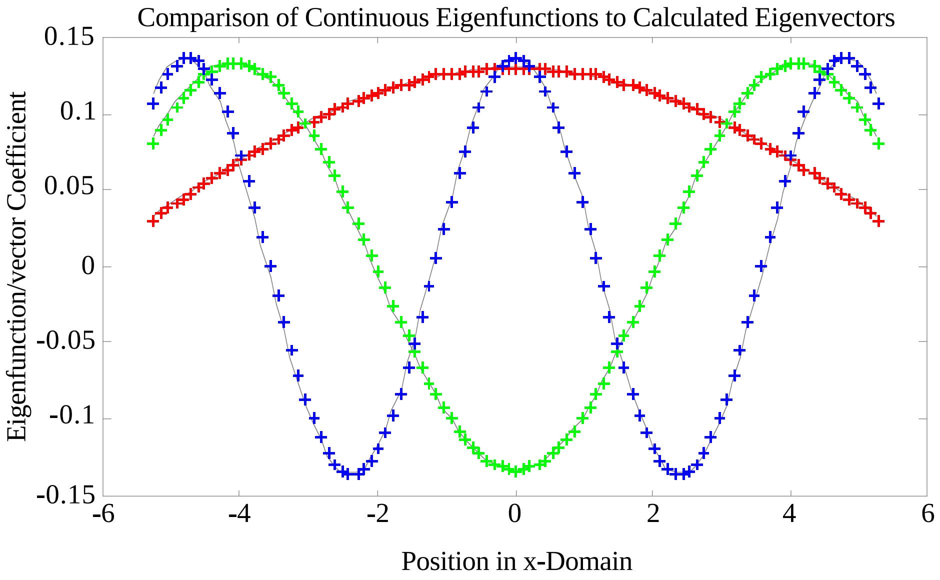

Figure 5.6. Coefficients of the Cosine Eigenvectors of the Discrete Matrix Shown with Eigenfunction Solutions to the Continuous Problem.

In addition to checking the frequencies of the eigenfunctions versus the eigenvectors, it is also possible to check the agreement with the corresponding eigenvalues. Because the scaling and size of the ACM affects the magnitude of the eigenvalues, it is not possible to compare the values directly. However, it is possible to show that the distribution of both sets of eigenvalues is the same. This is done in Figure 5.8 where the solid line corresponds to the calculated eigenvalues of the discrete problem. The stars correspond to the scaled eigenvalues of the continuous problem. The eigenvalues of the continuous problem were scaled so that the first eigenvalues in both distributions are equal. The agreement between the eigenvalues is quite good. There are slight discrepancies between the smaller eigenvalues.

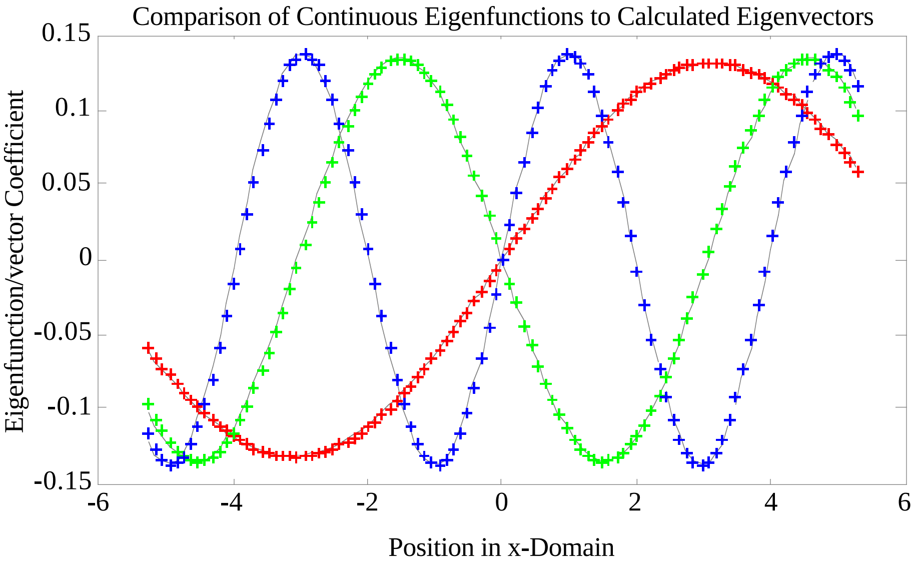

Figure 5.7. Coefficients of the Sine Eigenvectors of the Discrete Matrix Shown with Eigenfunction Solutions to the Continuous Problem.

Numerical simulations show that the agreement between eigenvector decompositions of ideal discrete ACMs and the continuous analog are very good, even as the matrices become quite small. The agreement between cases suffers somewhat more as the α parameter in the discrete case is decreased. This happens because for small α the ACM loses its "smoothness", *i.e.*, the differences between adjacent entries in the matrix is quite large. Decreasing the α parameter is equivalent to increasing the sample rate of the input signal.

In practice, when identifying FIR models the input covariance matrix will not be ideal, *i.e.*, the matrix will not be perfectly Toeplitz due to the finite data record. However, for typical cases where the number of coefficients is >10 and there are several hundred samples, the agreement between calculated principal components and pure sines and cosines is very good. Other ACM forms arrived at through higher order filters produce similar, though not identical, results. Certain cases, such as white noise through a second order under-damped filter, produce eigenvectors that still have periodic behavior, but are essentially combinations of frequencies and can be rather complex.

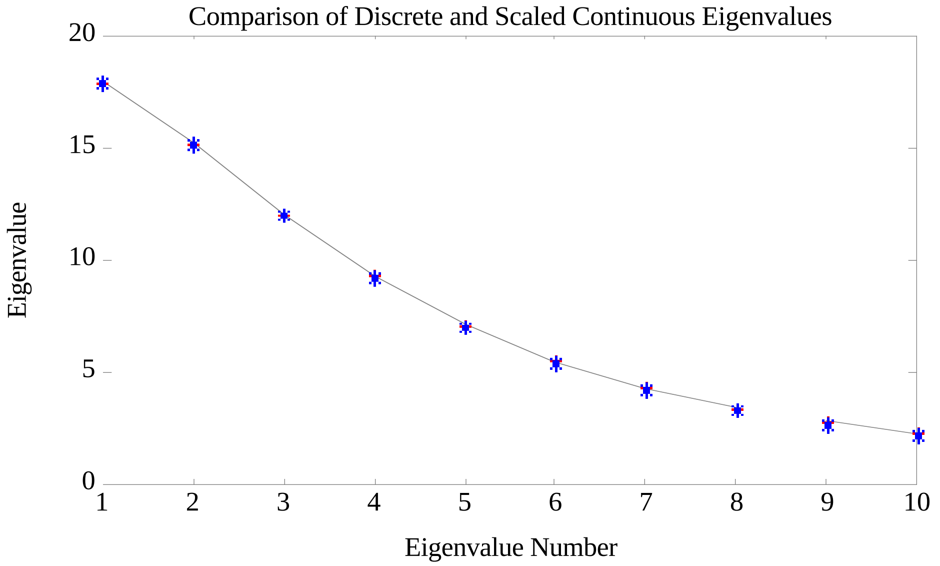

Figure 5.8. Eigenvalues of the Discrete Case Shown with Scaled Eigenvalues of the Continuous Case.

5.2.2 Frequency Domain Effects of PCR

Based upon the results of the previous section, it can be seen that, in some sense, PCR breaks the input signal up into components of differing frequencies. This has a direct effect on the models obtained from the technique. Two identification experiments are used here to illustrate this effect. In the first case the true system is first order, while in the second case the true system is second order under-damped. Both systems have unit gain at steady state. In both identification experiments a Pseudo Random Binary Sequence (PRBS) input signal was generated by filtering white noise through a second order Butterworth filter and taking the sign of the result. The input signal was considered to be 1 when this result was positive and -1 when it was negative. The process models and filter parameters are given in Table 5.1. The calculated PRBS was used to generate a calibration set of 500 samples from both processes. A segment of the PRBS input generated with these parameters is shown in Figure 5.9, along with the output from the Case 2 process. PCR was used to identify FIR models of the (noise free) processes. In each case 30 FIR coefficients were used. The frequency behavior of these models using different numbers of PCs was then tested.

Table 5.1. Numerator [$A(q^{-1})$] and Denominator [$B(q^{-1})$] Polynomial Coefficients.

	Numerator	Denominator
Case 1	.1426	1 -0.8574
Case 2	0.1129 0.1038	1 -1.5622 0.7788
Filter	0.0015 0.0029 0.0015	1 -1.8890 0.8949

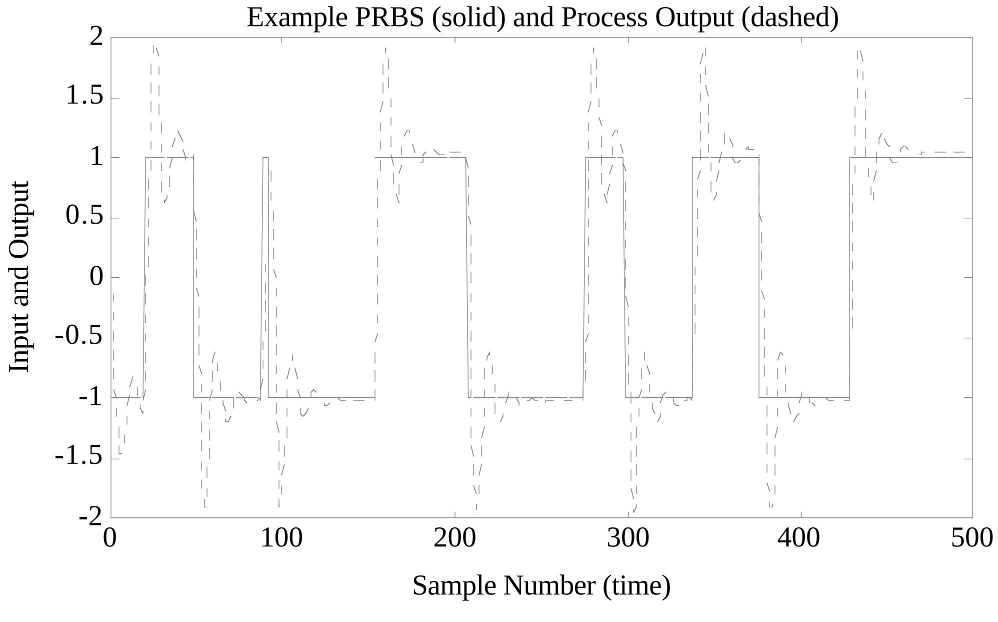

Figure 5.9. Example PRBS Input and Case 2 Process Output.

The results of the identification experiment are shown in Figures 5.10 and 5.11 which give the Bode gain magnitudes as a function of input frequency for each of the true systems and their corresponding FIR models. The true response of each system is shown as the hatched line. The frequency response of the PCR identified FIR models, using from 1 to 6 PCs, are also shown. Note how the 1 PC models accurately describe the process behavior at low frequency only. As PCs are added to the regression, the model matches the actual system response to progressively higher frequencies.

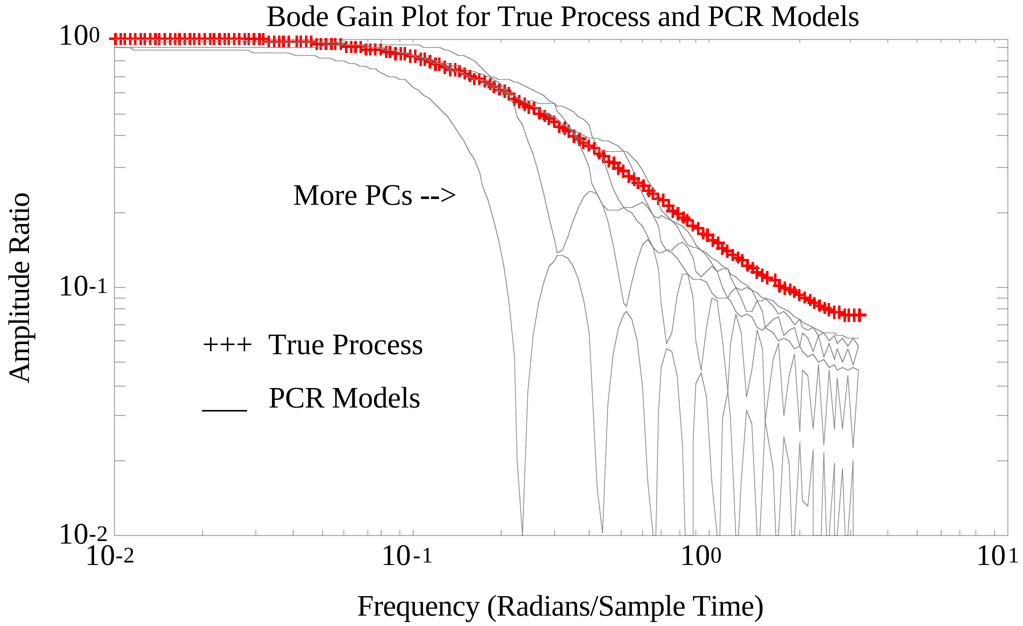

Figure 5.10. Bode Gain Magnitude Plot for First Order Process (Case 1) and PCR Models.

The "dips" in the gain for the PCR models are a consequence of the sinusoidal nature of the PCs used to construct the FIR models. For instance, it has been shown above that the FIR model identified using just 1 PC has coefficients that are a cosine function of a particular frequency. Certain input frequencies, therefore, can be orthogonal to this frequency and will not be passed by the model. The gain "dips" occur, in fact, at frequency intervals of 2π, which would be expected based on the behavior of orthogonal cosine functions.

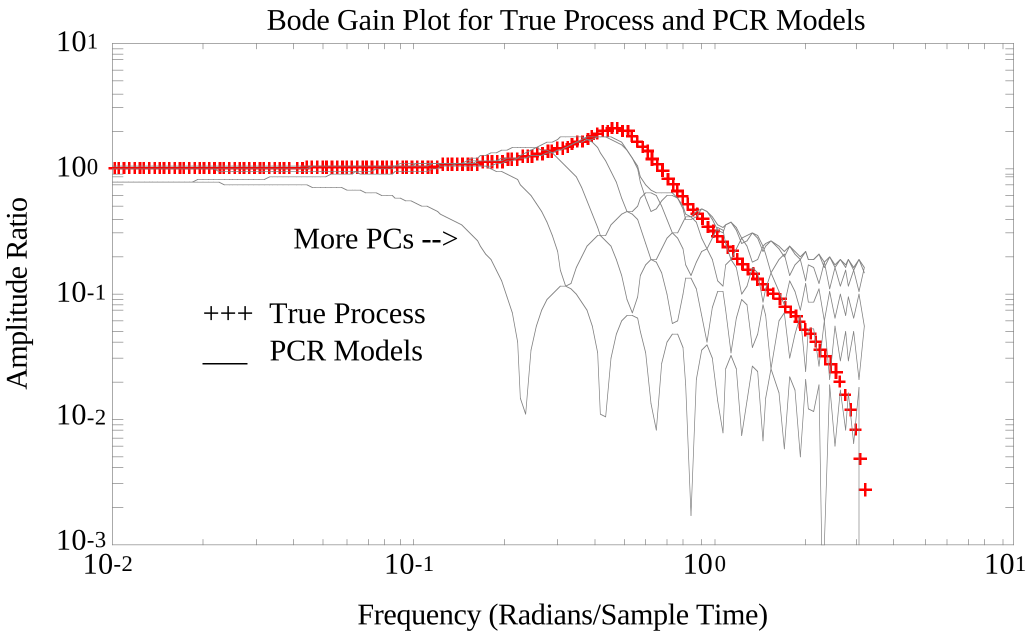

Figure 5.11. Bode Gain Magnitude for Second Order Process (Case 2) and PCR Models

The point concerning the consequences of the sinusoidal nature of the PCs is an important one and deserves some further investigation and clarification. Suppose that the first PC from the decomposition of the ACM can be represented as a continuous cosine function of period 4π (frequency = 0.5) over the interval from -π to π. Here the interval and period are chosen so that the function will go through one half cycle over the interval. This is similar to the first PC from a typical ACM decomposition, as demonstrated previously (compare to Figure 5.3). For a sinusoidal input the process output y(t) of the 1 PC (continuous) FIR model must then be

$$y_1(t) = c_1 \int_{-\pi}^{\pi} \cos(0.5\ x) \cos(m\ (x + t))\ dx \qquad (5.14)$$

where m is the frequency of the input signal and c_1 is the constant determined from regressing **y** onto the scores vector $\mathbf{t}_1$. The solution to this integral is

$$y_1(t) = \left[\frac{\sin\left[-mt + (0.5 - m)\,x\right]}{2\,(0.5 - m)} + \frac{\sin\left[mt + (0.5 + m)\,x\right]}{2\,(0.5 + m)} \right]_{x = -\pi}^{x = \pi} \qquad (5.15)$$

which when evaluated yields

$$y_1(t) = \frac{\sin\left[-mt + (0.5 - m)\,\pi\right] + \sin\left[mt + (0.5 - m)\,\pi\right]}{2\,(0.5 - m)} + \frac{\sin\left[mt + (0.5 + m)\,\pi\right] + \sin\left[-mt + (0.5 + m)\,\pi\right]}{2\,(0.5 + m)} \qquad (5.16)$$

It is easily seen that for values of m that are equal to an integer + 0.5 the value of the numerator in both terms is identically zero for all t. Thus it is apparent that frequencies of 1.5, 2.5, 3.5 etc. will not pass through the 1 PC model.

Let us further assume that the second PC can be represented as a sine function of period 2π (frequency 1) over the same interval (again, compare with Figure 5.3). The contribution of the second PC to the model will then be

$$y_2(t) = c_2 \int_{-\pi}^{\pi} \sin(x) \cos(m\,(x + t))\,dx \qquad (5.17)$$

Using a procedure similar to that shown in equations (5.13) to (5.16) it can be shown that when m is an integer greater than 1 the value of the integral in (5.17) is zero for all t. Thus we expect that the second PC will make no contribution to the 2 PC model at frequencies of 2, 4, 6 etc. and the response at these points will be equal to the 1 PC model response. It is also interesting to note that there are no values of m less than 1 for which the integral vanishes.

Considering once again Figures 5.10 and 5.11 it can be seen that the behavior shown is expected based on the mathematical argument given above. The 1 PC model does not pass certain frequencies that occur at even intervals. The 2 PC model passes these frequencies but adds nothing to the centers of the intervals of the 1 PC model. Similar behavior continues as more PCs are added to the model. Because the "period" of the first and subsequent PCs can change, given changes in the input signal, the exact location of the "dips" will also change. The behavior of the models will be similar in any case, however.

5.2.3 Effect of Filtering on ACM Decomposition

It was mentioned above that ACMs that are generated by processes other than white noise through first order filters have more complex decompositions. However, as Figures 5.10 and 5.11 demonstrate, even in the case where the data record is finite and the input signal was generated though a higher order filter (or modified to be a PRBS) the results are quite similar to the ideal ACM case.

The following example illustrates how different filtering options affect the ACM decomposition. An input signal was generated by filtering a 1000 sample white noise sequence through a first order process as in equation (5.2) with $\alpha = 0.8$. The 30 point autocorrelation matrix (the matrix which considered time shifts from -30 to +30 units) of this signal was then formed and an eigenvector decomposition of it computed. The original signal was then filtered through a first order process where the value of α was varied from 0.1 to 0.7. The eigenvalues of the ACM from the filtered signal are compared with those of the original signal in Figure 5.12. Note how the filtering has a larger affect on the small eigenvalues (associated with higher frequencies) than on the large eigenvalues. This is expected because filter is a low-pass type. Increasing the value of α lowers the cutoff frequency and begins to reduce the larger eigenvalues (associated with lower frequencies) to a greater degree.

Even though the eigenvalues are greatly affected by filtering, the eigenvectors of the ACM do not change a great deal as shown in Figure 5.13. The figure shows the coefficients of the first, sixth and tenth eigenvectors for all of the values of a tested. Note how they remain largely identical throughout the series.

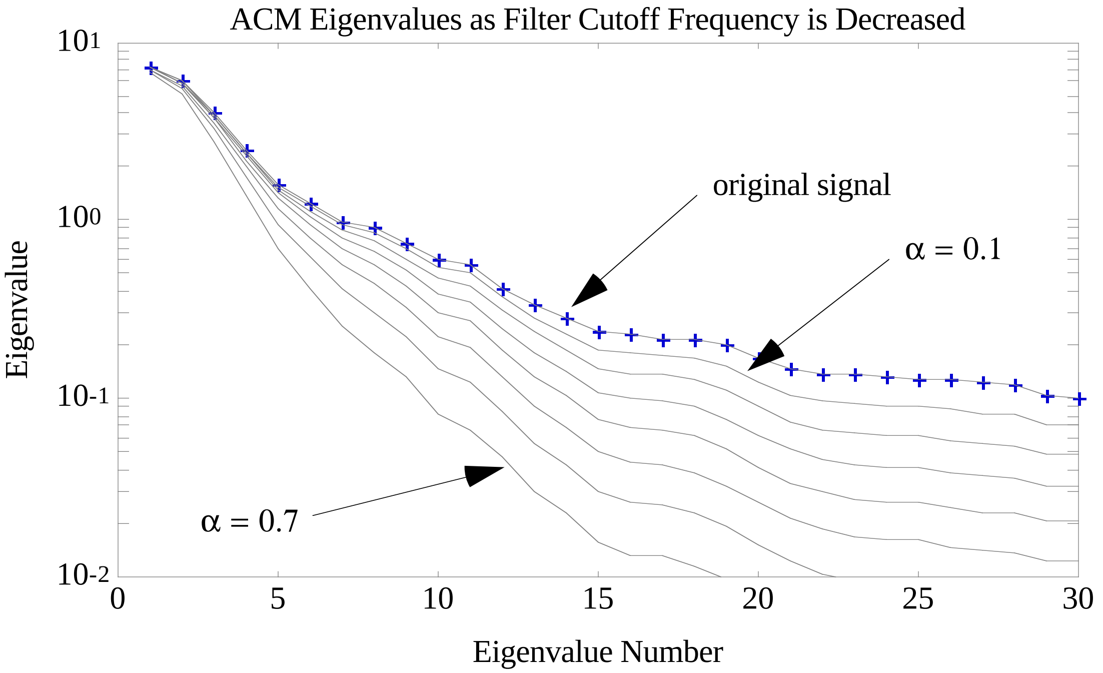

Figure 5.12. Eigenvalues of Autocovariance Matrix for Original Signal and Filtered Signals as Cutoff Frequency is Lowered.

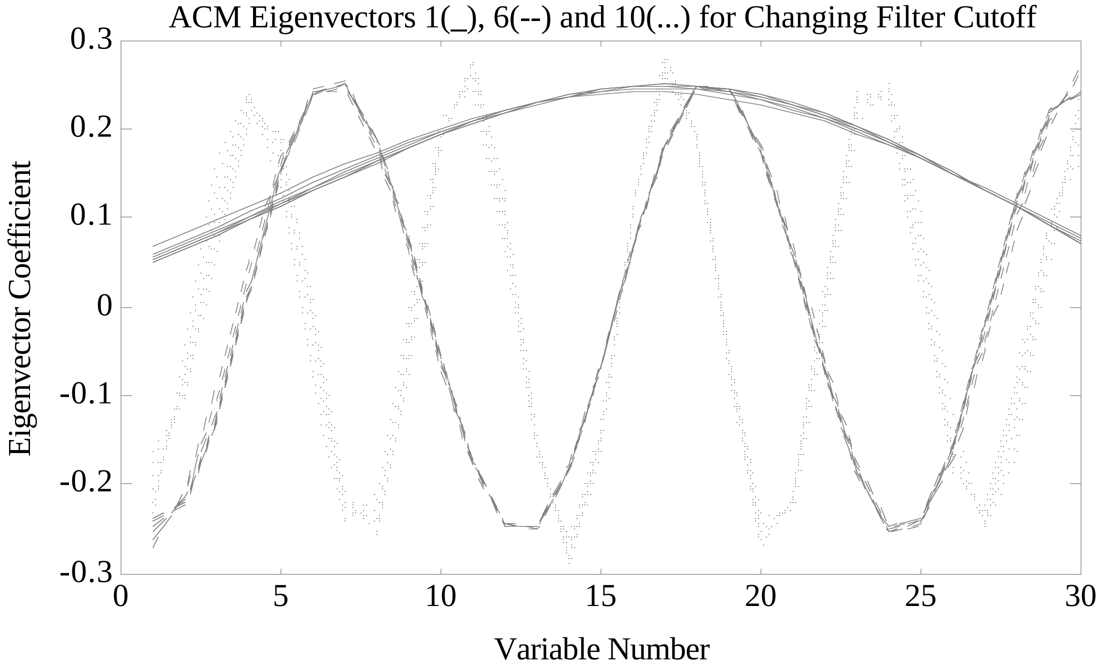

Figure 5.13. Coefficients of the First, Sixth and Tenth Eigenvectors of the ACM for

Varying Filter Parameters.

As an aside here, it is interesting to note that low pass filtering of the input and output data prior to identifying an FIR model is probably not a good idea, particularly if MLR is the method to be used. Low pass filtering tends to make the autocovariance matrix extremely ill-conditioned, and it is this matrix which must be inverted in MLR. On the other hand, high pass filtering, to de-trend the data, may still be advantageous in certain situations.

Numerical results indicate that FIR models identified from PCR will tend to fit first in the frequency ranges where there was the most power in the input signal. For instance, if a band-pass filter is used to generate the input signal, the models identified will be fit first in the frequency range where the input signal had the most power. As PCs are added to the regression, the models will fit in regions further away from the "band". This is consistent with the theory of eigenvector decomposition of the ACM developed here. Frequencies with the most power are associated with the largest eigenvalues and will be extracted from the input sequence first, and the resulting models can only be expected to fit the true process behavior in the range of the frequencies considered in the regression. Filtering to isolate frequency ranges of interest before modelling is a common approach, as indicated by Ljung (1987) and Rivera (1990a and 1990b).

5.2.4 Frequency Domain Effects of CR

If the frequency domain interpretation of PCR developed here is correct, then it would be expected that the CR techniques would show similar behavior in the frequency domain, especially for high powers of the singular values (CR techniques on the PCR side of the continuum). However, some differences should result because CR produces latent variables that are rotations of the original eigenvectors. Therefore, the resulting vectors would be less orthogonal to particular frequency inputs and would not produce gain "dips" as severe as those seen in PCR models. This result is shown graphically in Figure 5.14.

Here a collection of Bode gain plots for 1 latent variable models of the Case 1 (Table 5.1) process are shown as a 3-d surface. The continuum parameter of the models tested all lie in the range of PCR (infinity) to conventional PLS (1). Specifically, thus the powers of the singular values used in the continuum regression were 1 (PLS), 1.41, 2, 2.83, 4, 8 and infinity (PCR).

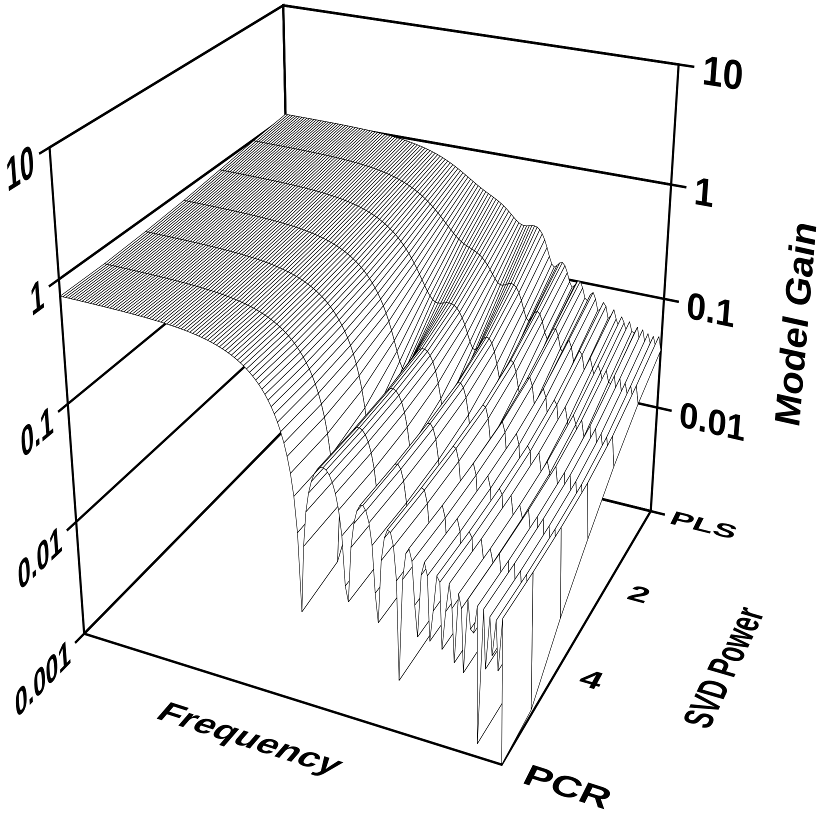

Figure 5.14. Bode Gain Magnitude for 1 Latent Variable Models of Second Order Under-damped Process--No Noise.

In Figure 5.14 it can be seen that the 1 latent variable PCR model has strong "dips" in the gain (as in Figure 5.10), however, these dips tend to be reduced in the models that allow for more rotation from the original eigenvectors (the models closer to PLS). As might be expected, the rotation results in better cancelation of gain "dips" in the lower frequencies than in the higher frequencies. Note that the lowest frequency gain dip in the PCR model is essentially non-existent in the PLS model while the gain dips at higher frequency change very little. This type of behavior is expected because the latent variables in PLS are biased towards rotation in the direction of other eigenvectors associated with relatively large eigenvalues and that have a high degree of correlation with the process output, *i.e.*, directions with a large amount of covariance. High frequency gain dips in the PCR model are not cancelled as well in the PLS model because cancelation requires a vector with a component in the direction of the eigenvector associated with those high frequencies. Because the eigenvalues of these high frequency eigenvectors are quite small, however, the latent vectors do not tend to get rotated towards them, *i.e.*, there is very little covariance in this direction so the PLS vectors do not tend to rotate this way.

5.2.5 Effect of Identification Conditions on FIR Modelling

Now that it is clear how the PCR method works for FIR model identification, the investigation into the effect of process parameters on models identified by the CR method may proceed. The goal of this section is to develop an understanding of the effects that process measurement noise level, process dynamics, amount of input excitation and mis-estimated time delays have on the location of the "best models" in the CR parameter space. The effects that these factors have on the relative advantage of the CR method over MLR will also be considered.

A collection of 7 representative process models is used to test the CR procedure. The numerator and denominator polynomials of the models are given in Table 5.2. These models are intended to span a variety of dynamic behaviors typical of chemical processes. The processes are as follows: first order, second order over-damped, first order over second order over-damped, second order over-damped with right half plane zero, fifth order over-damped and fifth order under-damped. The Bode Gain Magnitude plots of the test processes are shown in Figure 5.15. The frequency axis on the Bode plots is scaled so that the Nyquist frequency (twice the sampling frequency) corresponds to 2π on the plot.

Table 5.2. Numerator and Denominator Polynomial Coefficients for Test Models.

	Numerator	Denominator
Model 1	.1426	1 -0.8574
Model 2	0.0256 0.0215	1 -1.5503 0.5974
Model 3	-0.0742 0.1256	1 -1.5353 0.5866
Model 4	0.1707 -0.1330	1 -1.5834 0.6211
Model 5	0.1129 0.1038	1 -1.5622 0.7788
Model 6 numerator	0.0354 -0.0475 0.0222 -0.0041 0.0002	
Model 6 denominator	1 -2.9004 3.2427 -1.7335 0.4395 -0.0421	
Model 7 numerator	0.1976 -0.2723 0.1345 -0.0285 0.0023	
Model 7 denominator	1 -2.7819 3.0946 -1.6267 0.3753 -0.0278	

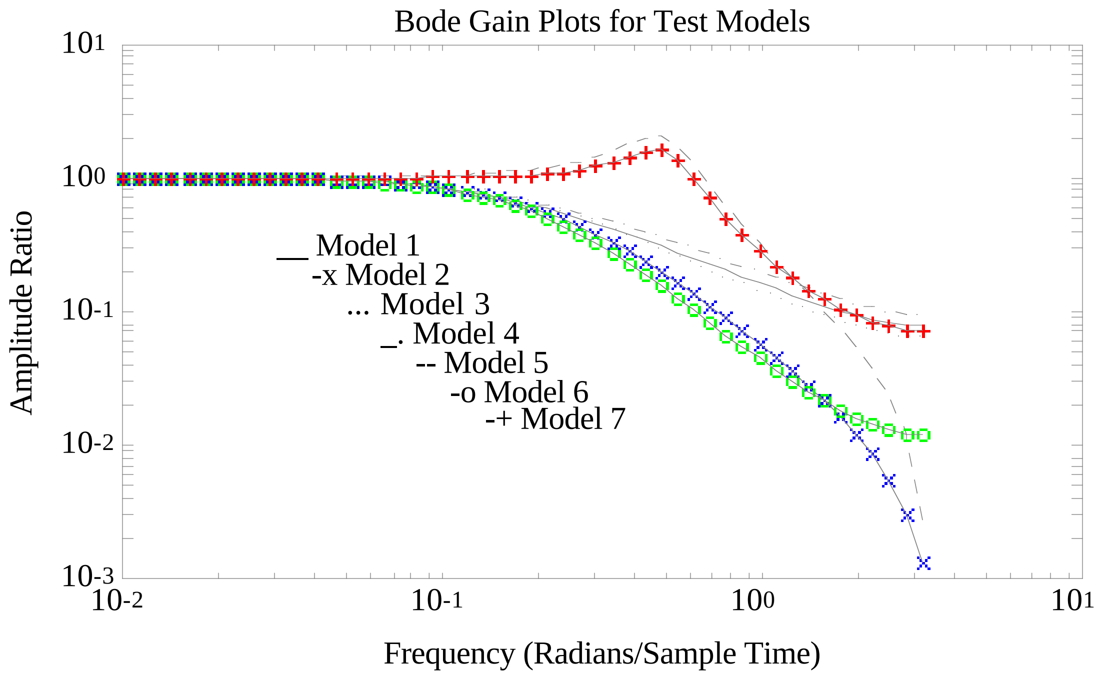

Figure 5.15. Bode Gain Magnitude Plots for the Seven Test Processes.

5.2.5.1 Effect of Process Measurement Noise

The first factor investigated was the effect of process measurement noise. In this series of tests the measurement noise level was varied so that the noise standard deviation was 5% to 100% of the process gain, which was unity. The FIR window width used in all of these experiments was 30 sampling periods. All the processes tested reach 99% of steady state gain after 30 sample periods. A PRBS input, developed using the filter in

Table 6.1, was used to generate 500 input/output samples from all seven processes. The process measurement noise levels were adjusted to have a standard deviation of 5, 10, 20, 40 and 100% of the process gain. Models were tested with a separate PRBS that was generated using the same filter as the PRBS used for calibration.

Some typical model error (PRESS) surfaces for varying noise levels are shown in Figures 5.16 and 5.17. Figure 5.16 shows the PRESS surface for identification of a first order process (Model 1 from Table 6.2) with 5% noise, while Figure 5.17 shows the PRESS surface when 40% noise is added. Note how, at the 5% noise level, the optimum models are not much better than the MLR models. At higher noise levels the "valley" is much deeper relative to the MLR "plain", which has risen. Note also that the location of the "valley" shifts to fewer latent variables at the higher noise levels. This is consistent with PCR and PLS experience; when more noise is added to the data the best models generally are those with fewer latent variables.

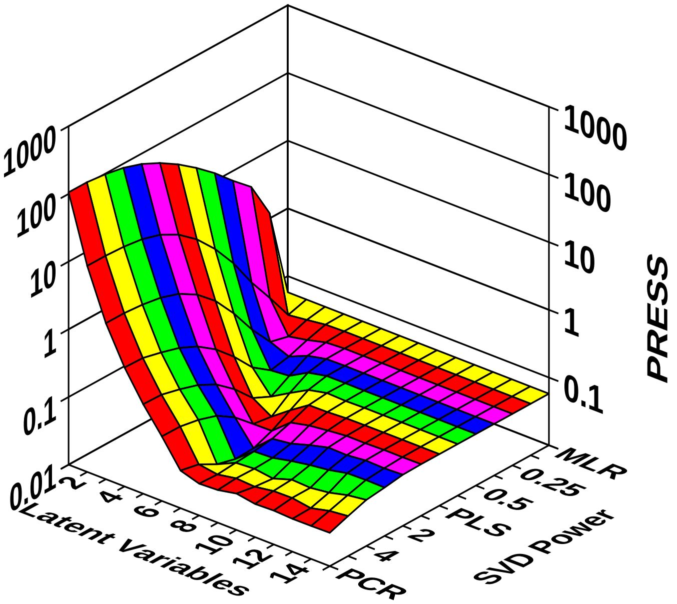

Figure 5.16. Continuum Regression PRESS Surface From Models of First Order Process (Model 1) with 5% Process Noise.

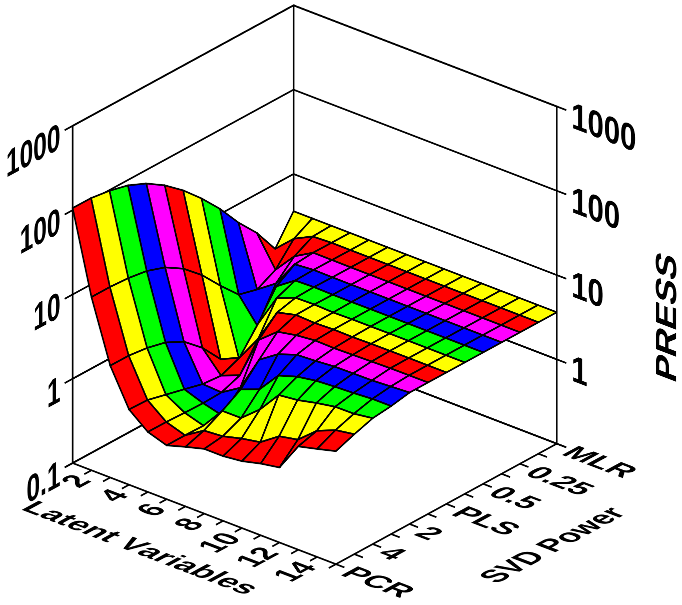

Figure 5.17. Continuum Regression PRESS Surface From Models of First Order Process (Model 1) with 40% Process Noise.

The same information in Figures 5.16 and 5.17 is presented in more compact form in Figure 5.18, which shows the location of the valley bottom for all the noise levels tested. In Figure 5.18 each line gives the number of latent variables in the optimum model for each continuum parameter. For example, the figure shows that at a noise levels of 5, 10 and 20% of the process gain there are 8 latent variables in the optimum PCR model. When the noise level is changed to 40% there are 6 LVs in the optimum model and only 4 LVs in the optimum PCR model for a noise level equal to 100% of the process gain. The "b's" in the figure indicate the location of the best overall model for each noise level tested.

With the frequency domain interpretation of the CR techniques in mind, the shift to fewer latent variables at higher noise levels can be seen as being similar to filtering of the process data. The filtering is done so that a range of frequencies is selected that will optimize the performance of the model resulting from the regression. In the CR techniques, as the regression models are built up, progressively higher frequencies are considered. While the power in the process output drops off at higher frequencies, the noise (which is white) does not. Eventually, a frequency is reached where the ratio of the power in the output to the power in the noise is not large enough to obtain an accurate process model at the frequency, *i.e.*, the model is as likely to describe the particular noise sequence as it is to describe the process. When this happens, adding this latent variable would not improve the predictive ability of the model. At higher overall noise levels, the point at which the noise variance overpowers the process variance occurs at lower frequencies. This results in fewer latent variables in the optimum models for the higher noise cases.

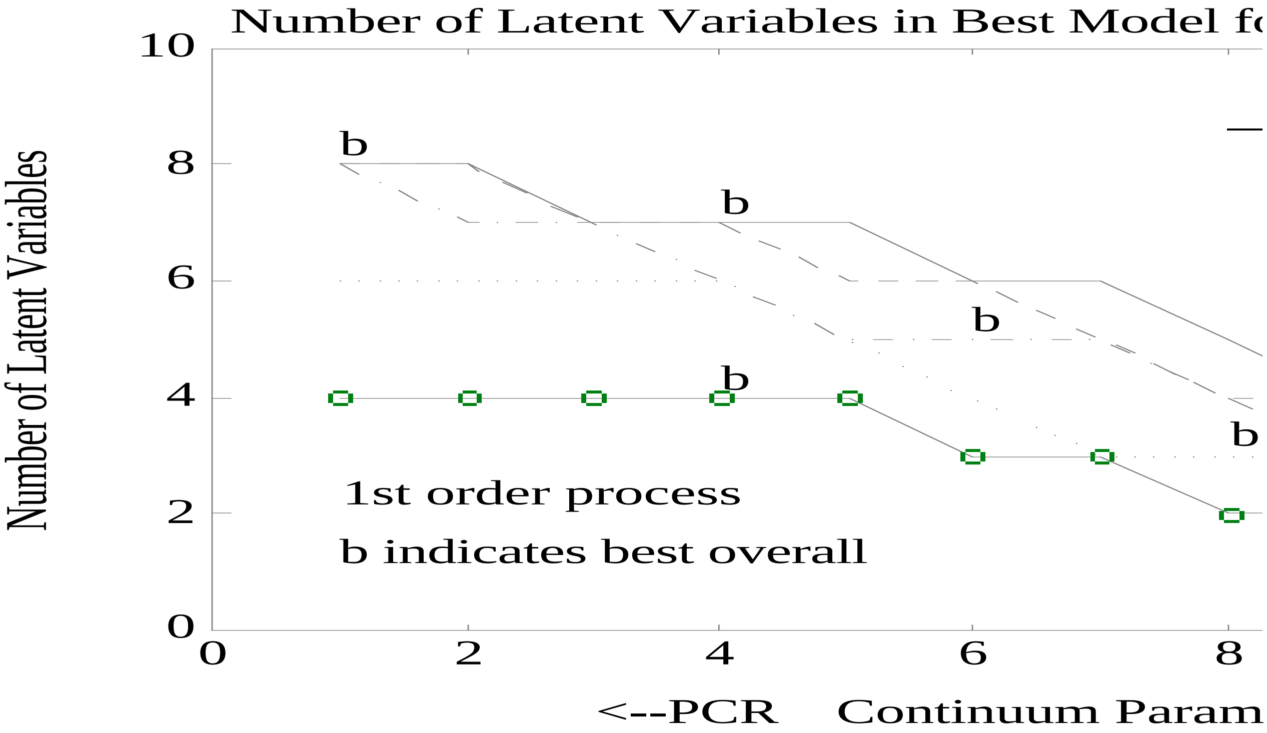

Figure 5.18. Effect of Process Noise Level on Number of Latent Variables in Best Model for Each Continuum Parameter.

5.2.5.2 Effect of Process Dynamics

The effect of process dynamics on the location of the best models in the CR parameter space was investigated by identifying models using an identical input sequence, test sequence and measurement noise. Once again, 500 samples were generated for each process and the same noise sequence, equal to 20% of the process gain, was added to each output.

Figure 5.19 shows the location of the PRESS valley bottom for each process. The "b's" indicate the location of the best model of each process. Note how some processes have more latent variables in the optimum models than others. The reason for the differences becomes more apparent after consideration of the Bode Gain plots of the processes shown in Figure 5.15 and the Bode Gain plots of typical PCR models shown in Figures 5.10 and 5.11. The PCR models tend to fit the gain behavior up to a certain frequency (determined by the frequency of the last PC used) then drop off at the rate of about a decade per decade. Processes like the second order under-damped system which go through a maximum gain then drop off rapidly require a large number of PCs to fit this behavior accurately. On the other hand, processes such as the 4th over 5th order system with gain that starts to drop off at relatively low frequency and decline with slope ~= 1 are fit very well using only a few PCs.

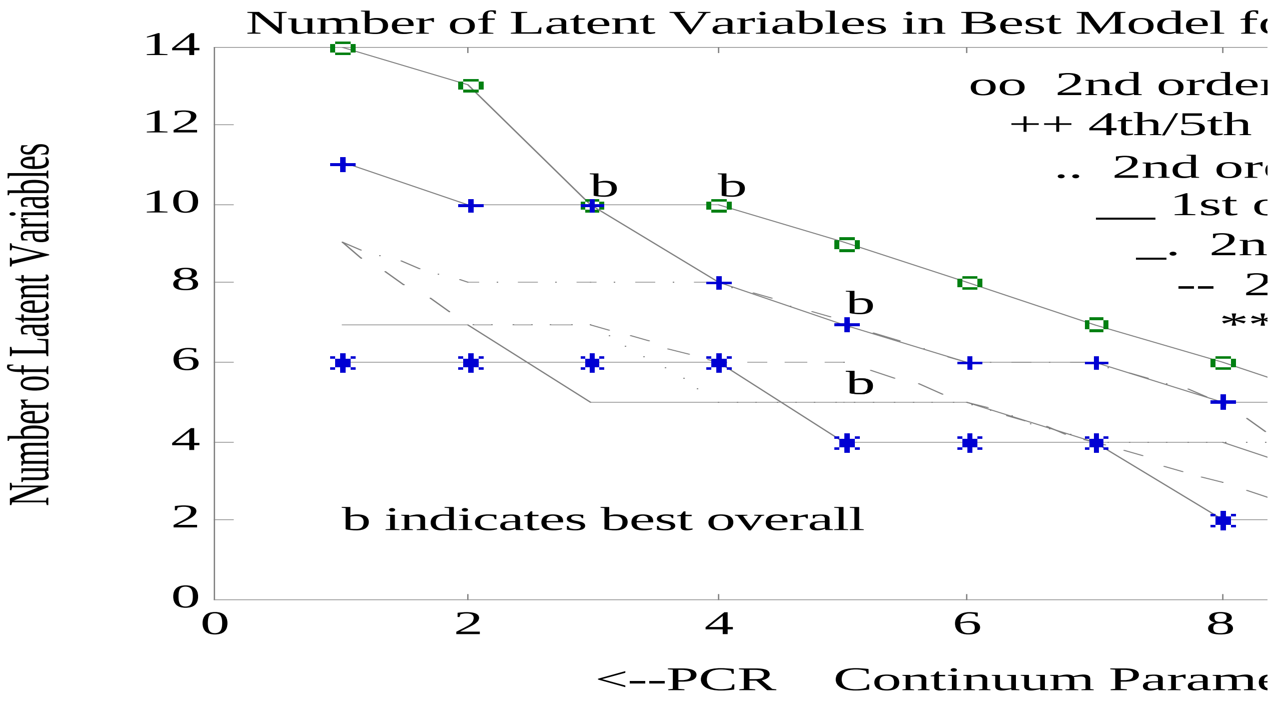

Figure 5.19. Effect of Process Dynamics on Number of Latent Variables in Best Model for Each Continuum Parameter.

The effect of dynamics on the location of the optimum models could also be seen from the filtering point of view considered in the previous section. Processes that have more output power at higher frequencies can be fit at these frequencies more accurately, because the ratio of deterministic to noise variance is greater. Therefore, more latent variables, descriptive of higher frequencies, can be used before the regression results degrade the model performance.

5.2.5.3 Effect of Input Excitation Level

The effect of differing levels of input excitation on the location of the best CR models is shown in Figure 5.20. The lines plotted correspond to the cutoff frequencies of the different filters used on the original white noise input signal. In each case a second order low-pass Butterworth filter was used. The cutoff given is in units such that 1 corresponds to the Nyquist frequency (twice the sample rate). Thus, the filter cutoff of 0.025 corresponds to an input signal with very little excitation in the frequency range covered by the model, whereas a cutoff of 0.4 constitutes a large amount of excitation.

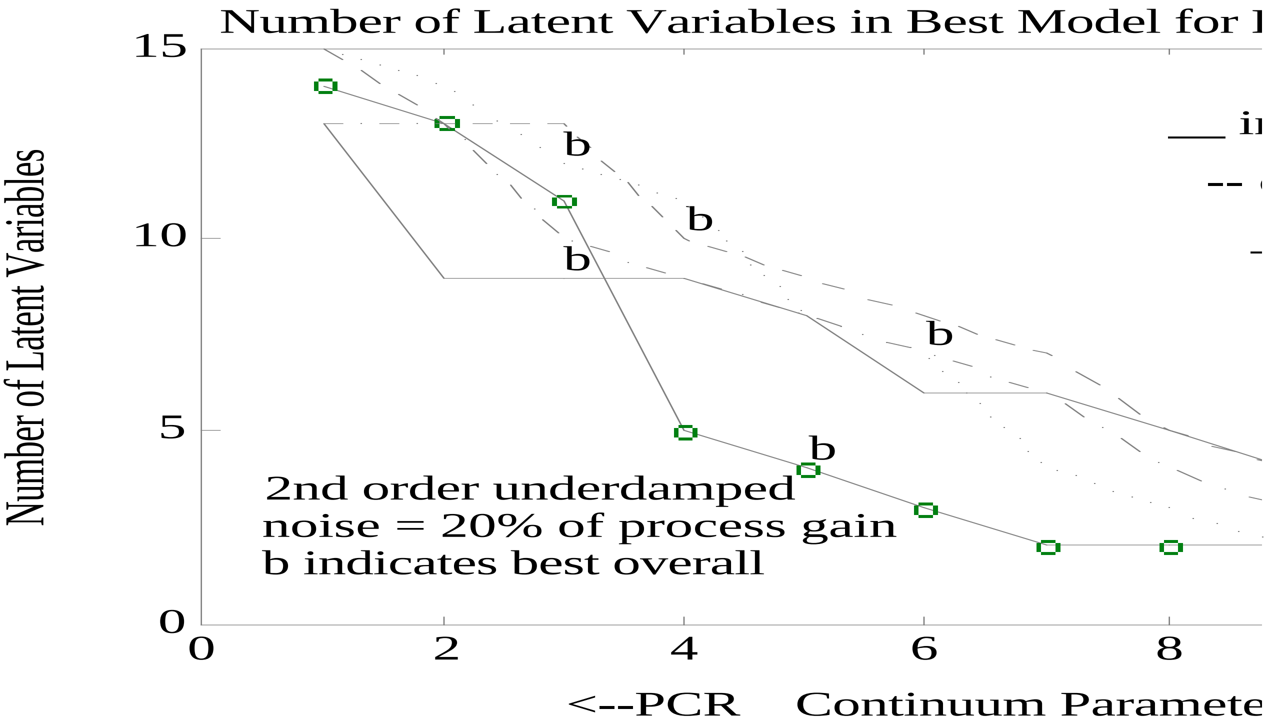

Figure 5.20. Effect of Input Excitation on Number of Latent Variables in Best Model for Each Continuum Parameter.

Note how the lines in Figure 5.20 cross. It appears that as input excitation is increased, more latent variables are retained in the models identified with large continuum parameters (techniques close to PCR) while fewer latent variables are used in the models identified with small continuum parameters (techniques close to MLR). This trend is understandable when one considers the frequency domain interpretation of how the techniques work. For MLR models, a greater amount of input excitation keeps the autocorrelation matrix better conditioned and results in a good solution. For PCR models, when there is a nearly equal amount of power at all frequencies (large amounts of input excitation), the PCs for a finite data record may be descriptive of any frequency or combination of frequencies, *i.e.*, the systematic extraction of eigenvectors of progressively higher frequency does not occur. To some extent these PCs may be quite arbitrary, and it may take a large number of them to adequately model the system response.

5.2.5.4 Effect of Time Delays

Processes often have pure time delays between input changes and output responses. If the length of these delays is known this is taken into account during the model identification. Only inputs that can effect the output are included in the regression matrix, which may mean that one or several of the left hand columns of **X** in section 4.1 are eliminated The objective is to avoid estimating any coefficients that are known to be zero since any estimate is necessarily less accurate than the true value of zero.

The effect of incorrect estimation of time delays is shown in Figure 5.21, which shows the location of the best models for time delay errors of 0 (time delay equal to time delay assumed) to 5 units (time delay 5 units greater than that assumed). The true process is Model 2 (second order over-damped), a PRBS input sequences was used to generate the data on the noise level is 20% of the process gain. As can be seen from the figure, the best models for the mis-estimated cases typically have more latent variables than for the case where the time delay estimate was correct.

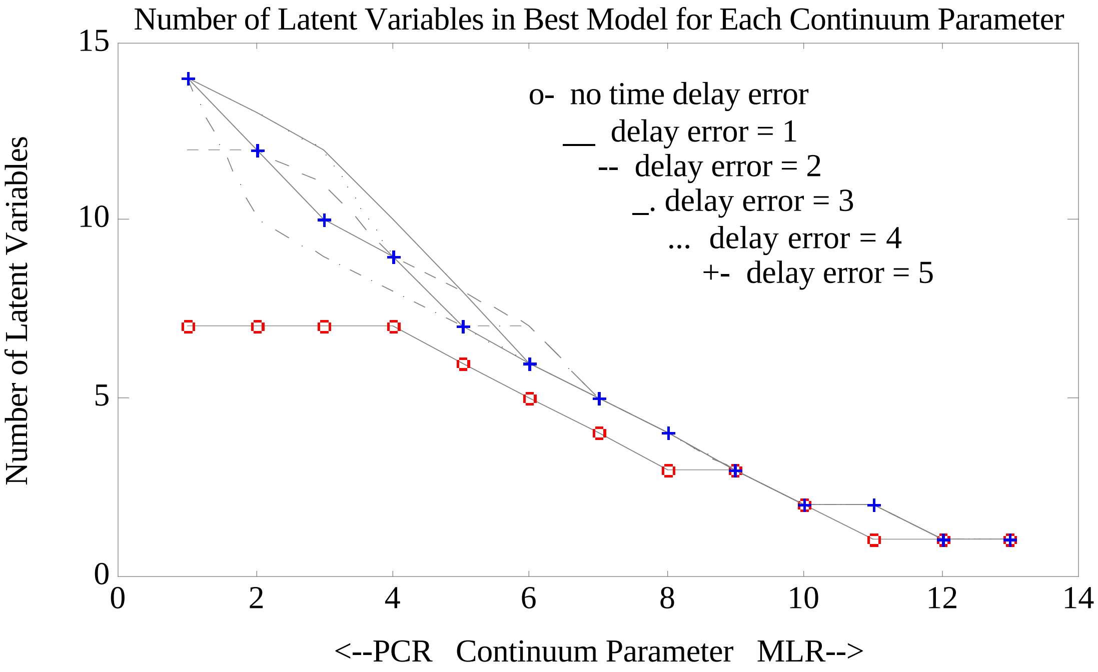

Figure 5.21. Effect of Mis-Estimated Time Delay on Number of Latent Variables in Best Model for Each Continuum Parameter.

The result of more latent variables in systems with incorrect time delay would be expected based on the way that the CR method imposes correlation on the FIR coefficients. The method can be thought of as approximating the true response with smooth functions that look like sines and cosines. The jump from a coefficient of 0 to some larger value is not "smooth" and requires PCs that look like higher frequency terms in order to fit it. Incorrect estimation of the time delay by progressively larger amounts does not further increase the number of latent variables in the optimum models, in fact, there is some evidence that the number of latent variables actually decreases.

The actual model errors for the best models for each time delay error are shown in Figure 5.22, *i.e.*, the figure gives the error of the models in the "bottom of the valley". It is apparent that the models identified by CR suffered a relatively larger adverse effect than the MLR models, but they are still better in an absolute sense. Note how the model error does not increase as the error in the estimate of the time delay is increased.

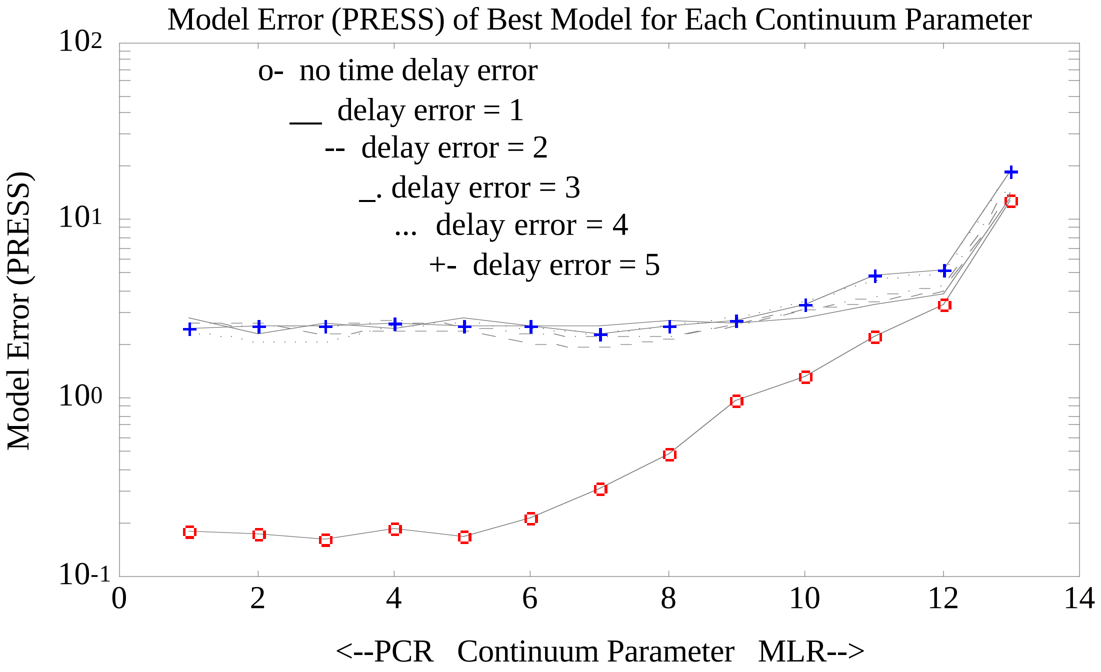

Figure 5.22. Model Errors for Correct and Incorrectly Estimated Time Delays.

It is important to point out that in the event of an incorrectly assumed time delay the leading FIR coefficients may not appear to be very near zero, and will also tend to look smooth as if they were modeling a true response. Thus, a cursory look at the leading coefficients may not lead one to check for a time delay. This is an artifact of the method, and the user should be aware of this potential problem. Furthermore, as Figure 5.22 shows, the model error improves dramatically when the system time delay is correctly estimated.

5.2.5.5 Effect of CR on FIR Coefficients

Before leaving the subject of FIR model identification, it is interesting to look at the effect of the CR identification method on the FIR coefficients for a typical case. A simulation was performed where a first order process (Model 1) was used to generate an input/output data set consisting of 500 samples. The input was a PRBS (generated using the filter in Table 5.1) and the process noise standard deviation was equal to 20% of the process gain. CR was used to identify FIR models of the process with 30 coefficients. The performance of the identified models was then assessed against the true model performance using three different inputs: white noise, a PRBS similar to the input from the identification data and a step. Figure 5.23 shows the true FIR coefficients of the first order process along with some of the models identified from input/output data.

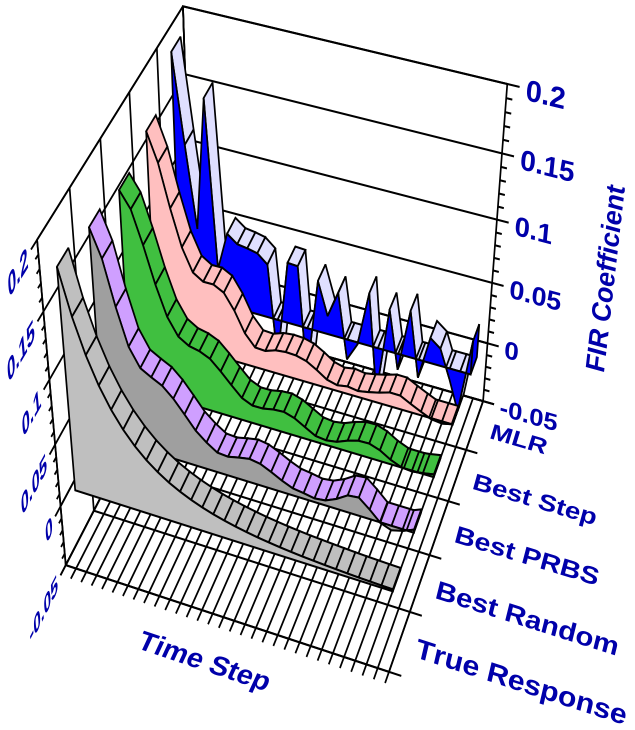

Figure 5.23. FIR Coefficients Identified from First Order System with Noise = 20% of Process Gain.

The models shown in Figure 5.23 are each "best" in some way, illustrating the point that the "best" model depends upon the criterion one chooses. The model labeled "Best Random" in the figure (second from the front) is most accurate relative to the true model when tested with a white noise input. The model labeled "Best PRBS" (third from the front) is best when tested against a PRBS similar to the one used for its identification. The model labeled "Best Step" (fourth from the front) has minimum error when a step test is used as the criterion. Finally, the MLR model (rearmost in the figure) is the one that "fits" the calibration data best in a least-squares sense. Note the jaggedness of the MLR model relative to the others. This is a result of the "ill conditioning" of the problem due to the near rank deficiency of the input autocorrelation matrix and the correlation in the FIR parameters themselves.

5.3 Continuum Regression for ARX Model Identification

In this section the effect of using Continuum Regression for the identification of ARX models is considered. It might be expected that there is less to be gained from using CR for ARX identification. There are typically fewer parameters to estimate and the correlation between them is less significant. Furthermore, from a theoretical standpoint, a properly posed ARX regression problem is not ill conditioned. If the proposed orders are either correct or less than the true orders of the process, the **X** matrix should be of full rank (provided that adequate input excitation has been used to generate the data). In fact, a rank deficient **X** matrix is an indicator that the problem has been over-parameterized, as pointed out in Ljung (1987).

Unlike the FIR case, it is difficult to characterize the result of a PCA decomposition of the input correlation matrix for ARX regression, *i.e.*, there does not appear to be a simple frequency domain interpretation of this decomposition. This is due to the more complex nature of the correlation matrix for ARX models. The **X** block includes both input and output values; this leads to a correlation matrix that includes correlations between lagged inputs and lagged outputs.

Numerical results indicate that ARX models identified with PCR show behavior that is similar to that seen in FIR identification. An example of this is Figure 5.24, which shows the Bode Gain plots of ARX models identified with PCR. The true system is model 7 from Table 5.2, a 4th over 5th order under-damped system. The process orders were correctly specified in the model identification procedure, and the data were noise free. A PRBS input was generated using the filter in Table 5.1, as described in section 5.2.2. The ARX models identified using 1-5 PCs are shown. As in the FIR case (compare with

Figures 5.10 and 5.11), the 1 PC model accurately describes only the low frequency behavior of the system. Additional PCs improve the high frequency accuracy. There is also some evidence of the pattern of gain "dips" as seen in the FIR case, but this artifact is not nearly as pronounced here.

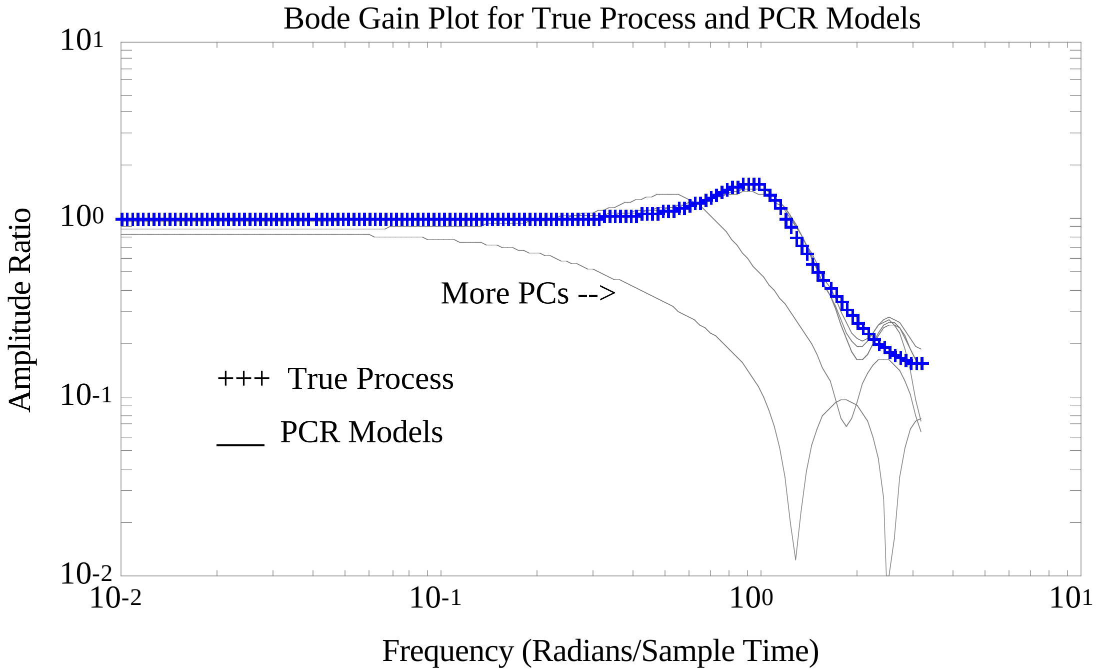

Figure 5.24. Bode Gain Plots of Process Model 7 and ARX Models Identified with PCR.

Tests were performed to investigate the effect of different factors on the quality of ARX models identified by CR. The same factors used in the FIR model identification experiments were considered, along with the additional possibility of an incorrect choice of model order. This was not a factor for FIR model identification because FIR models make no assumption about process order.

In almost all the cases considered, there was little difference between the best CR models and those obtained with MLR. In particular, when test processes with low orders were used, differences between the best CR models and the MLR models were typically very small. In these situations the optimum CR models were often those that used all the latent variables, *i.e.*, the MLR solution. The largest differences were for the cases of very high noise, over-parameterized models and very little input excitation. Generally, the trends observed in FIR identification with CR were similar to those observed in ARX model identification, but were usually the less clear than in the FIR case. Some highlights of the simulations performed are given in the paragraphs that follow.

PRESS surface plots for a very high noise test are shown in Figures 5.25 and 5.26. This test was performed using model 7 in Table 5.2 as the true system. The model orders were correctly specified and the noise standard deviation was 40% of the process gain. The input was a PRBS generated using the filter in Table 5.1. In Figure 5.25, a PRBS with frequency content similar to that of the calibration data input was used to test the models. In Figure 5.26 a white noise signal was used for the test. Note how most of the best CR models for each continuum parameter are only slightly better than the MLR model. The best overall model (with a continuum parameter of 2 and 5 latent variables) has about a 35% smaller error in the PRBS test. Also, while the PRESS surface is similar to that seen in the FIR case, the "valley" is not nearly so well defined and is somewhat fragmented. There are also multiple local minima.

It is also apparent that the PRESS valley is more pronounced in Figure 5.26 where a random input was used to test the model. This is consistent with the FIR identification experience in the previous sections. A random input signal emphasizes higher frequencies more than a PRBS, and in the tests performed here CR tends to get models with better behavior in the high frequencies. Thus the PRESS valley is more pronounced when a random signal is used as a test.

The effect of process noise on the location of the best models in the CR parameter space is shown in Figure 5.27. Once again the true process is model 7 from table 5.2, and the system order was properly specified. Separate PRBS input sequences generated using the filter of Table 5.1 were used for calibration and testing. The figure shows the number of latent variables in the best model for each value of the continuum parameter.

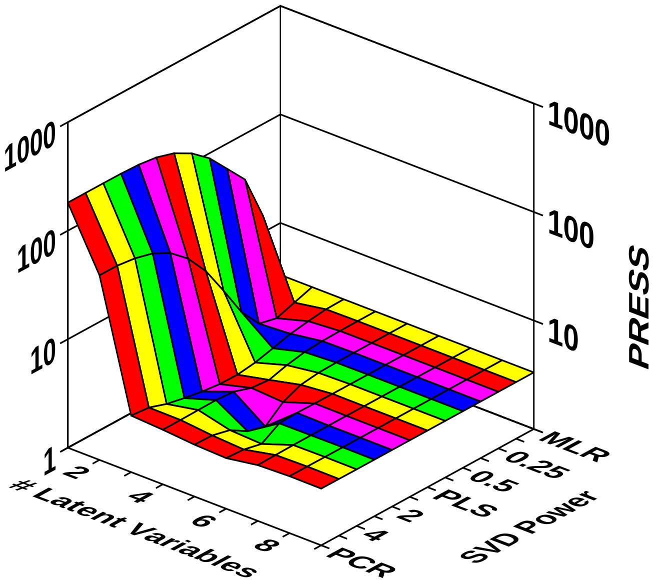

Figure 5.25. PRESS Surface for ARX Models Tested Against PRBS.

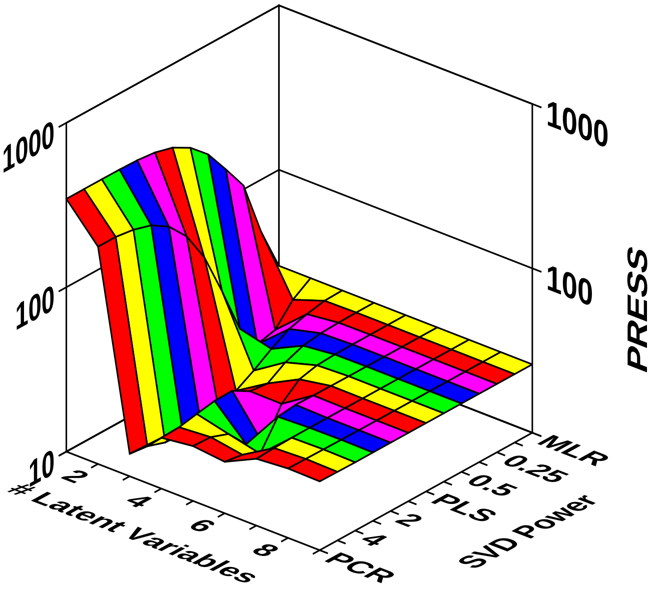

Figure 5.26. PRESS Surface for ARX Models Tested Against Random Input.

Note that while the general trend is as we expect, *i.e.*, there are fewer latent variables used at higher noise values, there is some irregular behavior in location of the best models. In particular, there are several instances where there are jumps to more latent variables as the regression technique moves towards MLR. This was seen in FIR identification only very rarely. In the ARX identification experiments performed here it was very common.

Tests with over-parameterized models showed similar behavior with added noise.

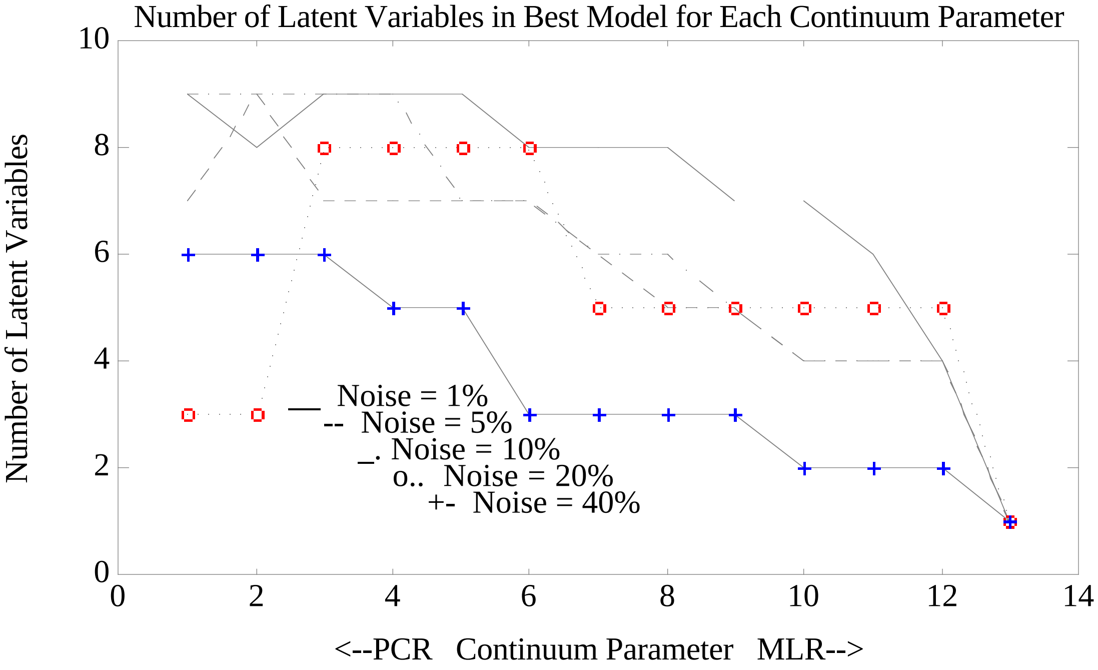

Figure 5.27. Number of Latent Variables in Best Models for Different Noise Levels, PRBS Test.

The effect of over-parameterization was also investigated. As mentioned previously, CR had a larger relative advantage over MLR with over-parameterized models. Even in highly over-parameterized models, such as specifying sixth over seventh order when the true process is fourth over fifth order, the difference is not great. The location of the best models in the CR parameter space tended towards more latent variables as higher process orders were specified, though not as fast as the order increased. For instance, if both the numerator and denominator orders were specified to be higher than the actual process order by 2, the number of latent variables at optimum might increase by 2, but not by 4. In general the trends in this series of experiments were not strong.

The effect of input excitation was similar to that found in FIR model identification. ARX models identified by MLR degraded more rapidly than those identified by CR as input excitation was decreased. The optimum models also tended towards fewer latent variables, though this effect was not as pronounced as with FIR. As an example, the PRESS surface for a very high noise case with very low input excitation is shown in Figure 5.28. The true model is number 7 from Table 2. The input for the calibration set was 7 random steps (4 up and 3 down) over a 500 sample time period. Process noise was 40% of process gain and the model orders were correctly specified. Only in these very adverse situations is the difference between the best CR models and the MLR model this pronounced.

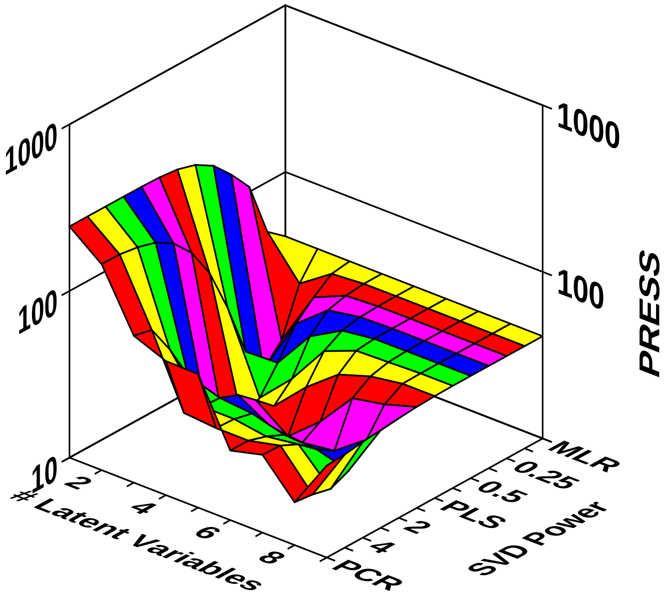

Figure 5.28. PRESS Surface for High Noise Low Input Excitation Case.

The Bode plots of the "best" models from the high noise low input excitation case are shown in Figure 5.29. The "Best Random Model" refers to the model that was best when tested against the true process using a white noise input sequence, while the "Best PRBS Model" refers to the model that is best when tested using a PRBS similar to the calibration input sequence. This figure demonstrates that the CR models are closer to the true response at most points, but not at all. Once again, the best model depends upon the criterion specified.

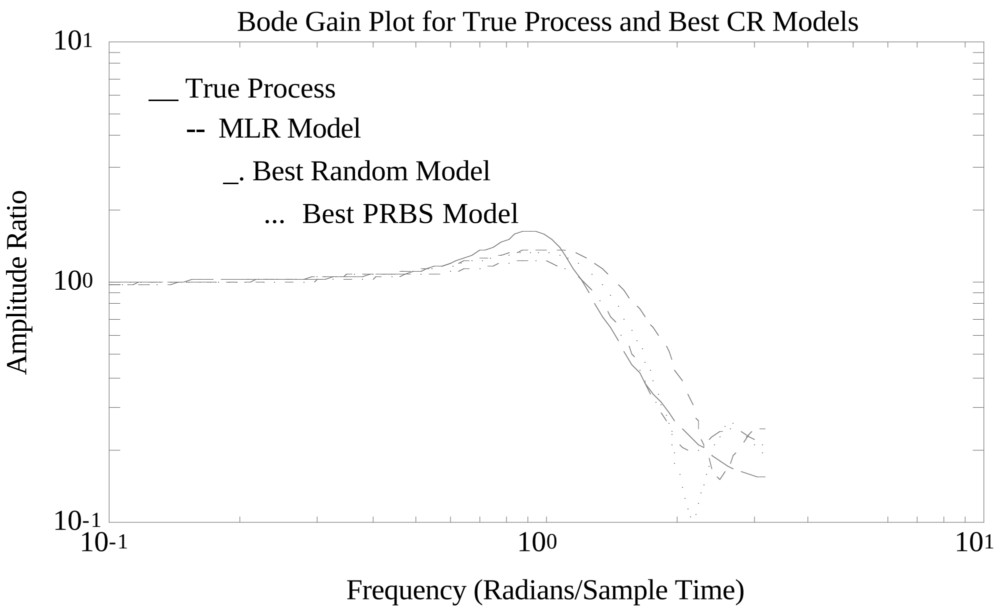

Figure 5.29. Bode Gain Plots for True Process and ARX Models Determined by CR and MLR.

When using CR for ARX model identification in practice, finding the optimum model may be quite difficult. It has been demonstrated that the minima can be quite shallow. These would be even more difficult to find when testing the models against noisy data sets. In this series of simulations we have had the luxury of testing the models against the true system. Even so, the location of the best model is often unclear.

In summary, it appears that CR offers advantages for ARX model identification only in very difficult situations, *e.g.*, when there is very little input excitation and a great deal of measurement noise. Under most circumstances, the relatively small potential gains in model accuracy combined with the difficulty in identifying the best models make existing methods more attractive.

5.4 PCR and PLS for Non-Linear FIR Model Identification

While PCR and PLS are linear techniques, it is easy to modify them for non-linear model identification. In this section the basic ideas behind non-linear PCR and PLS are discussed and an example of identification of a non-linear process model is given. It is not intended that this section provide a comprehensive treatment of the subject of non-linear biased regression. Instead, this example serves as an indicator of the potential of biased non-linear techniques.

5.4.1 Non-Linear Versions of PCR and PLS

A non-linear version of PCR can be implemented by proposing a non-linear relationship between the **X** block scores **T** and the output **y**. The form of the non-linear relationship can be arbitrary (such as a polynomial), or it may be arrived at through theoretical consideration of the process. Often, plots of the **X** block scores versus **y** will suggest a particular non-linear relationship.

As an example imagine that the output y is to be fit to the first k **X** block scores $\mathbf{T}_k$ using a second order polynomial. Thus it is proposed that

$$y = [t_1^2 \cdots t_k^2 \; t_1 \ldots t_k \; 1] \, \mathbf{b} \tag{5.18}$$

where **b** is a vector of regression coefficients to be determined. Taking some liberties with the notation, this can be rewritten for a collection of input output pairs as

$$\mathbf{y} = [\mathbf{T}_k^2 \; \mathbf{T}_k \; \mathbf{1}] \, \mathbf{b} \tag{5.19}$$

where it is understood that $\mathbf{T}_k^2$ indicates squaring the elements of $\mathbf{T}_k$, and that **1** indicates a vector of ones of appropriate length. Once the PCA decomposition of the **X** block has been obtained, the vector **b** can be estimated with the normal equations

$$\hat{\mathbf{b}} = ([\mathbf{T}_k^2 \; \mathbf{T}_k \; \mathbf{1}]'[\mathbf{T}_k^2 \; \mathbf{T}_k \; \mathbf{1}])^{-1}[\mathbf{T}_k^2 \; \mathbf{T}_k \; \mathbf{1}]'\mathbf{y} \tag{5.20}$$

This procedure can be contrasted with polynomial regression. If a second order polynomial fit is proposed, then the resulting relationship is

$$\mathbf{y} = [\mathbf{X}^2 \; \mathbf{X} \; \mathbf{1}] \, \mathbf{b} \tag{5.21}$$

where the notation is as in equation (5.20). Note that the matrix in brackets in (5.21) is no better conditioned than **X** alone, and is possibly worse. Thus, this problem is at least as unstable as the typical linear model problem.

Non-linear versions of PLS are accomplished in an analogous manner. A non-linear form is proposed for the PLS inner relationship. This inner relationship is normally calculated with equation (2.48), which assumes that the relationship between the **X**-block scores $\mathbf{t}_i$ and the **Y**-block scores $\mathbf{u}_i$ is a linear one. In non-linear PLS, a non-linear form such as that proposed in equation (5.19) is used, and the estimate of the coefficients are calculated from the normal equations as in equation (5.20). At each step the algorithm determines a vector in the **X**-block whose scores have the highest covariance with the **Y**-block residual. The scores are fit using the desired non-linear relationship, the estimates of **Y** (or its residual) are calculated and then subtracted off, forming the residual for the next step.

5.4.2 An Example Using Non-Linear PCR and PLS

As an example, non-linear PLS and PCR were applied to data from the tank apparatus described in Haesloop and Holt (1990a). This system consists of a tank with an outlet designed expressly so the tank outflow rate would be a highly non-linear function of the tank level. The output from this system is the voltage signal from the level measurement device. The input is the voltage signal to a pump which supplies water to the tank. Haesloop and Holt used this process to test a neural-net identification method. It is used here to test non-linear PLS and PCR for identifying a non-linear FIR model.

Haesloop performed a series of identification experiments in which the input was varied in a pseudo-random fashion and output data were collected. An example of such an experiment, which is used here as calibration data, is shown in Figure 5.30. The plotted input and output values are voltages.

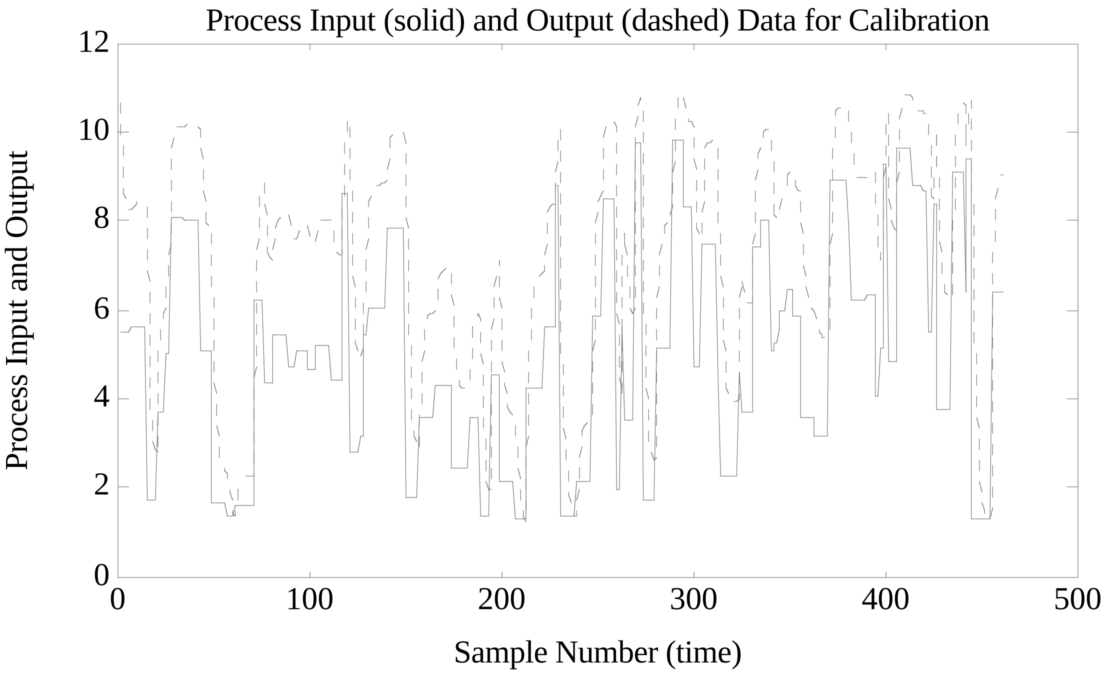

Figure 5.30. Non-Linear Tank Calibration Data.

After scaling the input/output data to zero mean and unit variance, several linear FIR models were identified using different numbers of past input values. The best results were when the last 6 values of the process input were used. At this point PLS and PCR models with polynomial inner relationships were used to model the process. Preliminary testing showed that the use of second-order polynomials provided better predictive models than higher order polynomials. A series of cross-validation tests using the calibration data showed that the PLS model with minimum prediction error (PRESS) was the one with 5 latent variables, while the best PCR model also retained 5 principal components. For comparison, a model was formed using second order polynomial regression as in equation (5.21). The estimate of **b** was obtained using the normal equations. Several attempts were also made to linearize the input/output data and fit a linear model. The best linearization method tested was simply to square the output voltage.

For comparison purposes, Haesloop used two neural nets with 6 input nodes, 6 hidden nodes and 1 output to perform an identical test using the same data. One of the neural nets had the standard configuration, while the other had Direct Linear Feed-through (DLF) terms, as described in Haesloop and Holt.

The fit of all of the regression models to the calibration data is shown for a 140 point segment in Figure 5.31. The fit errors (sum of squared residuals) for the entire calibration data set are given in Table 5.3. Note that the fit error of all the non-linear models and transformed linear model are about equal. The linear model fit is considerably worse. Of the non-neural net models the polynomial regression model has the best fit to the data, while overall the conventional (no DLF terms) neural net is best.

The models were then tested using data from another experiment. The purpose of this was to assess the predictive ability of the final models on a totally independent data set. This test data covered a narrower output range than the calibration data, thus this was an interpolation test. Each of the models was used to predict the process output and the PRESS was calculated for each model. The PRESS numbers are given in the right most column of Table 5.3. Of the regression models, the non-linear PLS model was found to have the smallest prediction error, followed by the non-linear PCR model and the polynomial regression. The linear model applied after the data transformation was somewhat worse than the non-linear models, and the strictly linear model had the largest error. The error of the conventional and DLF neural nets were the best overall, being about two-thirds that of the non-linear PLS model. Note that, even though the fit of the neural nets to the training data is about a factor of 5 better than for the regression models, the prediction error is only about 30% less.

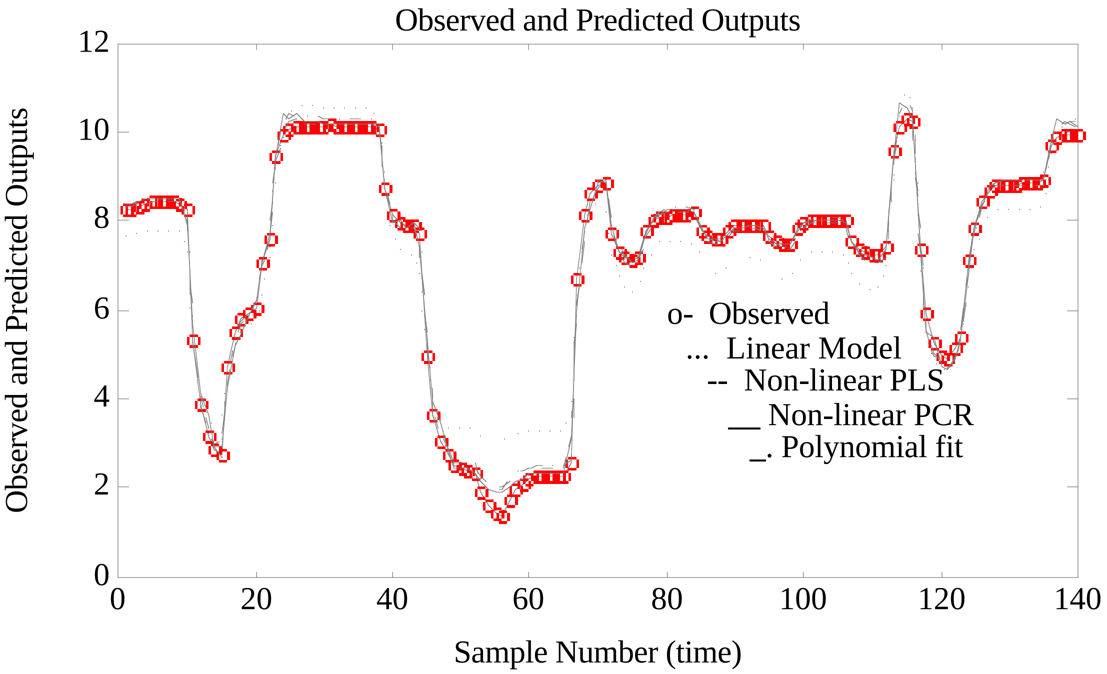

Figure 5.31. Segment of Calibration Data Set Showing Model Fits.

Table 5.3. Model Fit Errors and Prediction Errors for Non-Linear Process

Model	Fit Error	Prediction Error (PRESS)
Linear FIR	211.0	105.5
Non-linear PLS	20.6	3.7
Non-linear PCR	27.8	7.2
Polynomial Regression	20.1	8.4
Transformed Linear	27.5	11.8
Neural Net	3.8	2.5
DLF Neural Net	4.2	2.5

A segment of the actual and predicted data for the regression models is shown in Figure 5.32. While it is apparent that all models gave poor predictions under certain conditions, the model identified by non-linear PLS almost always lies closest to the actual process output. All of the models appear to be free of any fit artifacts, such as drastic overshoots after step changes. Such artifacts were observed in the non-linear PLS and PCR models using 6 latent variables. When using non-linear PLS and PCR, it is prudent to look for fit residuals with this type of odd behavior, which can be a sign of over-fitting.

This data set provides a classic example of the perils of over-fitting a data set. Note that (of the conventional models), while the polynomial model fit the calibration data the best, it did not predict the test data as well as either the PLS or PCR models.

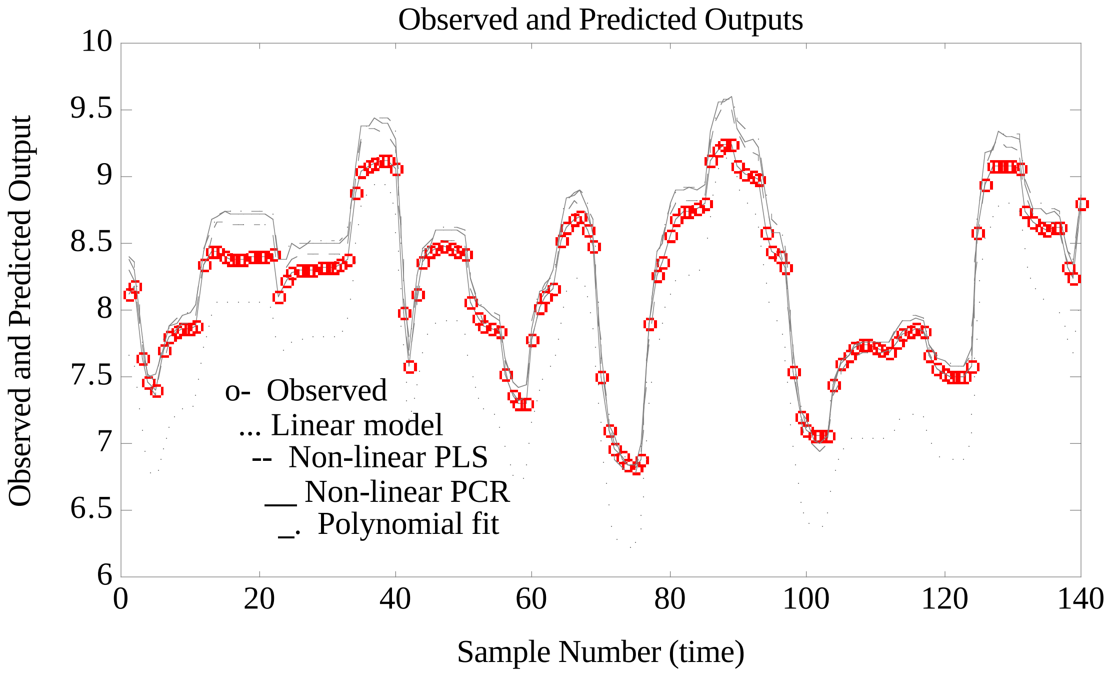

Figure 5.32. Test Data Set Showing Actual and Predicted Process Output.

The comparison to neural networks shows that the non-linear versions of PLS can be quite competitive with this technique. The neural network results are somewhat better, but all the non-linear models are quite good. The complexity of the neural network model (and its transportability) make it both harder to calibrate and more difficult to use. The computation time required for calibration of the neural network models was also very large relative to the PLS based models. Complete calibration and testing of the PLS models for this data set was done in ~5 minutes on a Macintosh IIx, while the neural network calibration required several hours on a high powered Apollo workstation. The much faster computation time of the PLS models would allow the modeler to become more involved in the model building process. Should massively parallel computers become readily available, however, the computational advantages of PLS would diminish.

5.5 Conclusions Concerning CR for Dynamic Modelling

In this chapter the effect of continuum regression on the identification of dynamic models has been considered both from a theoretical and a practical standpoint. It has been demonstrated that the theory developed here concerning the effect of eigenvector decomposition on input matrices accounts for the observed behavior of FIR models identified by CR.

There does not appear to be an advantage to using CR for the identification of ARX models, as might be expected. Under some very adverse circumstances, such as high process noise and low input excitation, this technique might produce better models than conventional techniques. However, because of the rather fragmented nature of the model error surface it may be difficult to locate the best models in the CR parameter space.

Finally, there does appear to be some incentive for the use of PLS with non-linear inner relationships. The technique can provide a quick alternative to more computationally-intensive techniques like neural networks, and provide very satisfactory results.

6.0 Conclusions

This work has considered several aspects of the process monitoring and modeling problem and how biased multivariate techniques can be used to attack them. The developments introduced in Chapters 3, 4 and 5 are reviewed in the sections that follow.

6.1 Process Monitoring with PCA

In Chapter 3 it was shown how PCA relates to the state-space model format. For processes with more measurements than states, a PCA model can be developed that captures the variations of the "immediately observable" states and leaves only the process noise in the PCA residuals. Statistical tests can be applied in order to determine whether the residuals from an evolving process violate the null hypothesis regarding their expected mean and variance. If so, this indicates a change in the underlying system. For example, sensor drift (added bias) is detected by a change in the mean of the PCA residuals and increased measurement noise results in changes in the variance of the residuals. Changes in individual sensors are detected by monitoring the PCA residuals from each sensor. In most cases the failed sensor can be identified as the one having the largest shift relative to its expected value.

It was demonstrated in Chapter 4 that, when actual process data is used, there is a trade-off between the sensitivity and specificity of PCA monitoring. As the first several PCs are added to the model, both the sensitivity and specificity improve. At some point, however, as the number of PCs retained in the process model is increased the specificity of the monitoring method suffers resulting in a model that is less effective overall.

In actual processes, the assumption that there are fewer process states than measurements is clearly an approximation. In many processes, however, the dynamics of the system are dominated by a few states and the effect of minor states is quite transient. Under these conditions, changes in the minor states can have the appearance of noise and can be treated as such. This is particularly true for sampled data systems, and in fact, an increase in the sample time (so that the decay time of the minor states is small relative to the sample time) can make the data appear as if there were fewer states. Because the assumption of fewer states than measurements is an approximation, in practice it may not be a simple matter to determine the "correct" number of PCs to retain in the process model. In this case, the number of PCs to retain becomes an optimization problem where the model developer would have to specify the desired weighting between sensitivity and specificity.

The most significant result of the low order approximation for the PCA models is that the residuals will be autocorrelated, though generally to a much smaller degree than the untreated variables. In this case, it is misleading to use statistics that assume independent observations to detect changes in the residuals. In this work, the rather brute force approach of using the observed mean and variance of data subgroups has been used to set limits. This option works effectively when the number of samples available for calibration is large, *i.e.* several thousand, as is the case in the examples included here. On the other hand it would be possible (and perhaps preferable) to use alternate procedures specifically designed to treat correlated observations. These approaches generally require a model of the autocorrelation, and one tests for significant deviations from this model.

While the PCA monitoring technique developed here has a limited range of applicability, *i.e.*, processes that can be adequately approximated as having more measurements than states, it has the advantage that it is very simple to develop from existing process data. It is not necessary to develop an entire dynamic model of the process. This is a distinct advantage in that there are many processes for which it is very difficult to develop an accurate dynamic model from either theory or observation. This is certainly the case with the LFCM studied here.

Advances in process instrumentation will undoubtedly result in more processes which meet the more measurements than states criteria, making PCA monitoring applicable. In particular, PCA monitoring will be quite useful for processes that utilize on-line spectroscopy where many wavelengths are considered.

6.2 Process Monitoring with PLS

Many of the conclusions regarding PCA monitoring can also true of PLS monitoring. This includes the tendency of the models to less specific as the order (the number of latent variables retained in the regression) is increased. There are also several important differences, as indicated in the paragraphs that follow.

It was shown that, even in the ideal case of a linear process with fewer states than measurements the PLS method did not produce uncorrelated residuals. This is a consequence of the fact that PLS regression models map some of the process state information into the residuals. In order to improve prediction, PLS emphasizes combinations of variables that have the highest covariance with the predicted variable. This results in biased estimates of the output. In the case of process monitoring, state information that is not very predictive may get left out of the model and end up in the residuals, resulting in autocorrelation. In spite of this, the PLS models, with residual limits adjusted to account for serial correlation, performed better in simulations than PCA models, particularly when the process input characteristics were similar in the test and calibration sets. Adding serial correlation to the process inputs degraded the performance of the PLS models, as would be expected due to the mapping of state information into the residuals.

A major performance advantage for PLS models is that models of mixed order may be developed. Collections of PLS models are not constrained to having the same number of latent variables in each model. In this sense, the PLS models can be thought of as an optimization of the number of latent variables to achieve the most robust prediction of each variable as opposed to the entire set of variables. It was demonstrated in Chapter 4 that mixed-order models can be a significant improvement over models of fixed order.

Unlike PCA models, PLS models do not suffer a large reduction in sensitivity as the model order is increased. In PCA models, this decrease in sensitivity is due to the fact that as PCs are added to the model, the subspace assigned to normal variation (driven by changes in process states) grows at the expense of the subspace of variation due to noise. Thus, unusual variations become more and more likely to fall into the subspace of normal variation. PLS models, on the other hand, only become less sensitive when the prediction error increases because of the addition of too many latent variables. To the degree that the prediction error does not increase greatly, the sensitivity of the PLS model does not suffer significantly.

The specificity of PLS models suffer as the number of latent variables is increased largely because the models tend to rely more heavily on fewer variables as more latent variables are added. In this sense, the models become less robust and prediction error (residuals) can become very large when sensors that are heavily weighted in the regression fail. Thus, the criteria for choosing the number of latent variables to retain in each of the individual PLS models cannot be based on prediction error alone, as this often results in models which are not very robust.

PLS models will apply under all conditions for which PCA models apply. Based on the results of simulations, particularly those employing actual process data, it would be expected that PLS monitoring would be more effective than PCA, particularly when the conditions are unfavorable. This would include situations where it is clearly an approximation to assume that the number of process measurements is much greater than the number of process states. One potential drawback of the PLS method, however, is that it appears to be more sensitive to input characteristics than PCA. When input characteristics change PLS model performance can be expected to suffer.

6.3 Continuum Regression for Process Modeling

In Chapter 5, the effect of continuum regression on the identification of FIR and ARX models was considered. Of particular interest was the frequency domain effects of the CR method on the identification of FIR models. It was shown here (both theoretically and by demonstrations) how PCR, one extreme of CR, has the effect of decomposing the input sequence into separate frequency components. Each successive PC in the PCR model represents a frequency with progressively less power in the input sequence. As models are built up from the PCs, they fit the behavior of the true process in the frequency ranges where there was the most power in the input signal. This frequency effect is also evident when other methods in the continuum are used to identify models, but, as would be expected, the frequency effect is not as pronounced as the methods move further from the PCR edge of the continuum.

In this work the CR predictive residual error surface was elucidated and explained in terms of the behavior of the underlying methods. The effects that process dynamics, noise, input excitation have on the location of the optimum model in the CR parameter space were all understood to be consistent with expectations based on knowledge of the methods. A particularly good example of this is the effect of process dynamics where it was shown that the true process dynamics had a large effect on the number of latent variables required for the model to adequately describe the process. It is clear from the simulations performed here that the CR method is a considerable improvement over MLR for the direct identification of FIR models.

It is much less clear whether CR offers any advantages for the identification of ARX models. It was shown here, both through theoretical considerations and through simulations that CR would probably not offer great advantages over existing methods except, perhaps, under very adverse circumstances. These circumstances would include cases of very high measurement noise, small amounts of input excitation and over parameterized models. It was also shown through simulation that the ARX model error surface, besides having shallow minima, could also have several local minima. This makes it uncertain that one could define a reliable method to find the optimum model in the CR parameter space.

Finally, the example use of PLS and PCR for identification of non-linear models demonstrates the potential of these techniques. It is argued here that the non-linear identification problem is at least as ill-conditioned as the corresponding linear problem, and therefore, that techniques designed to cope with ill-conditioning should offer some advantages. Furthermore, these techniques appear to be competitive with neural net techniques, at least in some applications. Computation time for the non-linear PLS based methods is very small relative to the amount required to train neural nets. These regression methods also have the advantage that, unlike neural nets, they produce an answer to a given calibration problem that is not dependent upon starting conditions. Thus, the methods are potentially more fool-proof since they require less judgement about whether the final solution is a good one. Though this work provides but a single example, it is evident that there is great promise in the non-linear biased techniques.

6.4 Suggestions for Future Work

While the process monitoring methods developed here show great promise, this work does not attempt to quantify the expected performance of these methods relative to existing techniques. More detailed studies should be performed that compare the PCA and PLS monitoring methods to methods requiring complete dynamic process models. One major advantage of the PCA/PLS methods is the ease of model development. Therefore, it would be particularly interesting to perform tests where the dynamic model of the process would have to be identified prior to development of the fault detection system. In these cases the errors in the dynamic model would be expected to have a significant impact on the detection system accuracy. It is the author's belief that it is in these situations where the methods developed here would have their biggest potential impact.

Future work in the area of CR for model identification would certainly include systems with multiple inputs. In this case, the frequency domain interpretation of the methods would not apply directly. Thus, it would be expected that some investigation into the effect of multiple inputs would be necessary. Finally, while the potential of biased non-linear identification was demonstrated here, very little was done here with the theory behind it. Further work in this area could be particularly fruitful.

References

Alt, F. B., Deutsch, S. J. and Walker, J. W., "Control Charts for Multivariate, Correlated Observations," in *1977 ASQC Technical Conference Transactions-Philadelphia*.

Anderson, T. W., *An Introduction to Multivariate Statistical Analysis*, 2nd. Ed., John Wiley and Sons, New York, 1984.

Åstrom, K. J. and Wittenmark, B., *Computer Controlled Systems: Theory and Design*, Prentice-Hall, New Jersey, 1984.

Barnes, S. M., Westsik, J. H., Jr. and Wise, B. M., "Instrumentation Concepts for Nuclear Waste Glass Melters," *Waste Management '85 Proceedings*, Tucson AZ 1985.

Basseville, M., "Detecting Changes in Signals and Systems-A Survey", *Automatica* 24, 309-326, 1988.

Beebe, K. R., and Kowalski, B. R., *Anal. Chem.* **59**, 1007A (1987).

Beebe, K. R., and Kowalski, B. R., "Nonlinear Calibration Using Projection Pursuit Regression: Application to an Array of Ion-Selective Electrodes," *Anal. Chem.* **60**, p. 2273, 1988

Beneke, M., Leemis, L. M., Schlegel, R. E. and Foote, B. L., "Spectral Analysis in Quality Control: A Control Chart Based on the Periodogram," *Technometrics* **30**(1), 1988.

Box, G. E. P., and Jenkins, G. M., *Time Series Analysis: Forecasting and Control*, Holden-Day, Oakland CA, 1976.

Burkholder, H. C., and Jarrett, J. H., compilers. *Nuclear Waste Treatment Program Annual Report for FY 1985*. PNL-5787, Pacific Northwest Laboratory, Richland, WA 1986.

Craig, R. J., "Normal Family Distribution Functions: FORTRAN and BASIC Programs," *J. Qual. Tech.*, **16**(4) Oct. 1984.

Crosier, R. B., "Multivariate Generalizations of Cumulative Sum Quality-Control Schemes," *Technometrics*, **30**(3) pp. 291-303, 1988.

Davis, Bruce, "Dynamic Modelling of a Slurry Fed Ceramic Melter Used in the Nuclear Waste Vitrification Process", Master's Thesis, University of Washington, 1989.

Deane, J. M. and MacFie, H. J. H., "Testing for Redundancy in Product Quality Control Test Criteria: An Application to Aviation Turbine Fuel," *J. Chemometrics*, Vol 3, 1989.

Diaz, H. and Desrochers, A. A., "Modeling of Nonlinear Discrete-time Systems from Input-Output Data", *Automatica*, Vol. 24, No. 5, pp. 629-641, 1988.

Frank, I. E., Feikema, J., Constantine, N. and Kowalski, B. R., "Prediction of Product Quality from Spectral Data Using the Partial Least-Squares Method," *J. Chem. Info. and Comp. Sci,* **24**(20), 1984.

Geladi, P. and Kowalski, B. R. "PLS tutorial" *Anal. Chim. Acta.,* **185**(1) (1986).

Geladi, P., "Notes on the History and Nature of Partial Least Squares (PLS) Modeling", *Journal of Chemometrics,* Vol. 2, 231-246 (1988)

Gibra, I. N., Recent Developments in Control Chart Techniques, *Journal of Quality Technology,* Vol. 7, No. 4, October 1975.

Haesloop, D. G. and Holt, B. R., "A Neural Network Structure for System Identification," *American Control Conference - 1990,* San Diego, CA, pg. 2460, 1990a.

Haesloop, D. G. and Holt, B. R., "A Combined Linear Non-Linear Neural Network for System Identification and Control," *AIChE Annual Metting,* Chicago, IL, 1990b.

Harris, T. J., "Statistical Process Control Procedures for Correlated Observations," Unpublished.

Healy, J. D. "A Note on Multivariate CUSUM Procedures," *Technometrics,* **29**(4), pp. 409-412.

Heikes, R. G., Montgomery, D. C. and Yeung, J. Y. H., "Alternative Process Models in the Economic Design of T^2 Control Charts," *AIIE Transactions*, **6**(1), p.55 (1974).

Himmelblau, D. M., *Fault Detection and Diagnosis in Chemical and Petrochemical Processes,* Elsevier Scientific Publishing Company, New York (1978).

Hoskuldsson, A., "PLS Regression Methods", Journal of Chemometrics, Vol. 2, 211-228 (1988).

Hotelling, H. “Multivariate Quality Control Illustrated by the Air Testing of Sample Bombsights,” in *Techniques of Statistical Analysis*, eds. C. Eisenhart, M. W. Hastay and W. A. Wallis, McGraw, New York, pp. 111-184 (1947).

Isermann, R., "Process Fault Detection Based on Modeling and Estimation Methods, A Survey," *Automatica,* **20**(4), pp. 387-409, (1984).

Jackson, J. E., and Mudholkar, G. S., "Control Procedures for Residuals Associated With Principal Component Analysis", *Technometrics* **21**(3), pp. 341-349, 1979.

Jackson, J. E., "Principal Components and Factor Analysis: Part 1-Principal Components," *J. Qual. Tech*., **13**(1), 1981.

Jackson, J. E., "Principal Components and Factor Analysis: Part 2-Additional Topics Related to Principal Components," *J. Qual. Tech*., **13**(2), 1981.

Jackson, J. E., "Principal Components and Factor Analysis: Part 3-What is Factor Analysis?" *J. Qual. Tech.*,**12**(4), 1980.

Jackson, J. E., "Statistics in Processing Control," *J. Appl. Photo. Eng.*, **2**(4), 1976.

Jensen, D. R. and Solomon, H., "A Gaussian Approximation to the Distribution of a Definite Quadratic Form," *J. Amer. Stat. Assoc.* Vol. 67, No. 340, pp. 898-902, 1972.

Kwakernaak, H. and Sivan, R., *Linear Optimal Control Systems,* Wiley-Interscience, New York 1972.

Kresta, J, MacGregor, J. F. and Marlin, T. E., "Multivariate Statistical Monitoring of Process Operating Performance", Submitted to Can. J. Chem. Eng., Feb. 1990.

Ljung, L., *System Identification Toolbox for use with MATLAB*, The MathWorks, Inc., 1988.

Ljung, L., *System Identification: Theory for the User,* Prentice-Hall, Inc., New Jersey, 1987.

Lorber, A. and B. R. Kowalski, "Parsimonious Regression", Center for Process Analytical Chemistry, University of Washington, Seattle WA., 1990.

Lorber, A. and Kowalski, B. R., "A Note on the Use of the Partial Least-Squares Method for Multivariate Calibration", *Applied Spectroscopy,* Vol. 42, No. 8, 1988.

Lorber, A., Veltkamp, D. J., and Kowalski, B. R., "Outliers Analysis in Multivariate Calibration," Center for Process Analytical Chemistry, University of Washington, Seattle WA., 1988.

Lorber, A., and Kowalski, B. R., "Estimating the Standard Error of Prediction for Multivariate Calibration Models," Center for Process Analytical Chemistry, University of Washington, Seattle WA., 1988.

Lorber, A., Wangen, L. E. and Kowalski, B. R., "A Theoretical Foundation for the PLS Algorithm," *J. Chemometrics* **1**, 19 (1987).

Lucas, J. M., and Crosier, R. B., "Fast Initial Response for CUSUM Quality-Control Schemes: Give your CUSUM a Head Start," *Technometrics*, 24, 199-205, (1982).

Lucas, J. M., and Crosier, R. B., "Robust CUSUM: A Robustness Study for CUSUM Quality Control Schemes," *Communications in Statistics-Theory and Methods*, 11, 2669-2687, (1982).

Lucas, J. M., "Combined Shewhart-CUSUM Quality Control Schemes," *Journal of Quality Technology*,14, pp.51-59 (1982).

Lucas, J. M., "Counted Data CUSUM's," *Technometrics*, 27, pp. 129-144, (1985).

MacGregor, J. F., "Statistical Process Control and Interfaces with Process Control," *Shell Process Control Workshop,* 1989.

MacGregor, J. F., "On-Line Statistical Process Control," *Chem. Eng. Prog.* October 1988.

MacGregor, J. F., "Multivariate Statistical Methods for Monitoring Large Data Sets from Chemical Processes," presented at AIChE Annual Meeting, San Francisco, November 6, 1989.

MacGregor, J. F., "Multivariate Statistical Process Control", presented at MADLUST 1990, Monday, July 2, 1990, Tromso, Norway.

MacGregor, J. F., Marlin T. E., Kresta, J. and Skagerberg, B., "Multivariate Statistical Methods in Process Analysis and Control," presented at CPC IV, South Padre Island, TX, February 17-22, 1991.

Malinowski, E. .R., "Theory of the Distribution of Error Eigenvalues Resulting From Principal Component Analysis with Applications to Spectroscopic Data," *J. Chemometrics* **1**, 33 (1987).

Malinowski, E. R., "Determination of the Number of Factors and the Experimental Error in a Data Matrix," *Anal. Chem*. **49**, 612 (1977).

Malinowski, E. R., "Statistical F-Tests for Abstract Factor Analysis and Target Testing," *J. Chemometrics* **3**, 49 (1988).

Malinowski, E. R., "Theory of Error in Factor Analysis," *Anal. Chem*. **49**, 606 (1977).

Mehra, R. K. and Peschon, J., “An Innovations Approach to Fault Detection and Diagnosis in Dynamic Systems,” *Automatica*, Vol. 7. pp. 637-640, 1971.

Mejdell T. and Skogestad, S., "Estimate of process outputs from multiple secondary measurements," 1989 American Control Conference Proceedings"

Miller, I., and Freund, J. E., *Probability and Statistics for Engineers,* 2nd Ed., Prentice Hall, New Jersey, 1977.

Minkkinen, P. and Karkkainen, J., "Application of SIMCA Pattern Recognition Method on Process Optimization," PTS-Symposium *Chemisch-Technologishe Probleme in Der Papierherstellung* vom 23. - 26.9. 1986 in Muchen

Montgomery, D. C. and Klatt, P. J., "Minimum Cost Multivariate Quality Control Tests," *AIIE Transactions*, **4**(2), p.103 (1972).

Moore, C., "Determining Analyzer Location and Type for Distillation Column Control," *Proc. 14th. Ann. Meeting. Fed. Anal. Chem and Spec. Soc.* 1987.

Moore, C., "What and Who is in Control? A Process Control Perspective on Statistical Process Control," *Shell Process Control Workshop,* 1989.

Morari, M. and Lee, J. H., "Model Predictive Control: The Good, the Bad and the Ugly," presented at CPC IV, South Padre Island, TX, February 17-22, 1991.

Morrison, D. F., *Multivariate Statistical Methods,* New York:McGraw-Hill (1967).

Naes, T., and Martens, H., "Principal Component Regression in NIR Analysis: Viewpoints, Background Details and Selection of Components," *J. Chemometrics*, Vol. 2, 1988.

Palazoglu, A. and Romabnoli, J. A., "Nonlinear Control of Chemical Processes Via Exact and Approximate Methods," presented at the American Institute of Chemical Engineers Annual Meeting, Chicago Ill., November 1990.

Pignatiello, J. J. and Kasunic, M. D., "Development of a Multivariate CUSUM Chart," in *Proceedings of the American Society of Mechanical Engineers' Computers in Engineering Conference,* eds. R. Raghavan and S. M. Rohde, New York:American Society of Mechanical Engineers, pp. 427-432 (1985).

Pignatiello, J. J., Jr., Runger, G. C. and Korpela, K. S., "Truly Multivariate CUSUM Charts," Working Paper 86-024, University of Arizona, Systems and Industrial Engineering Dept. (1986).

Priestly, M. B., Rao, T. S. and Tong, H., "Applications of Principal Component Analysis in the Identification of Multivariable Systems," *IEEE Trans. Auto. Control,* Vol. AC-19, No. 6, December 1974.

Prett, D. M., Skrovanek, T. A. and Pollard, J. F., "Process Identification - Past, Present, Future", *Shell Process Control Workshop,* 1989.

Ricker, N. L., "The Use of Biased Least-Squares Estimators for Parameters in Discrete-Time Pulse-Response Models," *Ind. Eng. Chem. Res*., **27**(2), p.343, 1988.

Ricker, N. L., "Multivariate Statistical Process Control: Analogy to the Kalman Filter", Submitted to *AIChE Journal,* March 1990.

Ricker, N. L. and Douglas, D. D., "Fault Detection In Dynamic Systems: Comparison of PCA and Maximum Likelihood Ratio Methods," *AIChE 1990 Annual Meeting,* November 1990.

Rivera, D. E., Pollard, J. F., Sterman, L. E. and Garcia, C. E., "An Industrial Perspective on Control-Relevant Identification" American Control Conference, San Diego, CA, 1990a.

Rivera, D. E., Pollard, J. F. and Garcia, C. E., "Control-Relevant Parameter Estimation Via Prediction-Error Methods: Implications for Digital PID and QDMC Control", presented at AIChE Annual Meeting, Chicago Il, 1990b.

Rutan, S. C. "Kalman Filtering Approaches for Solving Problems in Analytical Chemistry," *J. Chemometrics* 1987?

Sage, A. P. and White, C. C., III, *Optimum Systems Control, Second Edition,* Prentice-Hall, New Jersey, 1977.

Sharaf, M. A., Illman, D. L., and Kowalski, B. R., *Chemometrics*, John Wiley and Sons, New York (1986).

Shewart, W. A., *Economic Control of Quality of Manufactured Products,* D. Van Nostrand Co., New York, 1931.

Soderstrom, T. and Stoica, P. G., *Instrumental Variable Methods for System Identification*, Springer-Verlag, New York (1983)

Srivastava, M. S. and Worsley, K. J., "Likelihood Ratio Tests for a Change in the Multivariate Normal Mean," *Journal of the American Statistical Association*, 81, pp. 199-204, (1985).

Stoica, P., Eykhoff, P., Janssen, P., and Soderstrom, T., "Model-structure selection by cross-validation," *Int. J. Control*, Vol. 43, No. 6, pp 1841-1878, 1986.

Stone, M. and Brooks, R. J., “Continuum Regression: Cross-validated Sequentially-constructed Prediction embracing Ordinary Least Squares, Partial Least Squares, and Principal Components Regression”, Unpublished.

Strang, G., *Linear Algebra and Its Applications*, Academic Press New York, 1980.

Vance, L. C., "A Bibliography of Statistical Quality Control Chart Techniques, 1970-1980", *Journal of Quality Technology*, **15**(2), pp. 59-62, 1983.

Veltkamp, D., *User's Guide for PCA Modeling Program Version 1.0*, The Center for Process Analytical Chemistry, University of Washington, Seattle WA, 1989.

Veltkamp, D. J., Kowalski, B. R., Ricker, N. L. and Wise, B. M., "Multivariate Statistical Process Control Using Principal Component Analysis", submitted to *Journal of Chemometrics*,.1990.

Vogt, N. B., "Principal Component Variable Discriminant Plots: A Novel Approach for Interpretation and Analysis of Multi-Class Data," *J. Chemometrics*, Vol 2, 1988.

Wadsworth, H. M. Jr., Stephens, K. S. and Godfrey, A. B., *Modern Methods for Quality Control and Improvement*, John Wiley and Sons, New York (1986).

Wangen, L. E. and Kowalski, B. R., "A Multiblock Partial Least Squares Algorithm for Investigating Complex Chemical Systems," *J. Chemometrics*, Vol 3, 1988.

Willsky, A. S. "A Survey of Design Methods for Failure Detection in Dynamic Systems," *Automatica*, **12**, pp. 601-611, (1976).

Wise, B. M. and McMakin, A. H,."A Statistical Technique for Analyzing Data from Liquid-Fed Ceramic Melters, " *Glass Industry*, accepted March 1988.

Wise, B. M., Veltkamp, D. J., Davis, B., Ricker, N. L. and Kowalski, B. R., "Principal Components Analysis for Monitoring the West Valley Liquid Fed Ceramic Melter," *Waste Management '88 Proceedings*, Tucson AZ 1988.

Wise, B. M. and Ricker, N. L,."Feedback Strategies in Multiple Sensor Systems" in *AIChE Symposium Series*, No. 267, Vol 85, 1989..

Wise, B. M., Ricker, N. L. and Veltkamp, D., "Upset and Sensor Failure Detection in Multivariate Processes," *AIChE 1989 Annual Meeting*, November 1989.

Wise, B. M., Ricker, N. L., Veltkamp, D. J. and Kowalski, B. R., “A Theoretical Basis for the Use of Principal Component Models for Monitoring Multivariate Processes”, *Process Control and Quality*, Vol 1, Number 1, pages 41-51, 1990.

Wise, B. M., and Ricker, N. L., "The Effect of Biased Regression on the Identification of FIR and ARX Models," *AIChE 1990 Annual Meeting*, Nov. 1990.

Wise B. M., and Ricker, N. L., "Recent Advances in Multivariate Statistical Process Control: Improving Robustness and Sensitivity," Submitted to *IFAC Symposium on Advanced Control of Chemical Processes*, Toulouse, France, October 1991.

Wise, B. M., Veltkamp, D. J., Ricker, N. L., Kowalski, B. R., Barnes, S. M. and Arakali, V., "Application of Multivariate Statistical Process Control (MSPC) to the West Valley Slurry-Fed Ceramic Melter Process," *Waste Management '91 Proceedings*, Tucson AZ 1991.

Wold, S., “Pattern Recognition by Means of Disjoint Principal Components Models,” *Pattern Recognition*, Vol. 8, pp. 127-139, 1976.

Wold, S., S. Hellberg, T. Lundstedt and M. Sjostrom, "PLS Modeling with Latent Variables in Two or More Dimensions," *Frankfurt PLS Meeting*, Sept. 1987.

Woodall W. H. and M. M. Ncube, "Multivariate CUSUM Quality Control Procedures", *Technometrics* **27**(3), pp. 285-292, 1985.

Appendix 1. Dynamic Process Investigation with PCA

It is the purpose of this appendix to document the accumulated experience that the author has gained applying the PCA method to dynamic process data. Unfortunately, the process data available for this work has been limited. For this reason, it is difficult to reach definite conclusions about the general applicability of the method to other processes. However, there are several instances where the results of the analysis have been found to be quite instructive, and therefore, useful. There have also been some rather ambiguous results that serve as a warning to those that would try to read too much into PCA models.

The examples in this appendix come from two LFCMs of slightly different configuration. One of these is the West Valley melter described at the end of Chapter 2. The other is a slightly smaller melter (known as the Pilot Scale Ceramic Melter or PSCM) with a two electrode design that is located at Battelle Pacific Northwest Laboratories in Richland, Washington. With the exception of the electrode design, the two melters are quite similar.

To illustrate both the utility and the potential pitfalls of applying PCA to dynamic data, the initial PCA study on melter foaming (Wise and McMakin 1987) will be reconsidered. The results of this study will be combined with some more recent analysis of the same data. In the second section some results from the West Valley LFCM will be considered.

A1.1 PCA Study of Melter Foaming

The major objective of the PCA melter foaming study was to determine if PCA could be used to identify LFCM glass foaming events. Melt foaming has been a cause of concern since the early development of LFCMs (Burkholder and Jarrett, 1986). During normal operation, melter feed forms a "cold cap" in the center of the molten glass pool. The slurry often pools in the center of this cold cap, then boils continuously, evaporating water from the slurry. During a foaming episode, gas, primarily oxygen, is released in the melt. The gas bubbles to the surface of the melt, forming a stable foam that often completely obscures the cold cap.

Foaming is caused by dissolved water and oxygen in the glass melt, and for this reason highly oxidized glasses have a higher potential for foaming than reduced glasses. This is because oxidized glasses release oxygen when changed to a lower oxidation state. Foaming incidents tend to be "runaway"; once foaming begins, it forms an insulating layer on the melt surface which causes the melt temperature to rise rapidly. The temperature increase, in turn, promotes a change to lower oxidation states for some of the glass components, which causes more oxygen to be released. The result is a loss of process control, which can only be remedied by shutting off the flow of the melter feed and reducing the melter power input until the glass surface has cleared of foam. Such interruptions result in process downtime and decreased glass production rates. Foaming episodes also have the potential to damage equipment. Unfortunately, the mechanism that triggers foaming is not entirely understood.

Before foaming episodes can be controlled it is necessary to determine what happens prior to and during foaming. PCA was used to identify the processing trends associated with foaming. Data for this study was obtained during a 300-hr demonstration test at PNL in May and June of 1985 using the PSCM. Several foaming incidents were detected during the test, both by visual observation of the melt surface and by a prototype pneumatic sensor designed specifically to detect the presence of foam above the cold cap.

A1.1.1 Review of Original Foaming Study

In the original study, seven variables were chosen for the analysis: glass temperatures at five separate depths, the glass resistance, and the melter power. Due to software limitations, one hundred data points were selected for the analysis: 1 data point for each 3 hour period of the test and 5 data points from each of the 2 major foaming episodes that occurred during the test.

Because of the large differences in the variance of the variables, the data was autoscaled[10] before being subjected to PCA. The loadings plot of this data, Figure A1.1, shows that most of the variance in the data set (43%) was caused by changes in the bulk glass temperatures shown along the first principal component (horizontal axis). The melter power input, resistance and the surface temperature loaded most heavily into the second eigenvector (27% of the variance). Thus 70% of the total variance in the data set was captured in the first two eigenvectors.

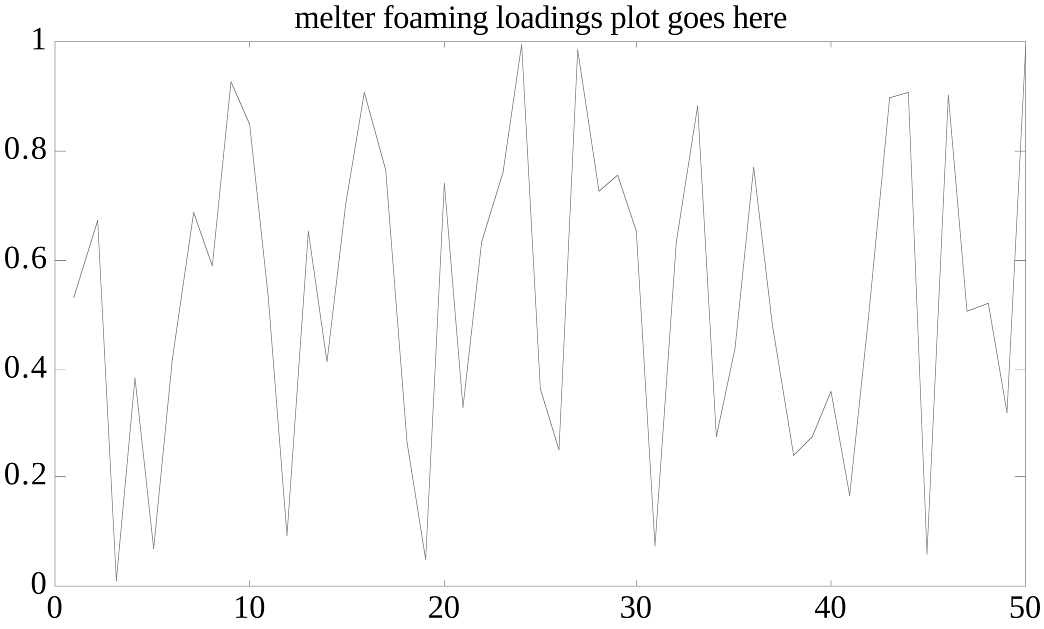

Figure A1.1. Loadings on First Two Principal Components in Melter Foaming Study.

The scores plot for the data is shown in Figure A1.2. The plot shows that the operating data clusters around the center point in a seemingly random "shotgun blast." However, the data points taken during foaming (shown as triangles) all lie in the upper

---

[10] The effect of other scalings for this data is considered in Section A1.1.2. The effect of scaling on a West Valley LFCM data set is also considered in Section A1.2.

right-hand corner. Even more significant, data points that were taken as much as 6 hours prior to foaming incidents (shown as squares) are located just to the left and below the foaming points.

The fact that this technique separates normal operating data from foaming and prefoaming data suggested that the method had potential to as a real-time process control aid, provided that it could be determined what caused the two groups of data points to be separated on the scores plot. In this case the scores plot could be interpreted by using the information on the loadings plot.

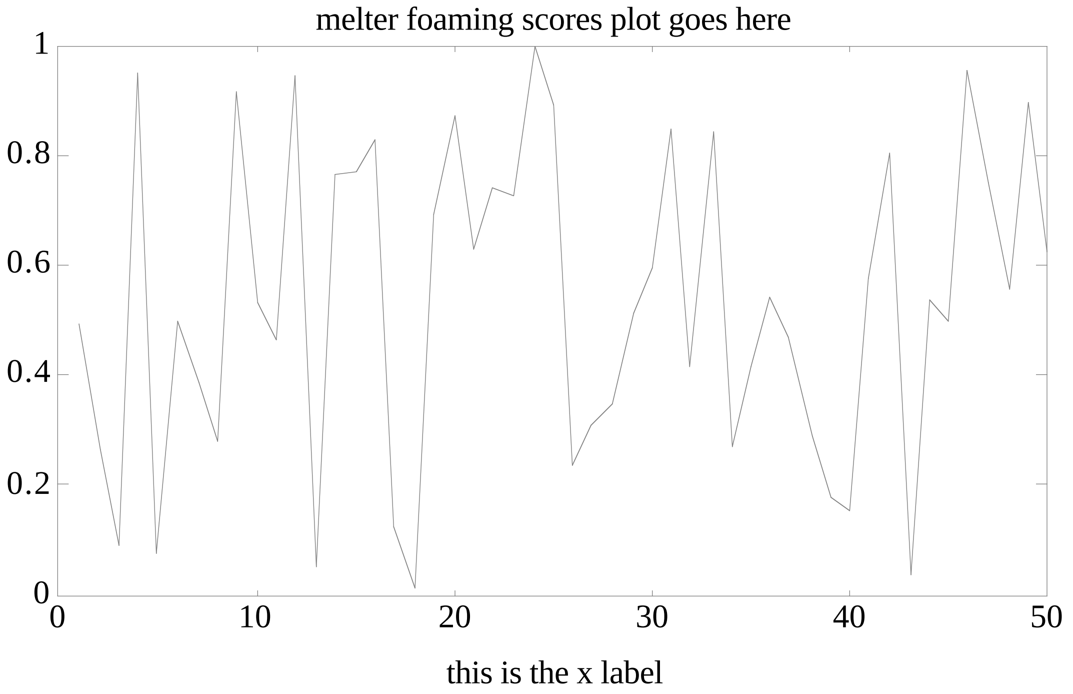

Figure A1.2. Scores on First Two Principal Components in Melter Foaming Study.

On the scores plot, the prefoaming are high (strongly positive) on both the first and second principal component . A score that is high on the first PC indicates that the sample was taken when the bulk glass temperature was above average: the temperature variables in the bulk of the glass all load positively into the first PC. A score that is high on the second PC indicates that the power is low relative to the surface temperature: power is loaded negatively into the second PC and the surface temperature is loaded positively. By the onset of foaming, the operating point has shifted further upward and to the right into the cross-hatched region of the plot, indicating yet a lower power relative to surface temperature, and somewhat hotter bulk glass. When the foaming gets underway, the operating point moves to the right (not apparent in the figure), indicating hotter bulk glass, and downward, indicating hotter surface temperatures and lower power as the melter controller begins to turn down the power.

Based on the principal component analysis it is possible to observe clearly the events which precede foaming. In the "normal" operating range, the temperature gradient is large between the bulk of the glass and the surface, and the power is high relative to the bulk glass and surface temperatures. In the prefoaming stage, however, the bulk of the glass (temperatures 2, 3 and 4) has become hotter, (causing a move to the right) and the ratio of the power level to the bulk and surface temperatures has decreased (causing a move upward). Temperature gradients are generally smaller than during "normal" operation. At the onset of foaming, the power-to-surface temperature ratio decreases further and the bulk glass temperature continues to increase.

The major question concerns the physical relationship between the reduced temperature gradient, lower power requirement and the onset of foaming. By using fundamental knowledge about the system and other data recorded by operators it is possible to theorize about the fundamental causes of foaming. Generally, foaming occurs during periods of high feed rate (a variable which was unavailable in this study) when the cold cap coverage is large. The results of numerical simulations such as that done by Davis (1988) agree with the observations of melter operators that power requirements actually decrease during high feed rate periods when the cold cap is large. This is due to the insulating properties and radiative heat transfer shielding effects of the cold cap. Thus, the cold cap should have the effect of flattening temperature gradients underneath it since heat transfer through the glass surface is reduced. It is logical that the mass transfer rate between the bulk glass and the surface would also be reduced under conditions of a flat temperature gradient. This would be due in part to the decreased convection in the glass when the driving force (temperature gradient) is reduced, and because the cold cap itself may form an impenetrable barrier to mass transfer. A decreased mass transfer rate would cause dissolved gasses, which normally are released gradually, to build up in the melt until their concentration increases enough to trigger a sudden, large release.

A1.1.2 Re-analysis of Foaming Study Data

One problem with PCA becomes apparent when the analysis is repeated with a different subset of the same data. In this case there were no software limitations on the number of samples for analysis so all of the data was used, with the exception of periods during manual power shutdowns. The data sampling rate was once every two minutes. It was found that it was difficult to obtain projections that effectively separated foaming, prefoaming and "normal" data[11]. Finally, after trying several scaling options, (scaling effects will be discussed further in the next section), a projection was found (the first versus third principal component) that has the same discriminatory power as the projection found in the initial study.

The scaling chosen was a variation of mean centering: all variables were mean centered, then the variance of the power and resistance variables was adjusted to be approximately equal to the average variance of the other variables. Mean centering of the temperature variables was chosen because it emphasizes temperatures which have greater variance. It was reasoned that these variables would be better indicators of variations in the melter system due to their greater "signal to noise ratio." However, mean centering alone

---

[11] In an unpublished study done for WVNS two other methods were used to create a projection that was useful for identifying foaming conditions. One of these methods was VARIMAX rotation; the other was a projection/rotation based on the means of the three process data subgroups of normal, prefoaming and foaming periods. The results of this study generally supported the interpretations arrived at during the initial PCA study, *i.e.,* projections were found that separated the data into the three regions and the original variables loaded into these projections in approximately the same way as in the original study.

leaves the power and resistance variables with very small variances relative to the temperatures and would greatly de-emphasize them in the PCA. Therefore, the variances of the power and resistance were adjusted to be equal to the average variance of the temperature variables based on the reasoning that these variables are as good of indicators of conditions as the "average" temperature variable.

The loadings of the first versus third PC are shown in Figure A1.3, the corresponding scores plot is shown in Figure A1.4. The stars on the scores plot correspond to the 10 points (collected over 20 minutes) taken during each of the three confirmed foaming incidents prior to manual power adjustment. The circles indicate the 10 points prior to each of the incidents.

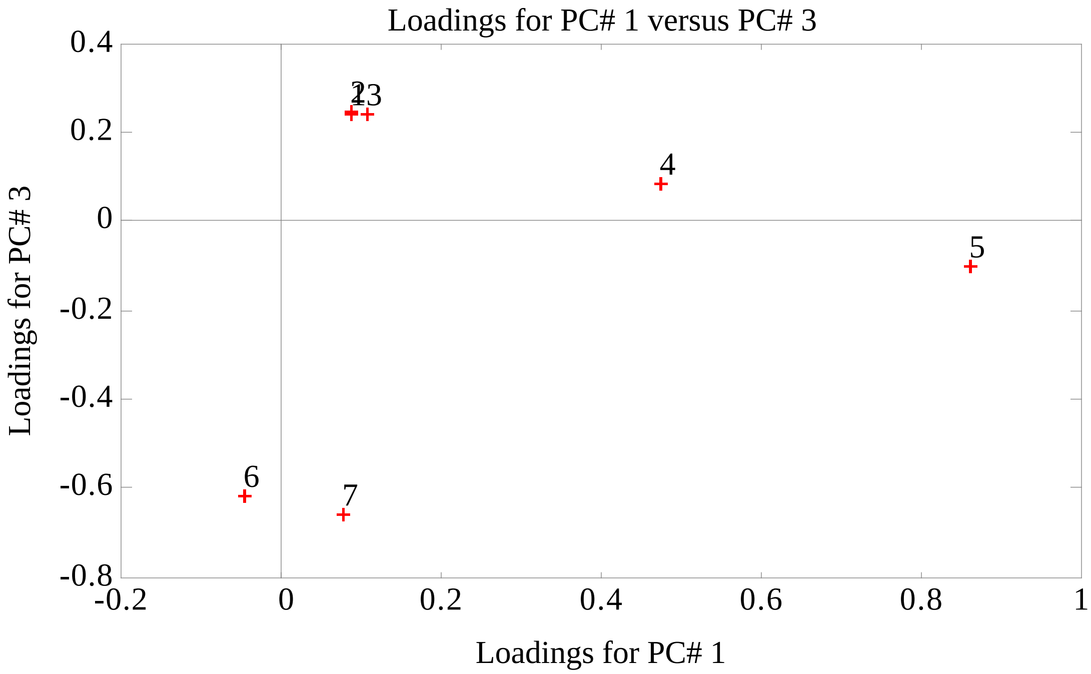

Figure A1.3 Loadings for First versus Third Principal Component for Revised Melter Foaming Study.

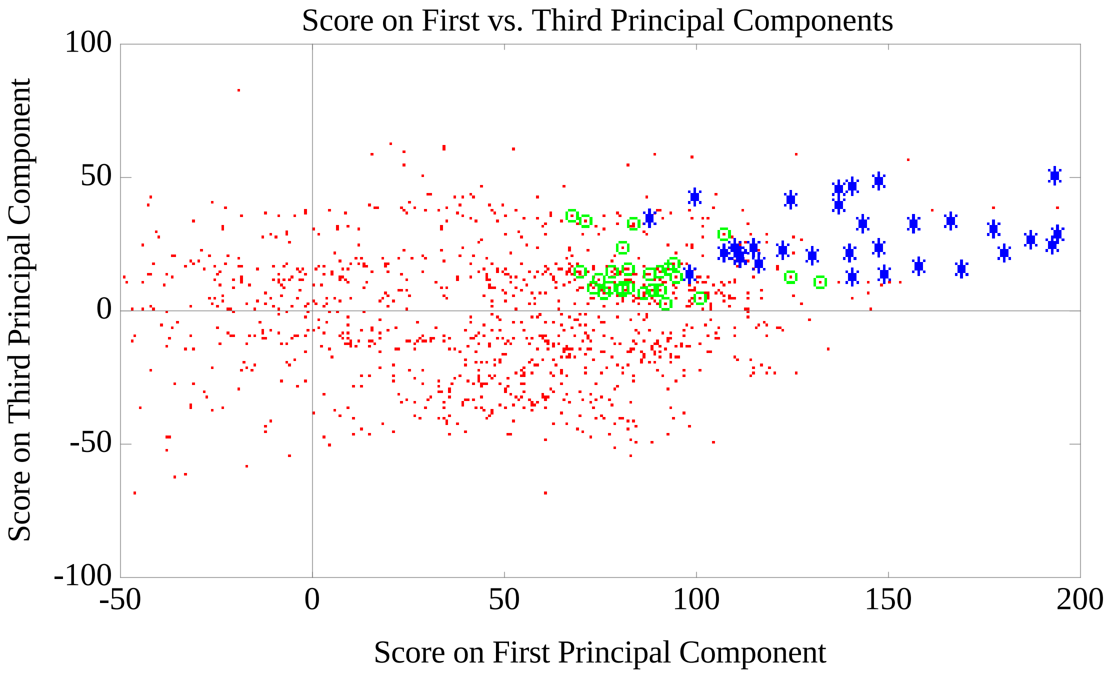

Figure A1.4. Scores on First versus Third Principal Component for Revised Melter Foaming Study.

Note that the loadings in Figure A1.3 look quite different from those shown in Figure A1.1. The scores plot, however, splits the operating data in a fashion similar to that of the initial study. The interpretation that prefoaming times are associated with a low power to bulk glass temperature ratios is borne out, but the ratio of the power to the surface temperature is apparently not critical for separating the points. The temperature gradient argument also appears intact as the prefoaming points are positive on the first PC which is most strongly indicative of surface temperature. It seems less likely, however, that the gradient theory would be arrived at from this set of PCs. Separate calculations of the gradient, however, have shown that it does correlate with foaming incidents. Furthermore, note that the correlation of melter glass resistance with foaming is entirely different in this analysis than in the original study. Apparently, resistance has little or no bearing on the foaming phenomena.

The sample times of the points mixed in with the circles and stars in Figure A1.4

were determined. It was found that these points came primarily from times either preceding (by periods of more than 20 minutes) or immediately after melter foaming. A small minority of the points are from times where foaming was not observed visually, but might be suspected based on the temperature record. It would appear that samples in the region on the plot are a necessary, but not sufficient condition for foaming to occur. This makes sense from a physical standpoint. Glass melts that have recently foamed certainly have a decreased propensity to foam again soon, as most of their dissolved gasses have been released. Reduced glasses, of course, will not foam under any conditions.

This example has shown that the selection of the proper data for test and training sets is critical to an analysis of this type. Apparently, this author was fortunate in the original analysis to have chosen data that resulted in axes that were good discriminators between foaming and non-foaming data. It took a considerably larger effort to arrive at a projection for separating the foaming and "normal" data in the second case. This illustrates that one cannot, in general, expect PCA to provide the optimum projection for separating samples into the desired classes. In order to obtain the best results, other techniques must sometimes be used. In all cases proper selection of the training data sets is very important, as is sound engineering judgement concerning scaling.

Interpretation of the PCA results, particularly the PC loadings, can also be difficult. It is important to try to attach physical interpretations to the factors and test these interpretations to the extent that the data and theory will allow. The need for an understanding of the physical processes underlying the phenomena under investigation cannot be stressed enough. The PCA results should be taken more as suggestive, rather than as a final answer. Furthermore, it is important to remember the difference between correlation and causality.

A1.2 PCA Analysis of SF-11: The Effect of Scaling

As another example of using of PCA for process data analysis, an examination of data from the West Valley melter taken during a run designated SF-11 is presented here. Specifically, the issue of data scaling will be considered. This will be illustrated by comparing the PCA results from autoscaled versus mean centered data.

The SF-11 run was performed in September, 1989. Run data is available at 5 minute intervals for 7 days, of which the continuous feeding portion of the run lasted approximately 5.5 days. Only the temperature information will be considered in this analysis, thus there are 20 variables as indicated in Table 2.1. A 500 point segment of the run, representative of nominal operating conditions, was selected for building the PCA model.

Because all of the variables are of a similar nature and would be expected to have similar noise characteristics, mean centering would seem to be the logical choice for scaling. This implies, however, that the variables with larger variances, which in this case will be the near surface and plenum temperatures, are in some sense more important than the variables with smaller variances, such as those in the bulk glass. On the other hand, if one believed that all variables were equally important over their observed range, (i.e. that a change from -2 to +2 standard deviations was equally significant for all variables), then autoscaling would be appropriate. In this case, it is hard to say beforehand, so the analysis is done below both ways and the results are compared.

The variance captured by the first 10 PCs in the mean centered PCA model is given in Table A1.1 below. As the table shows, this data set is very directional, with 76.6% of the variance being along the first PC. After the first 4 PCs the size differential between successive PCs becomes very small. All PCs after the tenth accounted for 0.22% of the variance or less.

Table A1.1 Percent Variance Captured by PCA Model of SF-11 Data using Mean

Centering.

PC#	Eigenvalue	%Variance	%Total Variance
1.0000	69.2271	76.6478	76.6478
2.0000	8.3099	9.2006	85.8484
3.0000	5.1848	5.7405	91.5889
4.0000	2.3246	2.5738	94.1627
5.0000	1.3395	1.4831	95.6458
6.0000	1.2346	1.3669	97.0127
7.0000	0.9534	1.0556	98.0683
8.0000	0.5219	0.5778	98.6461
9.0000	0.3425	0.3792	99.0253
10.0000	0.2712	0.3003	99.3256

The loadings for the first 4 PCs are shown in Figures A1.5 through A1.8. In this case the first three loadings appear to be interpretable. In the first PC the near surface temperatures (variables 6 to 9 and 16 to 19) load strongly. It is known from experience that this PC is strongly correlated with the melter level. As the melter level changes the cold cap slides up and down the thermowells. This causes the region of steep temperature gradient in the cold cap to pass over the near surface thermocouples and causing these temperature variables to vary strongly.

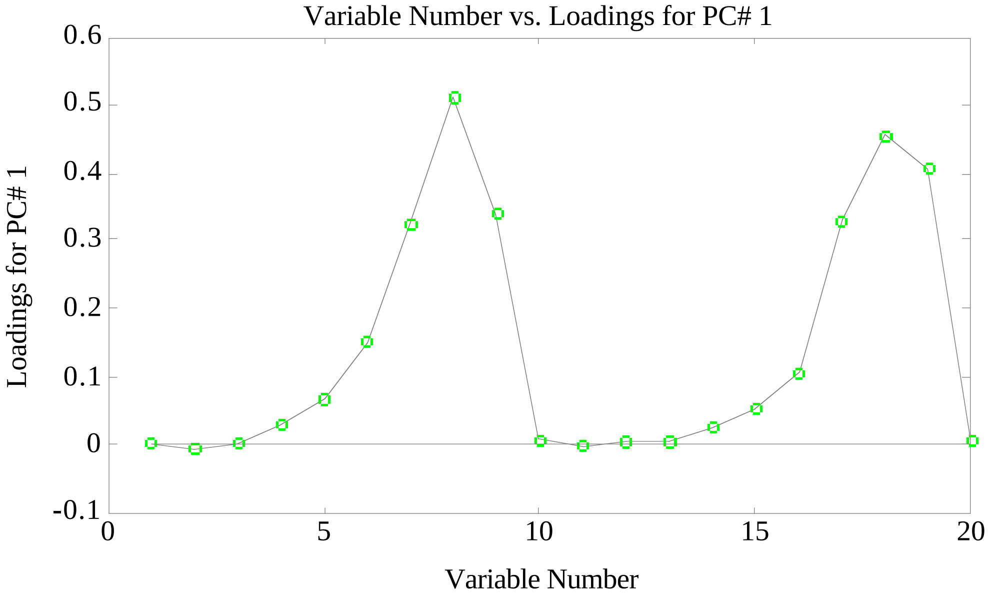

Figure A1.5. Loadings for First PC of Mean Centered SF-11 Data.

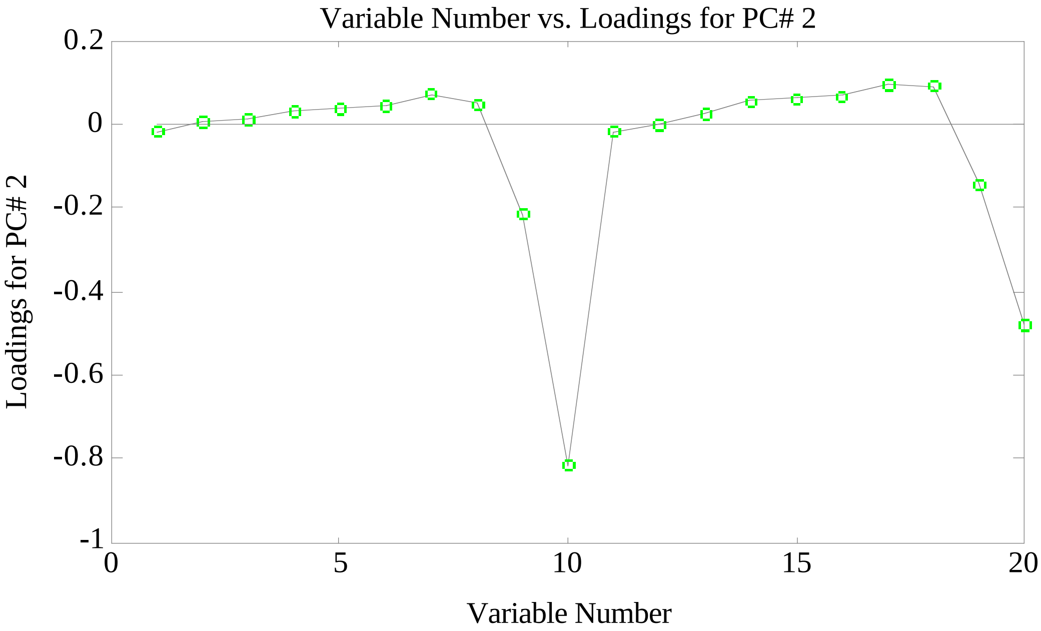

Figure A1.6. Loadings for Second PC of Mean Centered SF-11 Data.

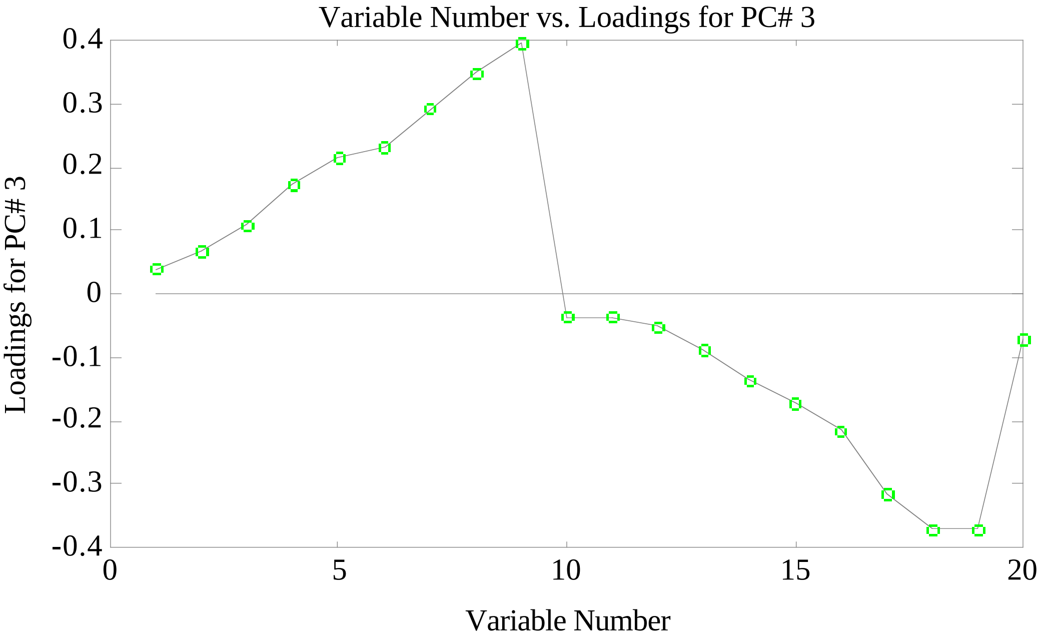

Figure A1.7. Loadings for Third PC of Mean Centered SF-11 Data.

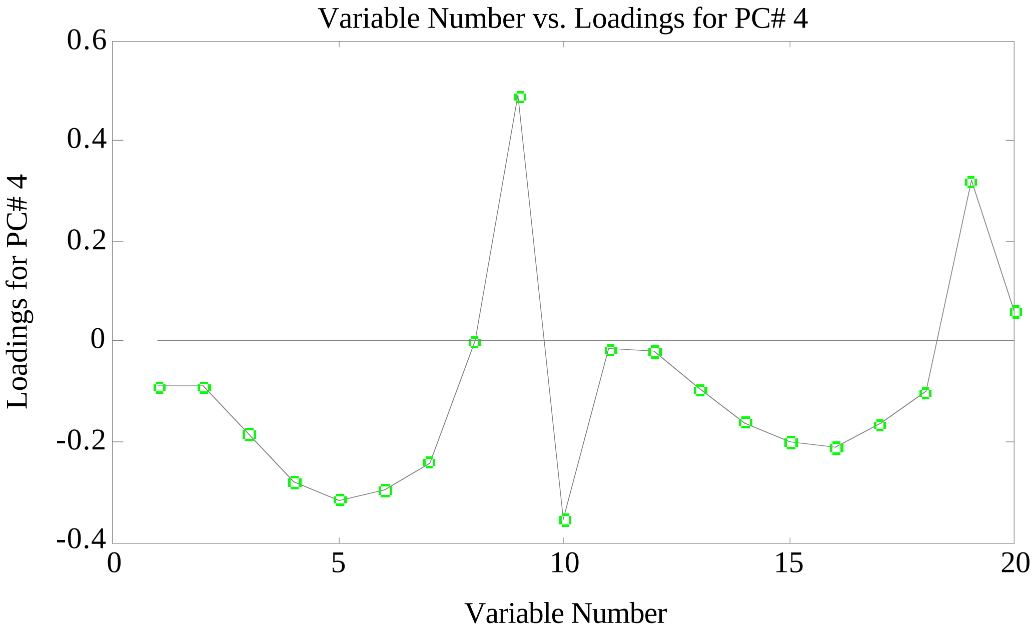

Figure A1.8. Loadings for Fourth PC of Mean Centered SF-11 Data.

The only variables which load strongly into the second PC are the two melter plenum temperatures (variables 10 and 20), which for physical reasons are highly correlated. The plenum temperatures also have a very large variance relative to the bulk glass variables. Thus it is logical that these variables load heavily into one of the first few (rather than latter) PCs.

In the third PC, the temperatures on either side of the melter are anti-correlated. The interpretation of this third PC is somewhat subjective. It would appear that there is a side-to-side temperature variation in the bulk glass that is superimposed over the general trend of the glass temperatures being all highly correlated. Another way to look at this would be to say that, while all melter glass temperatures are correlated, the temperatures are more highly correlated on each side of the melter than between sides. The fraction of the total variation attributable to this mode, however, is small (5.7%) relative to the variation due to level changes (76.6%). It is possible that there is some natural convection process in the melter that causes this variation. This is speculation, however.

The interpretation of the fourth PC is entirely unclear and serves (possibly) as an example of the effect of the orthogonality constraints (which may also be affecting the third PC) and non-linearities on the PCA factors. This type of effect is common when PCA is used to model data that may be inherently of low dimensions but, due to non-linearities, spans a higher dimensional space. An example of this would be a dish shaped collection of data in 3 space. The data is inherently 2 dimensional, but it takes 3 directions to describe it in linear coordinates. It is important to remember that each PCA factor simply describes the majority of the variance that is orthogonal to the previously determined PCs. When physical processes result in trends that are non-linear, confusing factors can result. Non-orthogonal trends can also lead to confusing results.

The SF-11 analysis is now repeated using autoscaling for data pretreatment. The variance captured by the PCA model is given in Table A1.2 below. Comparison of Table A1.2 to A1.1 reveals that autoscaling had the effect of spreading the variance more evenly over the principal components. Where with mean centering the first two PCs captured 85% of the variance, here only 56% of the variance is captured. Furthermore, if each variable has the same (absolute) noise level as we would expect here, autoscaling will tend to amplify the noise in the data set. In the autoscaling procedure this occurs because variables with small variances, and thus a relatively large percentage of variance due to noise, are multiplied by larger factors than other variables in order to achieve a data set where all the variables have equal variance.

The loadings for the first four PCs from the autoscaled SF-11 data are shown in Figures A1.9 to A1.12. Once again, in the first PC the near surface temperatures are loaded most strongly, and this PC very highly correlated with melter level. The general "shape" of the loadings is similar to those shown in Figure A1.3, but the bulk glass temperatures (variables 3 to 6 and 13 to 16) are loaded more strongly. The difference is due to the autoscaling, which increases the relative variance attributable to the bulk glass temperature variables. These variables then become more influential in the PCA model and load more strongly into the earlier PCs.

Table A1.2 Percent Variance Captured by PCA Model of SF-11 Data using Autoscaling.

PC#	Eigenvalue	%Variance	%Total Variance
1.0000	7.8638	39.3188	39.3188
2.0000	3.2934	16.4671	55.7858
3.0000	2.6747	13.3734	69.1592
4.0000	1.9412	9.7058	78.8650
5.0000	1.1103	5.5517	84.4167
6.0000	0.8967	4.4835	88.9002
7.0000	0.6435	3.2176	92.1178
8.0000	0.3718	1.8592	93.9769
9.0000	0.2950	1.4751	95.4521
10.0000	0.2779	1.3897	96.8417

The loadings of the second PC (Figure A1.10) are similar to those of the third PC from the mean centered data (Figure A1.7). Once again, the loadings indicate that the temperature on each side of the melter are anticorrelated. However, as seen in the first PC of the autoscaled data set, the shape of the loadings has changed. The near surface temperatures with large relative variance are de-emphasized and the bulk glass temperature variables load more strongly. It is clear that the side-to-side variation shown in the third PC of the mean centered analysis, whether due to actual variation or noise, is amplified in importance when the scaling to changed to autoscaling.

The third PC from the autoscaled data (Figure A1.11) is very hard to interpret. Like the fourth PC of the mean centered data (Figure A1.8), it appears to be more an artifact of the method than a true factor. The fourth PC (Figure A1.12) is quite similar to the second PC from the previous analysis (Figure A1.6), with the plenum temperatures loading most strongly. Here however, there is more apparent correlation of the plenum temperatures with the near bottom temperatures.

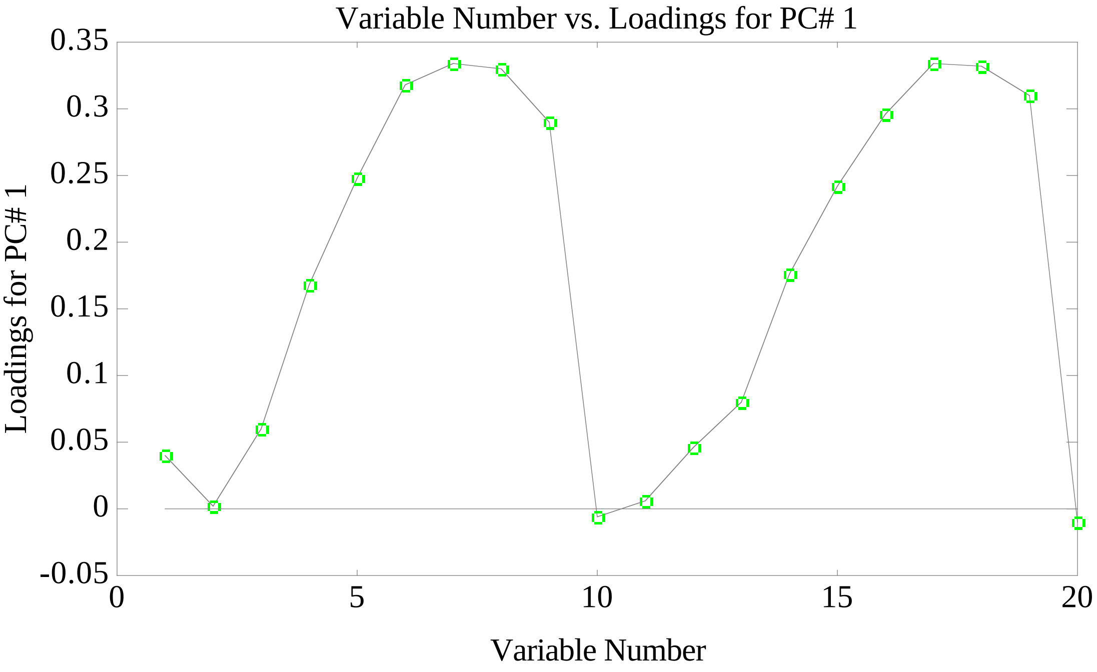

Figure A1.9. Loadings for First PC of Autoscaled SF-11 Data.

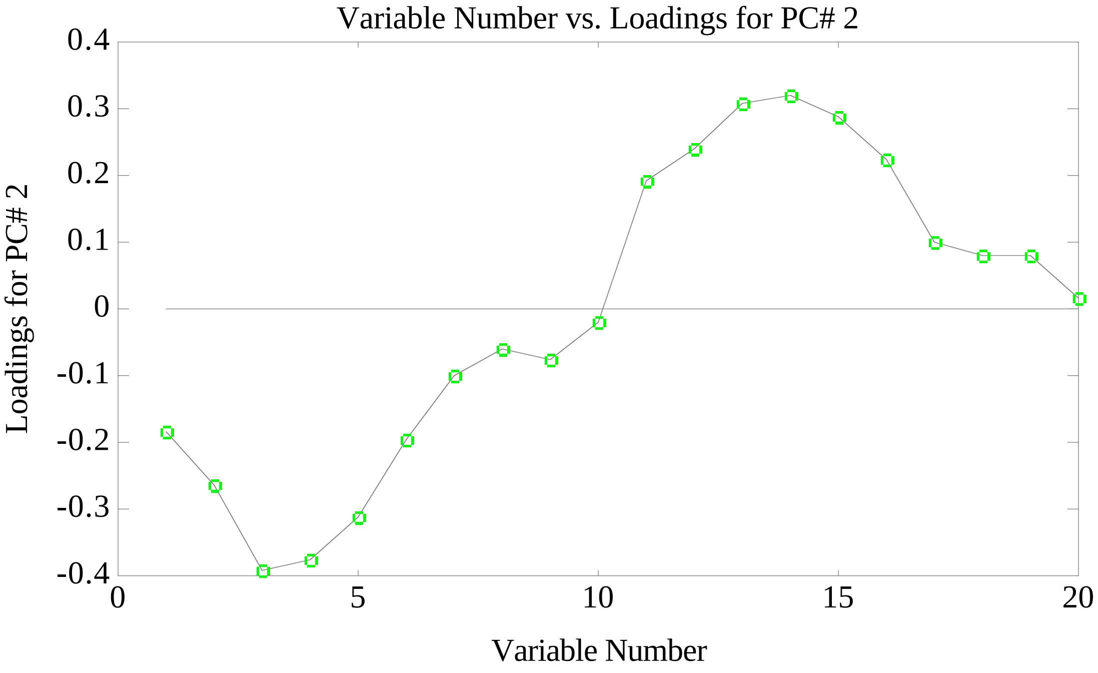

Figure A1.10. Loadings for Second PC of Autoscaled SF-11 Data.

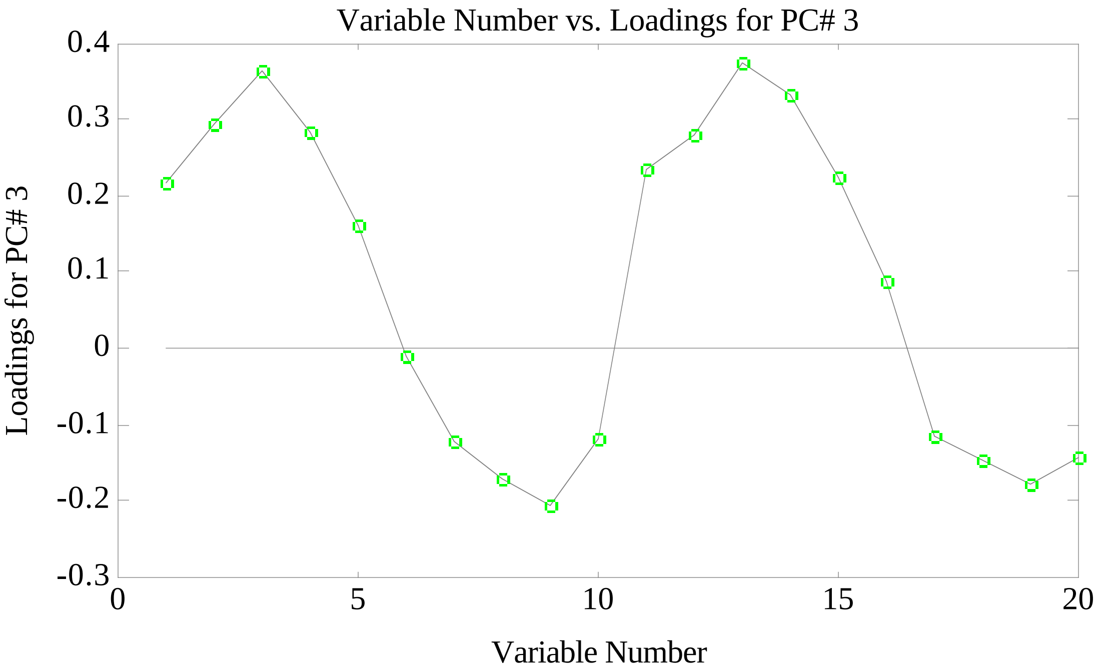

Figure A1.11. Loadings for Third PC of Autoscaled SF-11 Data.

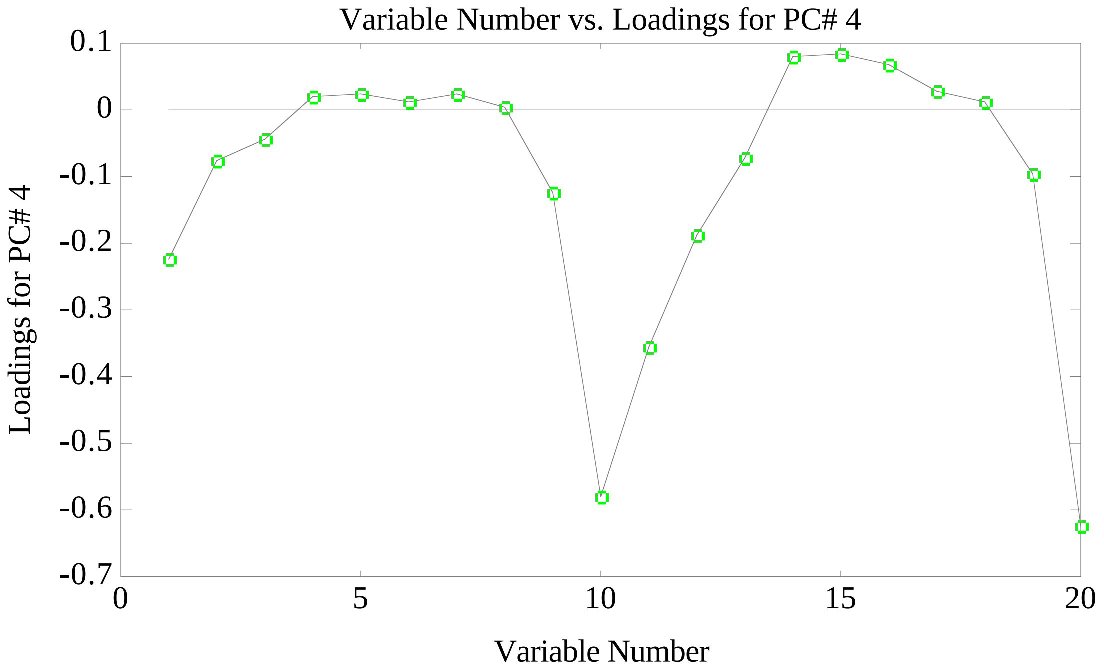

Figure A1.12. Loadings for Fourth PC of Autoscaled SF-11 Data.

This example shows that scaling can greatly affect PCA results. Often, similar factors result regardless of the scaling employed, but there is no guarantee of this. Furthermore, the apparent relative importance of the factors can shift; autoscaling tends to emphasize factors with more variables while mean centering emphasizes factors with more "unscaled" variance. In both the example given and in other similar situations experienced by the author, the mean centering results are more satisfying, and this scaling option is recommended for initial investigations using data of this type. Autoscaling is the preferred option for data sets where the variables are of different units and there is no *a priori* knowledge of the relative importance of the variables. User specified scalings may be employed when there is a fundamental understanding of the system under investigation. In any case, judgement should be used concerning the relative importance of the variables considered and more weight should be given to physically important variables.

It is interesting to note the useful results of the analysis of SF-11, regardless of the scaling option employed. Application of PCA has shown that the major variation (using either scaling) in the temperature data was due to level variations, which are controlled by the operators. Another major variation was the plenum temperatures, which are used as indicators of over or under feeding conditions. There appears to be some side-to-side temperature variation, but it is not possible to determine the source of this effect. Other useful information concerning the run was obtained from the scores plots and the Q statistics, which have not been included here. From this, it was determined the run data did not show any significant trends with time, and that data from all feeding portions of the run was statistically similar.

The data reduction aspect of PCA should also be considered. In the example where mean-centering was employed it was shown that the 20 variable system could be reduced to just three factors that explained nearly 92% of the process variance. This certainly makes the run data much easier to display and digest.

A1.3 Assessment

The examples given demonstrate that PCA can be a useful tool for interpreting multivariate process data. Great caution is urged, however, because it is also possible for the results to be ambiguous or misleading. Furthermore, changes in the data set, such as inclusion or exclusion of data, or changing the ratio of certain "types" of data, can lead to very different results. Scaling can also affect the results. When in doubt about the choice of scaling, several scaling options should be tried and the results of the analyses compared. Caution is urged when the results change drastically with a change in the scaling. Finally, PCA is useful for reducing data sets with large numbers of variables down to a smaller number of factors.

While one cannot expect the PCA results to always be useful, it happens quite often that projections of the data can be found that are good indicators of specific process trends, such as the foaming incidents in the previous examples. Monitoring these plots in real time could provide a useful indicator of process upsets, or perhaps impending process upsets. In the foaming example, this upset could be avoided by lowering feed rate to reduce cold cap coverage which would promote the transfer of dissolved gases to the melt surface. This is essentially the same type of approach as that adopted by MacGregor et. al., as presented at the 1989 annual AIChE meeting (MacGregor 1989) and at the Tronso chemometrics meeting (MacGregor 1990). In the examples presented, a polymerization reactor and a distillation column were monitored using a projection of the process data onto the first two PCs. The Q statistics were also monitored.

When sample classification is the objective, other cluster analysis techniques may be more appropriate. PCA does not necessarily determine the optimal factors for separating different classes of data. When interpretation of the factors is important, techniques such as VARIMAX rotation should be considered. The objective of VARIMAX is to rotate the PCA factors is such a way that variable loadings on each factor tend to be either large or small but not in between (the rich get richer and the poor get poorer) resulting in more interpretable factors. (See for example Veltkamp 1990).

In summary, PCA can be a useful tool in the analysis of multivariate data from dynamic systems provided that proper caution is used in the use and interpretation of the results.

Appendix 2: Equivalence of Replacement and Rebuilding

In this appendix the equivalence between rebuilding the model and using the corrected samples on the old model is demonstrated. Let us call the loadings vectors from our original model **P**. For convenience, assume that the variable to be replaced is the first one. Let us then partition our loadings vectors such that

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_b \\ \mathbf{P}_g \end{bmatrix} \qquad \text{(A2.1)}$$

where $\mathbf{P}_b$ corresponds to the row of loadings of the "bad" sensors and $\mathbf{P}_g$ corresponds to the loadings of the "good" sensors. If we partition our sample **x** as before

$$\mathbf{x} = [\mathbf{x}_b \mid \mathbf{x}_g] \qquad \text{(A2.2)}$$

then the residuals on the original model $\mathbf{r}_o$ can now be expressed as:

$$\mathbf{r}_o = [\mathbf{x}_b(\mathbf{I} - \mathbf{P}_b\mathbf{P}_b^T) + \mathbf{x}_g\mathbf{P}_g\mathbf{P}_b^T \mid \mathbf{x}_b\mathbf{P}_b\mathbf{P}_g^T + \mathbf{x}_g(\mathbf{I} - \mathbf{P}_g\mathbf{P}_g^T)] \qquad \text{(A2.3)}$$

We can now substitute the estimate of the bad sensor outputs from the equations given in the text

$$\mathbf{x}_b = -\mathbf{x}_g\mathbf{R}_{21}\mathbf{R}_{11}^{-1} \qquad \text{(A2.4)}$$

Note however, that $\mathbf{R}_{21}$ and $\mathbf{R}_{11}^{-1}$ can be expressed in terms of $\mathbf{P}_b$ and $\mathbf{P}_g$:

$$\mathbf{R}_{11}^{-1} = (\mathbf{I} - \mathbf{P}_b\mathbf{P}_b^T)^{-1} \qquad \text{(A2.5)}$$

$$\mathbf{R}_{21} = \mathbf{P}_g\mathbf{P}_b^T \qquad \text{(A2.6)}$$

By substituting equations (A2.5) and (A2.6) into equation (A2.4), then substituting that result into (A2.3), we obtain an expression for the residuals of the "corrected" sample $\mathbf{r}_c$:

$$\mathbf{r}_c = [\mathbf{0} \mid \mathbf{x}_g[(\mathbf{I} - \mathbf{P}_g\mathbf{P}_g^T) - \mathbf{P}_g\mathbf{P}_b^T(\mathbf{I} - \mathbf{P}_b\mathbf{P}_b^T)^{-1}\mathbf{P}_b\mathbf{P}_g^T]] \qquad \text{(A2.7)}$$

Here we see that the residuals on the replaced variables is zero, as expected.

We will now calculate the residuals on the new model. First, however, we must form the new model by projecting the old model down out of the "bad" sensor dimension. The basis for this "bad" sensor dimension is

$$\mathbf{x}_v = [1\ 0\ 0\ \ldots\ 0]^T \qquad \text{(A2.8)}$$

Thus, a basis $\mathbf{P}_n$ for the new space can be formed by projecting the bad variable out of the original basis

$$\mathbf{P}_n = (\mathbf{I} - \mathbf{x}_v\mathbf{x}_v^T)\mathbf{P} \qquad \text{(A2.9)}$$

The resulting basis will have the form

$$\mathbf{P}_n = \begin{bmatrix} \mathbf{0} \\ \mathbf{P}_g \end{bmatrix} \qquad \text{(A2.10)}$$

where the top row is a vector of zeros. This basis will not be orthonormal, however, it is easy to get the residuals on this new model $\mathbf{r}_n$ from

$$\mathbf{r}_n = [\mathbf{0} \mid \mathbf{x}_g]\,[\mathbf{I} - \mathbf{P}_n(\mathbf{P}_n^T\mathbf{P}_n)^{-1}\mathbf{P}_n^T] \qquad \text{(A2.11)}$$

where we have now used the appropriate equations for projection onto an non-orthonormal basis. Substituting equation (A2.10) into (A2.11) we obtain an expression for the residuals on the new model:

$$\mathbf{r}_n = [\mathbf{0} \mid \mathbf{x}_g[\mathbf{I} - \mathbf{P}_g(\mathbf{P}_g^T\mathbf{P}_g)^{-1}\mathbf{P}_g^T]] \qquad \text{(A2.12)}$$

Now, in order to show that the residuals on the new model $\mathbf{r}_n$ are equal to the corrected sample residuals on the old model $\mathbf{r}_c$, we need only show that

$$[\mathbf{P}_g(\mathbf{P}_g^T\mathbf{P}_g)^{-1}\mathbf{P}_g^T] = [\mathbf{P}_g\mathbf{P}_g^T + \mathbf{P}_g\mathbf{P}_b^T(\mathbf{I} - \mathbf{P}_b\mathbf{P}_b^T)^{-1}\mathbf{P}_b\mathbf{P}_g^T] \qquad \text{(A2.13)}$$

By manipulation of both sides of this equation we will show that the two sides are indeed equal. Our first step is to notice that all terms of this equation are surrounded by the factors $\mathbf{P}_g$ and $\mathbf{P}_g^T$. We can factor these out to obtain:

$$\mathbf{P}_g[(\mathbf{P}_g^T\mathbf{P}_g)^{-1}]\mathbf{P}_g^T = \mathbf{P}_g[\mathbf{I} + \mathbf{P}_b^T(\mathbf{I} - \mathbf{P}_b\mathbf{P}_b^T)^{-1}\mathbf{P}_b]\mathbf{P}_g^T \qquad \text{(A2.14)}$$

Clearly if this holds, then

$$(\mathbf{P}_g^T\mathbf{P}_g)^{-1} = \mathbf{I} + \mathbf{P}_b^T(\mathbf{I} - \mathbf{P}_b\mathbf{P}_b^T)^{-1}\mathbf{P}_b \qquad \text{(A2.15)}$$

which can be rearranged to yield

$$(\mathbf{P}_g^T\mathbf{P}_g)^{-1} - \mathbf{P}_b^T(\mathbf{I} - \mathbf{P}_b\mathbf{P}_b^T)^{-1}\mathbf{P}_b = \mathbf{I} \qquad \text{(A2.16)}$$

Because our original set of vectors **P** were orthonormal, then their inner product should equal **I**. This leads to

$$\mathbf{P}_g^T\mathbf{P}_g = \mathbf{I} - \mathbf{P}_b^T\mathbf{P}_b \qquad \text{(A2.17)}$$

which may be substituted into equation (A2.16) to obtain

$$(\mathbf{I} - \mathbf{P}_b^T\mathbf{P}_b)^{-1} - \mathbf{P}_b^T(\mathbf{I} - \mathbf{P}_b\mathbf{P}_b^T)^{-1}\mathbf{P}_b = \mathbf{I} \qquad \text{(A2.18)}$$

We now multiply both sides by $(\mathbf{I} - \mathbf{P}_b^{T}\mathbf{P}_b)$ and obtain after some cancellation

$$(\mathbf{I} - \mathbf{P}_b^{T}\mathbf{P}_b)\mathbf{P}_b^{T}(\mathbf{I} - \mathbf{P}_b\mathbf{P}_b^{T})^{-1}\mathbf{P}_b = \mathbf{P}_b^{T}\mathbf{P}_b \qquad \text{(A2.19)}$$

Multiplying through and rearranging yields:

$$\mathbf{P}_b^{T}(\mathbf{I} - \mathbf{P}_b\mathbf{P}_b^{T})^{-1}\mathbf{P}_b - \mathbf{P}_b^{T}\mathbf{P}_b\mathbf{P}_b^{T}(\mathbf{I} - \mathbf{P}_b\mathbf{P}_b^{T})^{-1}\mathbf{P}_b - \mathbf{P}_b^{T}\mathbf{P}_b = \mathbf{0} \qquad \text{(A2.20)}$$

Factoring out the $\mathbf{P}_b^{T}$ and $\mathbf{P}_b$ around each of the terms leads to:

$$\mathbf{P}_b^{T}[(\mathbf{I} - \mathbf{P}_b\mathbf{P}_b^{T})^{-1} - \mathbf{P}_b\mathbf{P}_b^{T}(\mathbf{I} - \mathbf{P}_b\mathbf{P}_b^{T})^{-1} - \mathbf{I}]\mathbf{P}_b = \mathbf{0} \qquad \text{(A2.21)}$$

which implies that

$$(\mathbf{I} - \mathbf{P}_b\mathbf{P}_b^{T})^{-1} - \mathbf{P}_b\mathbf{P}_b^{T}(\mathbf{I} - \mathbf{P}_b\mathbf{P}_b^{T})^{-1} = \mathbf{I} \qquad \text{(A2.22)}$$

Finally, by multiplying each term on the right by $(\mathbf{I} - \mathbf{P}_b\mathbf{P}_b^{T})$ we obtain

$$(\mathbf{I} - \mathbf{P}_b\mathbf{P}_b^{T}) = (\mathbf{I} - \mathbf{P}_b\mathbf{P}_b^{T}) \qquad \text{(A2.23)}$$

which is what we wanted to show.

Appendix 3: Comparison of t- and T^2- Statistics

The sensitivity of the t- and T^2-test for detecting bias errors was investigated by performing several simulations. The model given in section 3.2.1 was used as the test system. A calibration sequence of 1000 samples was generated and a PCA model of the process identified (no PLS models were considered in this test). The t- and T^2-statistics were then developed from the calibration data based on a 20 sample window. The T^2 limits were established based on a 99% confidence interval. In order to put the t-test on equal footing and improve the comparison, the t-test was based on a 99.9% confidence interval. This was done in order to assure that the total number of false alarms was the same for both methods. Because the t-test is done on each variable, and there are 10 variables in the system tested, a 99% limit would result in a total false alarm rate of about 10%. By basing the limits on 99.9%, the total number of false alarms should be equal for both the t- and T^2-tests.

Simulations were conducted to determine the number of false alarms as a check of the calculated limits. In a 1000-trial test, the T^2-test signaled a significant change 6 times. It was found that at least one of the t-test limits was also violated 6 times in this test. Thus, it was verified that the total number of false alarms was about equal for both methods.

A simulation was then performed where biases of magnitude 0.5, 1.0 and 1.5 (in scaled variable units) were added to each output variable in turn. 1000 sample segments of 20 samples each were generated and each of the ten variables was tested, giving 10,000 total tests. The number of detections by each method is shown in Figure A3.1. These results are also presented in Table A3.1. Two separate columns are given for the t-test in the figure. The rearmost column is the total number of times that at least one of the t limits was violated, indicating an added bias. The middle column is the number of times that the t-test not only detected a bias but correctly identified the biased variable. The foremost column is the number of detections from the T^2 test. The number of total and correct responses of the t-test was greater than the number of T^2 responses at all bias levels tested.

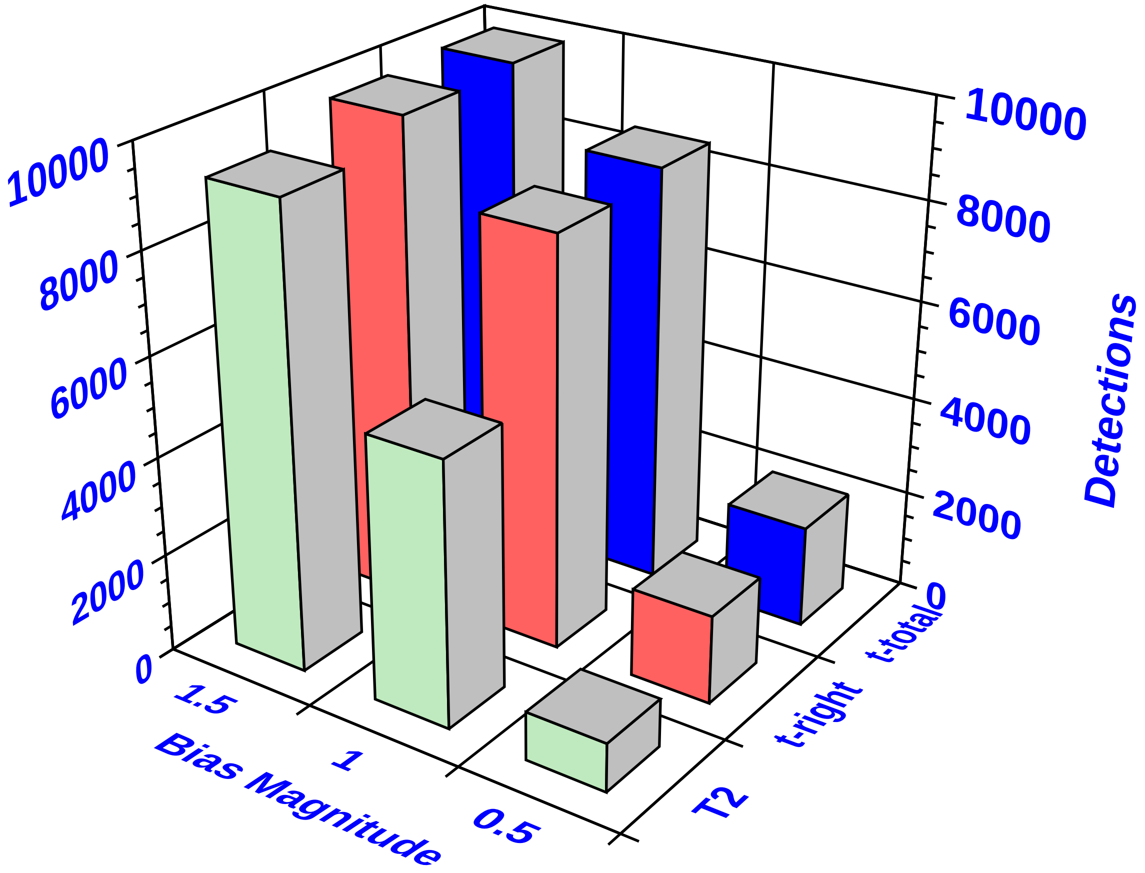

Figure A3.1. Responses of t- and T^2-Tests to Biases on Specific Sensors.

Table A3.1. Response of t- and T^2-Tests to Sensor or Random Biases.

	t-test			T^2-test	
Sensor Bias Magnitude	right	wrong	none	right	none
0.5	1802	264	7937	979	9021
1.0	8298	201	1501	5331	4669
1.5	9779	107	114	9259	741
Random Bias Magnitude	right		none	right	none
0.5	115		885	116	884
1.0	562		438	517	483

An additional simulation was done to check the sensitivity of the methods to random bias vectors. In these tests a randomly generated vector of magnitude 0.5 or 1.0 was generated and added to the simulated process output. The t- and T^2-tests were then used to detect the bias. This was repeated 1000 times. The results are given in Table A3.1. Here there is no "correct" response for the t-test since the bias is not applied to a specific variable. Therefore, if any of the residuals go out of their calculated limits it is listed as a correct response. Under these conditions, the response of the two methods is much more similar. When compared to the specific sensor bias tests, it can be seen that the t-test performance has degraded (in terms of percentage responses to a bias of the same magnitude) while the T^2-test performance is approximately the same. This would be expected because the T^2-test does not look for biases in any particular direction. The t-test, on the other hand, was designed based on individual residuals, which are most sensitive to individual sensor biases.

In practice, the T^2-test, (as implemented here) would also suffer from restrictions on the size of the window required for the test. Because the covariance of the independent residuals **S,** defined in equation (2.26), must be inverted in equation (2.23), there must be at least as many samples as independent residuals. In our example, this means that the window width must be at least 5 samples for the T^2-test, while for the t-test the minimum is technically 1, though from a practical standpoint several samples would always be used in order to increase sensitivity.

As the size of the system increases, and particularly, as the ratio of the number of variables to the number of principal components retained in the model increases, the disparity between the performance of the T^2- and t-tests would be expected to increase. For example, given a system with 100 variables that is well described by a 2 PC model (such as the output of a two component system being analyzed by a 100 channel spectrometer), the T^2-test would require a window of at least 98 samples and would have to take into account the normal deviations over all the variables. The t-test, however, would require a window of only a few samples. Furthermore, under these circumstances, the residuals would be a very good estimate of the measurement noise on each of the particular variables. Any changes in the underlying sensor behavior, then, would have a very specific affect on the particular sensor and the method would become very sensitive to individual sensor faults.

Biographical Note

Barry Mitchell Wise was born September 5, 1958 in Chelan, Washington. Barry grew up in nearby Manson, where his parents, Nyle C. and Frances E. Wise raised apples. He attended Manson High School and graduated in 1976. Barry received Bachelor of Science degrees in both Chemistry and Chemical Engineering from the University of Washington in 1982. From fall of 1982 to fall of 1985 Barry worked at Battelle Pacific Northwest Laboratories where he performed research that was primarily related to nuclear waste solidification. Barry returned to graduate school at the University of Washington in fall of 1985 and received his Master of Science in Chemical Engineering under Harold Hager in fall of 1987. Barry then began working with N. Lawrence Ricker towards his Doctorate. On August 11, 1988 Barry met Jill Raisler, who became his wife on June 16, 1989.