University of Alberta

Process Characterization and Control using Multivariate Statistical Techniques

by

Lakshminarayanan,S. ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Doctor of Philosophy**.

in

Process Control

Department of Chemical and Materials Engineering

Edmonton, Alberta

Spring 1997

Dedicated to

My Parents, Murali & Bhoopathi

Subha and Shreya in appreciation of their sublime love

Lakshmi, Nandu and Sirish with gratitude for their support

Abstract

Fast paced developments in electronic hardware technology have resulted in heavily instrumented chemical plants. Process data from these units are frequently logged on to computers leading to data overload. To cope with these trends, data mining tools that extract useful information from the database have been proposed. These include methods based on simple visualization, multivariate statistical techniques (such as principal components analysis (PCA), partial least squares (PLS) and canonical correlations analysis (CCA)), artificial intelligence (induction or rule based) and neural networks. Recent studies indicate that a new data mining prototype is introduced every three months.

In this thesis, the use of multivariate techniques in the characterization and control of chemical processes (continuous and batch/semibatch) is explored. Utilizing the dimension reduction properties, these tools have long been used for applications related to process monitoring and fault detection in a statistical process control (SPC) framework. In certain situations (e.g. inferential model building), these methods have provided a robust alternative to the ordinary least squares regression procedure. Besides describing the theory and applications of these techniques in such *traditional* areas, we have investigated their suitability in the modelling and control of dynamic multivariable systems.

A powerful empirical (black-box) identification strategy that provides multivariable state space models (Canonical Variate Analysis, CVA) is reviewed. Extensive simulations are used to establish the superiority of CVA over another popular state space identification

algorithm (N4SID). Extension of the CVA method to model a class of nonlinear systems. the Hammerstein structure, is provided.

Identification and control of univariate (single input single output - SISO) processes represents a relatively mature field; it is easily understood and readily implemented. We propose a novel multivariate modelling and controller synthesis strategy that is based on a combination of the PLS technique and the identification/control theory developed for SISO systems. Recognizing that industrial plants usually operate in the regulatory mode. expressions for the design of multivariable feedforward controllers are developed. To cope with constraints on the process variables, the PLS model has been integrated into the Model Predictive Control framework. The domain of applicability extends to nonlinear systems - the Hammerstein and Wiener models provide motivating examples.

Case studies involving simulations, laboratory experiments and industrial data are included wherever appropriate.

Preface

The work presented in this thesis is aimed to provide an insight into the theory and application of multivariate statistical techniques such as Principal Components Analysis (PCA). Partial Least Squares (PLS) and Canonical Correlations Analysis (CCA). The applications are chiefly in the area of process modelling (identification), control and monitoring.

The statistical techniques are presented at a tutorial level in chapter 1 along with real world applications involving inferential model building and process monitoring. The basics of black-box modelling are reviewed in chapter 2 with the focus being on state space structures. A powerful state space identification technique - Canonical Variate Analysis (Larimore, 1990) - based on CCA is described and compared with related identification techniques. Further, the CVA approach is extended to model a class of nonlinear systems - the Hammerstein model. Illustrative examples are provided using the simulation of a nonlinear CSTR, an acid-base neutralization tank and real data obtained from two experimental heat exchange systems.

A novel modelling strategy that results in a simple controller design is discussed in chapter 3. Utilizing the latent variables generated by the PLS algorithm. this method captures the dynamic information from process data in a diagonal structure and is capable of handling nonlinear and nonsquare systems. Expressions for the synthesis of multivariable feedforward controllers under the above framework are also provided. Some of the material presented here can be considered as extensions to a related method proposed by Kaspar and Ray (1992. 1993) for linear systems. The theory developed for this modelling and control (feedback plus feedforward) approach is supported by simulation studies using the Wood-Berry distillation column and the acid-base neutralization tank. In chapter 4, issues related to the integration of the dynamic PLS models (generated using the ideas presented in chapter 3) with advanced model based predictive control algorithms (such as the Dynamic Matrix Control - DMC) are dealt with. The fusion of the PLS based modelling strategy with the model based predictive control algorithms results in a synergistic effect - an elegant multivariate modelling tool built on the foundations of univariate identification methods and a control mechanism capable of meeting the process constraints. Modelling and constrained control of a simulated pH neutralization system (using a nonlinear Wiener-PLS model) and a laboratory stirred tank heater illustrate DMC applications operating in the PLS latent space. In contrast to the earlier chapters which focussed primarily on continuous

systems. the monitoring and fault detection of batch and semi-batch processes is considered in Chapter 5. Operation of such processes are characterized by the presence of multiple rates of measurements. To handle this multirate scenario, the PLS based monitoring and fault detection technique developed by MacGregor and coworkers (1994a, 1994b. 1995) has been extended. In line with the material presented in chapter 1, a PLS model is obtained using a database of normal batches. Online monitoring of new batches is performed by comparing the current process data with the template provided by the PLS model. It may so happen that the database characterizing normal plant operations contain batches of varying run lengths. A method is suggested to make use of all these batch runs in the construction of the nominal data-based model. The monitoring algorithm is evaluated on : (1) a fed-batch antibiotic producing fermentation and (2) a semi-batch polymerization reactor and is found to give quick and early detection of faults along with good predictions of the final product quality. Chapter 6 provides conclusions based on the material presented in chapters 1 through 5 as well as recommendations for future work.

The formulation of PCA. PLS and CCA in an optimization framework that results in their expression as eigenvalue-eigenvector problems is detailed in appendix A. In the process of developing the equations for the PLS based multivariable feedforward controllers. the Cramer's rule for the solution of a system of linear equations have been extended to include nonsquare systems (an elaborate description of the extended Cramer's rule is provided in appendix B).

All data analysis, simulation and experimental studies were performed using MATLAB/Simulink running on IBM compatible personal computers and Unix workstations.

Acknowledgements

I wish to express my heartfelt gratitude to my supervisors - Professors Sirish Shah and Nandakumar for their excellent and enthusiastic guidance during the entire course of my thesis research. Both of them have been a tremendous source of support during my stay here. Mere words cannot express the measure of gratitude that I owe them.

Prof. Grant Fisher's comments on my work during the control group meetings have helped me to keep track of the overall perspective. I am indebted to him for the constructive suggestions he has provided as well as for making available state-of-the-art software and hardware. I was fortunate to learn most of my theoretical control concepts from a fine teacher - Prof. Ray Rink. These will, hopefully, provide a strong foundation for my future work. I also like to thank Prof. Reg Wood and Dr. Fraser Forbes for their sincere and timely feedback on my research. Prof. Dave Jobson (Faculty of Business) and Dr. Jim Kresta (Syncrude, Canada) deserve special mention for teaching me the basics of Multivariate Statistics. The Faculty of Business entrusted me the responsibility of teaching an undergraduate course - it was a wonderful experience to interact with another section of the university community. Prof. Prem Talwar provided me useful suggestions and teaching tips which made things easier for me. I would like to thank NSERC, University of Alberta (for the PhD Scholarship and the Dissertation Fellowship), Pan Canadian, DuPont, PAPRICAN and the Department of Chemical Engineering (U of A) for their financial assistance during the course of this PhD research.

It has been my privilege to work as a part of a wonderful computer process control group at the U of A. The camaraderie and the cooperative spirit within the group has allowed me to debate and discuss my ideas freely. The broad range of talent within this group has also allowed cross-fertilization and nurturing of many different ideas that have invariably influenced the direction of this thesis. A big thanks to my student colleagues : Sreekanth Lalgudi, Dr. Pranob Banerjee, Dr. Ravindra Gudi, Randy Miller, Dr. Biao Huang, Dr. Munawar Saudagar, Dr. Lanre Badmus, Dr. Danyang Liu, Dr. Kent Qi, Hiroyuki Fujii, Dr. Mary Bourke, Dr. Steve Niu, Dr. Ezra Kwok, Chee Wong, Amy Yiu, Albert Chiu, Ricky Leung, Rohit, Anand, Sirajul Khan, Ms. Kamrunahar, Arun, Dayadeep (Misha), Daniel, Lisa, Aseema, Li and others for providing an intellectually stimulating environment. Special mention must be made of Dr. Ravindra Gudi (*flute, veena and the flutter will never be forgotten*) and Rohit for being such wonderful office partners - interactions with them

have been a truly rewarding experience. Outside the group, Dr. Philip Mees. Kevin Dorma. Andy Jenkins and Gagandeep have offered their expertise and help at various times. Thank you, guys.

I like to thank the staff at the Chemical Engineering Office : Cindy, Bev, Diane and Shantel for their help with administrative matters. Bob Barton deserves a big THANK-YOU for ensuring that the computers were alive and well.

Srini and Manisha have provided solid companionship over the last four years along with Murugappan, the (Ravindra-Vasudha-Dhanvini) Gudi family, Anand (Gope) and Balaji. We have shared some wonderful moments at Edmonton, in the mountains (Jasper and Banff) and during our great American trip. My friends at the Kannada Association (notably, Mr. Krishna and Mrs. Latha Bhat, Dr. Somayaji and family, Prof. Prasad and family), my host family (Mrs.Usha Thakker and Mr. Mahendra Thakker). Narayani and Gopal. have been very helpful at some crucial times - to them goes heartfelt thanks from me and Subha.

Finally, I like to thank my wonderful in-laws - Smt. Sakunthala Hariharan and Sri. Hariharan for being very understanding and accommodative.

Contents

List of Tables

List of Figures

Chapter 1

Introduction to Multivariate Statistical Methods

1.1 Overview

In this chapter, an introduction to three multivariate statistical techniques that will be employed for modelling, control and monitoring of multivariable processes is presented. Principal Components Analysis (PCA), Partial Least Squares (PLS) and Canonical Correlations Analysis (CCA) are finding increased use in chemical engineering and process applications. The mathematical and the algorithmic details of these techniques are presented. Two industrial data sets are analyzed. Data from a distillation column in Mitsubishi Chemicals. Japan is used to develop a PLS based inferential model for distillate composition control[2]. The second data set is from the Shell styrene unit located in Scotford, Alberta (Canada)[3]. This data is used to illustrate the utility of PCA in detecting process shifts and fault isolation.

[1]Sections of this chapter have been submitted for possible presentation as : H. Fujii, S. Lakshminarayanan and Sirish L. Shah, "Application of PLS to the Estimation of Distillation Tower Top Composition", Submitted to the IFAC ADCHEM '97 Meeting, August 1996.

[2]This work was done in collaboration with Mr. Hiroyuki Fujii of Mitsubishi Chemicals. Japan. His contribution in providing the data as well as in its analysis is gratefully acknowledged

[3]Sincere thanks are due to Dr. David Onderwater (Shell Canada) for providing this data

1.2 Contributions of this chapter

- A tutorial overview of three multivariate statistical methods that will be used extensively in this thesis is presented. It is hoped that this overview will help in understanding the material presented in later chapters. It must be pointed out that such introductory material is also available from other sources (e.g. Wise (1991), Kresta (1992), Kaspar (1992) and Phatak (1993)).

- The PLS modelling approach was used to develop an inferential model for a distillation column (Mitsubishi Chemicals, Japan). This model was implemented on the plant and significant improvement in control was obtained.

- The PCA method is used to analyze process data from a styrene plant (Scotford complex of Shell). Interesting observations can be made from the analysis of this data.

1.3 Introduction

Advances made in the areas of instrumentation and data acquisition have made it possible to collect large amounts of data in the process industry. Use of univariate statistical process control (SPC) charting procedures are very common in the *parts* industry and to a lesser extent in the process industry. Univariate SPC charts (EWMA/CUSUM etc.) are used to monitor key process variables in order to detect the occurrence of abnormal episodes. By detecting the source of this abnormality, improvements in the operation of the process (in terms of safety, waste reduction etc.) and consequently product quality can be realized. When such a univariate approach is used to analyze multivariate data, interaction between the variables is not taken into account. This not only results in misleading process information but also makes the interpretation and diagnosis tasks difficult.

It is in this context that multivariate statistical methods such as PCA, PLS and CCA are finding increased use in the analysis and archival of multivariate data sets. Brought to the centerstage of chemical engineering by MacGregor and coworkers (e.g. Kresta (1992), Nomikos and MacGregor (1994)) and Wise (1991), PCA and PLS have been applied to a variety of problems involving multivariate process monitoring and modelling (Qin and McAvoy (1992a), Qin (1993), Ricker (1988)). In these applications, the data compression facility offered by these methods were utilized in condensing the variance of the process into a very low dimensional latent subspace. This data compression feature provides a low-dimensional window into the process and facilitates the tasks of monitoring and fault detection (Kresta *et al.*, 1991). An interesting dynamic PLS modelling procedure that can be directly utilized for multivariable control system design has been reported (Kaspar and Ray, (1992,1993)). The use of CCA for chemical process modelling and fault detection applications is reported in Schaper *et al.* (1994), Lakshminarayanan *et al.* (1995) and Wang

et al. (1996). Perhaps, the first application of the PCA technique in an industrial setting came from Moteki and Arai (1986) who used it to derive optimal operating conditions to synthesize specific polymer grades.

This chapter is organized as follows. A tutorial introduction is presented for the three multivariate techniques considered here. Each of these methods is also cast as constrained optimization problems whose solution is obtained by solving related eigenvalue-eigenvector problems.

1.4 Background

Consider two blocks of measurements X and Y. The X block is comprised of the process or causal variables such as temperature, pressure and flow rate measurements. The Y block is comprised of quality variables such as product purity, molecular weight etc. If the goal is to perform data compression and extract the process information from only one block of data , then PCA is an appropriate technique. Often times, we seek to predict the Y space using only the X space measurements, with a linear model as given by equation (1.1). Such models may be useful in applications such as inferential control. This linear estimator can also be used to model dynamic systems if lagged values of the inputs and/or the outputs are included in the X block. In any case, most of the parameter estimation problems that occur in engineering practice can be reduced to the form given in equation (1.1) and has therefore received considerable attention since the time of Gauss (early 19th century).

$$Y = XC + Noise \tag{1.1}$$

The ordinary least squares solution (OLS) of the above system of linear equations is given by

$$C_{OLS} = (X^T X)^{-1} X^T Y \tag{1.2}$$

It is seen that no attention is paid to the correlational structure of Y, nor is any dimension reduction attempted in the X space. The OLS procedure focuses exclusively on the model fit (predictions) - no consideration is given to the numerical stability aspects of the linear regression problem. Such an approach creates problems in the presence of correlated process measurements (which is often the case with industrial data), where the X matrix is illconditioned - in the extreme situation the inverse may not exist. Even if the inverse can be computed, the variance of the estimated parameters will be large indicating that the estimator will be unstable. Poor performance of the routinely used OLS procedure in the presence of correlated measurements makes it necessary to opt for other available choices. Several multivariate techniques such as PLS, Principal Components Regression (PCR) etc. have been proposed for this task. These methods circumvent the collinearity problem associated with multivariate data by constructing and relating *latent* or *virtual*

variables (linear combinations of the original variables) instead of the original variables. The philosophy governing the choice of the latent variables for the X space differentiates these methods. Two attributes are of major importance for the estimator : (i) numerical stability and (ii) obtaining good fit of the data. The linear combinations must account for much of the variation of X and must correlate well with the variables in the Y space to achieve the objectives of model stability and goodness of fit. Each of the multivariate methods accomplish a different level of balance between these two goals. Stone and Brooks (1990) and Wise (1991) describe the nature of these tradeoffs in their discussion of continuum regression - a common framework that encompasses several multivariate methods including those considered in this thesis.

1.4.1 Terminology

Let us assume that the X and Y blocks consist of nx and ny variables respectively. The number of observations in each of them is N. We shall also consider that the X and Y block variables are mean centered and suitably scaled. Without scaling, in PCA and PLS it is possible to bias the results towards variables that have a larger magnitude.

$$X_{N\times nx} \rightarrow [x_1|\,x_2|\,\cdots|\,x_{nx}]$$

$$Y_{N\times ny} \rightarrow [y_1|\,y_2|\,\cdots|\,y_{ny}]$$

The 'linear combinations' of the columns of the X and Y spaces are represented by $t_i = X\,j_i$ and $u_i = Y\,l_i$. j_i and l_i are vectors of weights that are used to obtain a particular linear combination. The subscript i denotes the i^{th} linear combination of the corresponding space. The first 'k' linear combinations of the X space will be expressed as

$$T = [t_1|\,t_2|\,\ldots|\,t_k] = X\,[j_1|\,\cdots|\,j_k] = XJ_k \tag{1.3}$$

In the following, the covariance matrices are denoted by Σ with the appropriate subscripts. For example, Σ_{xy} will signify $\frac{X^TY}{N-1}$ - the covariance between the X and Y space.

1.5 Principal Components Analysis

The PCA procedure is concerned with the analysis of one block of data X. The goal is to form new orthogonal variables which are linear composites of the original variables. If the original variables are correlated, it is possible to summarize most of the variability present in the *nx-variable* space in terms of a lower *n-dimensional* subspace ($n \ll nx$). Principal components analysis essentially reduces to identifying a new set of orthogonal axes. If a substantial amount of the variability present in the original data set is accounted for by a few new variables (or principal components), then these principal components (also called *latent* or *virtual* variables) can be used for further interpretational or analysis purposes.

Assuming that we are interested in forming the following nx linear combinations :

$$\begin{aligned} t_1 &= j_{1,1}x_1 + j_{1,2}x_2 + \cdots + j_{1,nx}x_{nx} \\ t_2 &= j_{2,1}x_1 + j_{2,2}x_2 + \cdots + j_{2,nx}x_{nx} \\ &\vdots \\ t_{nx} &= j_{nx,1}x_1 + j_{nx,2}x_2 + \cdots + j_{nx,nx}x_{nx} \end{aligned} \tag{1.4}$$

In compact form, the above equation system can be represented as $T = X\,J_{nx,PCA}$ where $J_{nx,PCA} = [j_1 \mid \cdots \mid j_{nx}]$.

The first principal component, t_1, accounts for the maximum variance in the data, the second principal component, t_2, accounts for the maximum variance that has not been accounted for by the first principal component, and so on. To achieve the above objective, some constraints need to be placed on the weight vectors j_1 to j_{nx}. These are given by

$$j_i^T j_i = 1 \quad (i = 1, \cdots, nx) \tag{1.5}$$

and

$$j_i^T j_k = 0 \quad (i \neq k) \tag{1.6}$$

The first constraint as given by equation (1.5) is somewhat arbitrary. This condition (to fix the scale of the new variables) is necessary because it is possible to increase the variance of a linear combination by just scaling the weights[4]. The condition given by equation (1.6) ensures the orthogonality of the principal components. The mathematical problem of determining the weight vectors j_1 to j_{nx} can be approached in terms of the following optimization problem :

- Objective : Find a linear combination of the X variables that has the maximum variance amongst all possible linear combinations.
- Objective function : $max\left\{j_1^T \Sigma_{xx} j_1\right\}$
- Constraint : $j_1^T j_1 = 1$
- Solution[5] : j_1 is the first left singular vector of $\Sigma_{xx}^{1/2}$
- Remarks :
  1. The linear combination of X variables that has the next highest variance subject to the condition of being orthogonal to the first linear combination is Xj_2. j_2 is the 2^{nd} left singular vector of $\Sigma_{xx}^{1/2}$.

[4]For example, it is possible to increase the variance of the first principal component by just doubling the weights

[5]Derivation of the PCA,PLS and CCA solutions as eigenvalue-eigenvector problems is provided in Appendix A

2. Likewise, we can extract 'nx' orthogonal linear combinations. Thus. $J_{nx.PCA} = [j_1 | \cdots | j_{nx}]$.

The principal component weights matrix, J, is therefore obtained from a singular value decomposition (SVD) of the data matrix, X (or equivalently, $\Sigma_{xx}^{1/2}$). If there are redundancies in the X block (because of correlation among the variables) - some of the singular values will be insignificant implying that the corresponding principal components (say, dimensions n+1 through nx) are insignificant. Noise and redundancies present in the data set are confined to these insignificant PCA dimensions.

From another view point, we can consider PCA as a technique that decomposes a data matrix X into a sum of rank 1 matrices (similar to spectral decomposition) as follows :

$$X = t_1 p_1^T + t_2 p_2^T + \cdots + t_n p_n^T + E_{n+1} = TP^T + E_{n+1} \tag{1.7}$$

While writing the above equation, it is assumed that all of the insignificant information in the data set is confined to the error matrix, E (which lumps the PCA dimensions n+1 through nx). In the above representation, the matrix P (size $nx \times n$) is called the *loadings matrix* (note that $P = J$) - the matrix composed of weights attached to the original variables in creating the principal components. Matrix T represents the values of the new variables (projection of the samples on to the lower n-dimensional subspace) and is called the *principal components scores matrix*.

1.5.1 Identifying the Optimal Dimension of the PCA Model

The number of principal components, n, that are to be extracted is an important factor to consider. The decision depends on how much information can be sacrificed (as unaccounted variance) to enable data compression. This, no doubt, is a subjective decision but some of the common rules adhered to are :

1. In the case of standardized data, retain only those components whose eigenvalues (of Σ_{xx}) are greater than one. This is known as the eigenvalue-greater-than-one rule.

2. From the plot of variance explained by each principal component versus the number of components (*scree plot*), an elbow is located. The position of the elbow determines the number of PCA dimensions to retain.

3. Often, the number of PCA dimensions is determined based on a fixed percentage (usually about 80%) of the cumulative variance explained.

4. Cross validation techniques are often considered as statistically sound procedures for determining n, the number of principal components to retain. These techniques (Wold, 1978) use only a portion of the training (or calibration) data set to obtain the PCA model and then compute the Prediction Error Sum of Squares (PRESS) for the unused

portion of the training set. This procedure is repeated by retaining different data portions for model building - the dimension that gives the lowest cumulative PRESS is chosen as the optimal value of n.

However, no rule provides best results under all circumstances. The purpose of the study, the type of data, the interpretability of the principal components, the amount of variation that needs to be explained, the parsimony principle are the key factors to be considered while deciding on the number of retained principal components.

1.5.2 Tools for Online Process Monitoring

The PCA model constructed using data collected during normal operation of the plant can be used to perform online monitoring of the process (i.e. detect and diagnose faults). The *in-control* PCA model of the process forms a reference against which future plant operations can be compared. In this section, an overview of the analytical tools available for determining out-of-control status (fault detection) and the underlying cause(s) for the abnormal event (fault isolation) will be provided. The fundamental tools for achieving this are the scores and loadings plots.

1. Score Plots

   The scores plot is a depiction of the principal component scores for any two PCA dimensions (e.g. t_1 versus t_2; see Figure (1.10) for an illustration). Usually, it serves to indicate the relationship between the various samples. Two similar samples by virtue of their similar scores, will lie close to each other in the scores plot. It is easy to conclude that all data points that are similar in nature tend to cluster together in the scores plot. The scores plot are thus excellent tools to detect abnormal process behavior.

   The scores for the n principal components can be plotted against each other (plotting these for the first few components is usually adequate) forming two dimensional monitoring charts. The control limit contour depicting the normal operating region is an ellipse (joint confidence region). Any abnormal shift in the process variables (whether the basic correlation between variables remains intact or not) is clearly indicated in the scores plot because the projected scores move out of the normal operating zone. With the help of a loadings plot (e.g. p_1 versus p_2), the fault can then be isolated. If no abnormal shift occurs in the process variables but the correlational structure breaks down, then the score plots will not be able to detect the fault. To avoid this, a third dimension showing the squared prediction error (SPE) is included (see Figure 1.10).

   In order to compute the SPE, it is necessary to define the error which can also be viewed as the model-plant mismatch. If this mismatch gets larger, it is an indication

that the PCA model no longer reflects the current status of the plant. If the process variables deviate from normal values but retain the correlational structure found during acceptable plant operation, then the SPE will not be large. It is apparent that the SPE is a useful measure to detect process upsets.

Let x_{new} denote the new multivariate observation (a vector of dimension $1 \times nx$). This observation can be projected onto the hyperplane defined by the PCA loading vectors to obtain the score value $t_{new} = x_{new} P$. t_{new} is the $1 \times n$ vector of scores from the model and P is the $nx \times n$ matrix of loadings determined from the *normal* plant data. The PCA model prediction for x_{new} is given by $\hat{x}_{new} = t_{new} P^T = x_{new} P P^T$. The $1 \times nx$ dimensional error vector is given by $e_{new} = x_{new} - \hat{x}_{new}$ from which the SPE can be calculated as $e_{new}^T e_{new}$. The SPE can be considered as a scalar measure of the plant-model mismatch - a small value of SPE indicates that the model is still a good representation of the plant and a large SPE value indicates otherwise. Should a large SPE value occur, the process operators should be alerted and the fault diagnostics procedures must be initiated. The confidence limits on SPE can be calculated as follows (Jackson and Mudholkar, 1979) :

$$SPE_\alpha = \Theta_1 \left[1 + \frac{c_\alpha h_0 \sqrt{2\Theta_2}}{\Theta_1} + \frac{\Theta_2 h_0 (h_0 - 1)}{\Theta_1^2} \right]^{\frac{1}{h_0}} \tag{1.8}$$

where

$$\Theta_i = \sum_{j=n+1}^{nx} \lambda_j^i \quad (i = 1, 2 and 3) \tag{1.9}$$

and

$$h_0 = 1 - \frac{2\Theta_1 \Theta_3}{3\Theta_2^2} \tag{1.10}$$

Here, the λ_i's denote the eigenvalues of Σ_{xx} (equivalently, the square of the singular values of X or $\Sigma_{xx}^{1/2}$) and c_α is the normal deviate corresponding to the upper $100(1 - \alpha)$th percentile. Usually, a value of 0.05 is used for α (95th percentile).

Alternately, the n principal components extracted can be plotted individually (since the principal components are uncorrelated this is valid) in a manner similar to the univariate Shewhart chart. In this case, each of the charts represent the monitoring of not a single process variable but rather a group of original variables. By examining the n individual score plots and with a knowledge of the loadings matrix (or a loadings plot), it is possible to pinpoint the cause(s) for the abnormality. In this case, SPE must be monitored individually using a univariate procedure.

2. Loadings Plots

   To help in isolating the reasons for abnormalities in process operation. it is necessary to interrogate the underlying PCA model. Some of these methods are discussed in MacGregor *et al.* (1994a). One common fault isolation technique is the use of the loadings plot which shows the relationship between the process variables in exactly the same way as the scores plot exhibits the relationship between the observations. All variables sharing the same information content (i.e., correlated variables) tend to cluster together. Such clusters of variables usually dominate different PCA dimensions.

   If abnormal scores are noticed for any particular PCA dimension, the variable cluster(s) that dominate the dimension may be responsible for the unusual event. Loadings plots help in visualizing these variable clusters.

3. Contribution Plots

   The SPE values computed above can also be utilized in an effective manner for fault isolation. The fractional contribution of each process variable to the overall SPE can be computed as :

$$\gamma_i = \frac{SPE_i}{SPE} \quad (i = 1, 2, \cdots nx) \tag{1.11}$$

   where SPE_i denotes the square of the i^{th} element of the error vector e_{new}. If the fractional contribution of any variable is significant (say greater than 10%). then it is very likely the cause for the abnormality. This is the basis of the contributions plot concept proposed by Miller *et al.* (1994).

   Though an unambiguous answer regarding the source of the fault is not provided by either the loadings or the contribution plots, they definitely provide a focal point for detecting the possible cause(s). An *expert system* can be fired up at this stage to *zero in* on the exact fault.

1.5.3 Principal Components Regression

Once a PCA description of X is obtained, the latent variables can be used to determine C in equation (1.1). This is the basis of Principal Components Regression (PCR). In PCR, no consideration is given to the relationship between the Y-block variables, but the orthogonal latent space (spanned by the n principal components) formed for the X-block are employed in the regression. This is in contrast to the OLS procedure. The PCR solution for equation (1.1) is given by

$$C_{PCR} = (\hat{X}^T \hat{X})^{-1} \hat{X}^T Y \tag{1.12}$$

where $\hat{X} = TP^T$ is an *approximate but stable* representation of X. It is seen that PCR places more emphasis on the description of the X-block (stability) while paying little or no attention either to the model fit or the correlational structure of the Y block.

1.6 Canonical Correlations Analysis

CCA is a popular technique for identifying relationships between two sets of variables. It is often possible to designate one block of data as the predictor block and the other as the criterion block. For example, the process measurements form the predictor (independent) block and the quality variables make up the criterion (dependent) block. Then the objective is to determine if the predictor set of variables affects the criterion set of variables. If the objective is to ascertain the relationship between two sets of variables, then it is not even necessary to designate the two sets of variables as the dependent and independent sets. The predictor and the criterion blocks will be denoted as X and Y respectively.

In CCA, the goal is to relate linear combinations of the X and Y spaces - the linear combinations of the X and Y space (also called canonical variates) are generated in pairs *such that* the correlation between them is maximum (the correlation is referred to as the canonical correlation). Once the first pair of linear combinations is extracted, the second pair is selected such that the two pairs of canonical variates are uncorrelated. To pose the optimization problem correctly, it is necessary to constrain the variance of the canonical variates to unity. The mathematical details of the CCA algorithm are presented next.

1.6.1 CCA : The Optimization Approach

- Objective : Find a linear combination pair (from among all possible pairs) of X and Y spaces that have maximum correlation between them.
- Objective function : $\max \left\{ \frac{j_1^T \Sigma_{xy} l_1}{\sqrt{j_1^T \Sigma_{xx} j_1} \sqrt{l_1^T \Sigma_{yy} l_1}} \right\}$
- Constraints :

  $$j_1^T \Sigma_{xx} j_1 = 1$$
  $$l_1^T \Sigma_{yy} l_1 = 1$$

- Solution :

  $j_1 = \Sigma_{xx}^{-1/2}$ * First left singular vector of $\Sigma_{xx}^{-1/2} \Sigma_{xy} \Sigma_{yy}^{-1/2}$

  $l_1 = \Sigma_{yy}^{-1/2}$ * First left singular vector of $\Sigma_{yy}^{-1/2} \Sigma_{yx} \Sigma_{xx}^{-1/2}$

- Remarks :

  1. The next best linear combination pair (Xj_2,Yl_2), orthogonal to the first pair. is obtained by choosing j_2 and l_2 using the second left singular vectors in the expression above. Subsequent pairs are chosen orthogonal to all previous pairs.

2. We can extract a maximum of min(nx,ny) such pairs. In matrix notation we may write $J_{CCA} = \left[j_1 \mid \cdots \mid j_{\min(nx,ny)}\right]$.
3. In the context of CCA, Xj_1 is the best predictor of the X space while Yl_1 is the easily predicted linear combination of the Y space. Xj_2 is the best predictor of the residual X space and Yl_2 is the easily predicted linear combination of the residual Y space. Similar arguments hold for other pairs as well.

The CCA technique reduces to the OLS technique if the Y-block contains only one variable. When there are multiple variables in the X and Y blocks. rather than looking at the $nx \times ny$ possible correlations, it is enough to concentrate on and interpret the $n \leq min(nx, ny)$ canonical correlations and variates. In this sense, CCA can be considered as a dimension reduction technique. Determination of the number of canonical variates needed to adequately represent the association between the two sets of variables is an important decision to be made - a statistical test of significance of the canonical correlations is given in Sharma (1996).

A CCA based black-box modelling technique for multivariable systems is described in the next chapter. The Akaike Information Criterion (AIC) will be employed in order to determine the number of canonical variates to be used in the model.

1.7 Partial Least Squares

The linear partial least squares technique has established itself as a robust alternative to the standard least squares (multiple linear regression) method in the analysis of correlated data. First proposed by Wold (1966), this method has been applied to analyze data in a variety of disciplines such as sciences, social sciences, engineering and medicine. A tutorial description of PLS along with a simple example has been provided by Geladi and Kowalski (1986a, 1986b); for the theoretically inclined reader, Manne (1987) and Höskuldsson (1988) provide an excellent analysis of the mathematical properties of the algorithm. In fact. the knowledge and use of PLS has become so commonplace that it warrants no fundamental introduction.

In PLS, the goal is to arrive at a stable estimate for C (see equation (1.1)) while performing data compression on both the X and Y blocks. Thus the correlational structure of both the X and Y blocks is considered. The principal components (latent variables) for the X-block are constructed with reference to the Y-space. Therefore. a compromise solution that takes into consideration both the stability (via dimension compression) and the model fit aspects of the regression problem (through the construction of X-block latent variables with reference to the Y space).

1.7.1 A Simplistic Overview of PLS

For practical applications of the PLS algorithm, it may be necessary to scale the X and Y blocks suitably in view of the fact that the measurement units can be grossly different. Without proper scaling, the PLS latent variables may be significantly biased towards variables with larger magnitude. Scaling may be performed using some a priori knowledge, e.g. assigning larger weights to some key variables; often, all variables are autoscaled (mean centered and scaled to unit variance). This scaling information is stored in the matrices S_x and S_y for the X and Y blocks respectively. The scaled X and Y blocks i.e., $X\,S_x^{-1}$ and $Y\,S_y^{-1}$ are then processed by the PLS algorithm. The raw plant data is assumed to be scaled in this manner in all of the development that follows.

$$S_x = \begin{bmatrix} sx_1 & 0 & 0 & \cdots & 0 \\ 0 & sx_2 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & sx_{nx} \end{bmatrix} \tag{1.13}$$

$$S_y = \begin{bmatrix} sy_1 & 0 & 0 & \cdots & 0 \\ 0 & sy_2 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & sy_{ny} \end{bmatrix} \tag{1.14}$$

In PLS, the X and Y data are decomposed as a sum of a series of rank 1 matrices as follows :

$$X = t_1 p_1^T + t_2 p_2^T + \cdots + t_n p_n^T + E_{n+1} = TP^T + E_{n+1} \tag{1.15}$$

$$Y = u_1 q_1^T + u_2 q_2^T + \cdots + u_n q_n^T + F_{n+1} = UQ^T + F_{n+1} \tag{1.16}$$

In the above representation, T and U represent the matrices of scores while P and Q represent the loading matrices for the X and Y blocks. To determine the dominant directions in which to project data, a maximal description of the covariance within X and Y is used as a criterion (see the objective function in the optimization framework discussed later). The first set of loading vectors (direction cosines of the dominant directions within the data set). p_1 and q_1, is obtained by maximizing the covariance between X and Y. Projection of the X and Y data respectively onto p_1 and q_1 gives the first set of scores vectors t_1 and u_1. This procedure is depicted by the block "PLS OUTER MODEL (1)" in Figure 1.1. The matrices X and Y are now indirectly related through their scores by the "Inner Model" which is just a linear regression of t_1 on u_1 yielding $\hat{u}_1 = t_1 b_1$. $\hat{u}_1 q_1^T$ can be interpreted as the part of the Y data that has been predicted by the first PLS dimension ; in doing so, the $t_1 p_1^T$ portion

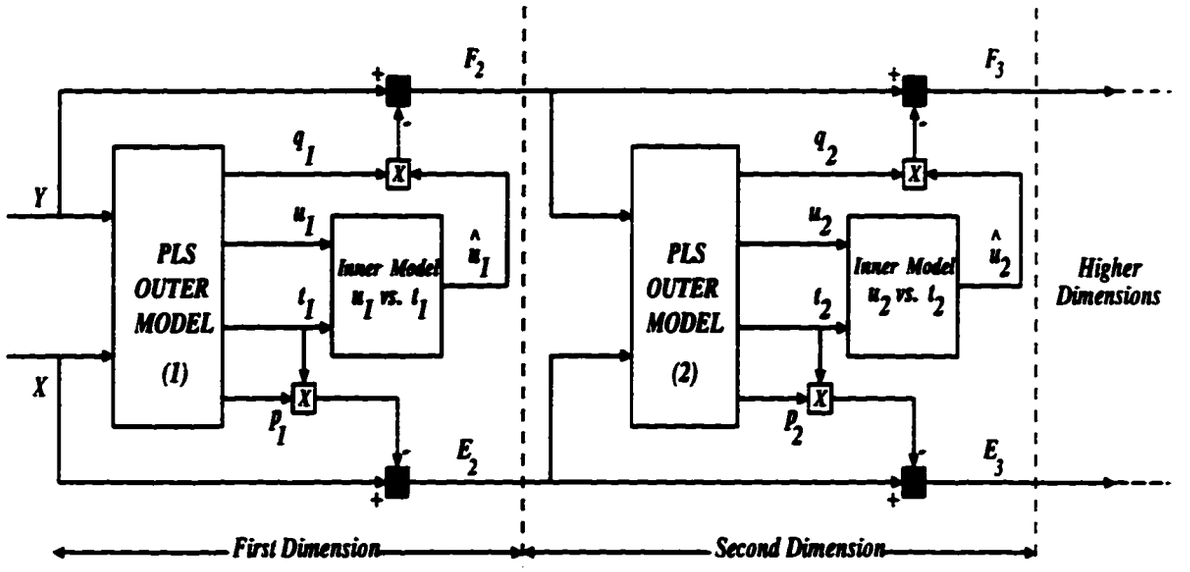

Figure 1.1: The standard linear PLS algorithm. The boxed x denotes vector outer product

of X data has been used up. Denoting $E_1 = X$ and $F_1 = Y$, the residuals at this stage are computed via the deflation process (shown as dark squares in Figure 1.1) :

$$E_2 = X - t_1 p_1^T = E_1 - t_1 p_1^T$$

$$F_2 = Y - \hat{u}_1 q_1^T = Y - b_1 t_1 q_1^T = F_1 - b_1 t_1 q_1^T$$

The procedure of determining the scores and loading vectors and the inner relation is continued (with the residuals computed at each stage) until the required number of PLS dimensions (n) are extracted. In practice, the number of PLS dimensions is determined based on the percentage of variance explained or by the use of statistically sound approaches such as cross validation (explained in the PCA context). The directions considered irrelevant in the data sets (such as noise and redundancies) are confined to the error matrices E_{n+1} and F_{n+1}.

From a practical viewpoint, PLS can be considered as a technique that breaks up a multivariate regression problem into a series of univariate regression problems. The original regression problem is handled by constructing 'n' inner relationship models (usually, $n \ll nx$). In addition to the PLS outer model (cf. equations 1.15 and 1.16), we can write the following equation for describing the inner model of the PLS technique :

$$Y = TBQ^T + F_{n+1} \tag{1.17}$$

In certain versions of the PLS algorithm, the regression coefficients b_i (i=1, $\cdots$, n) are

absorbed into the corresponding q_i vectors. In such cases $b_i = 1 \forall i$; thus B is an identity matrix (note that B is a diagonal matrix with the b_i's as the diagonal elements). The PLS technique has also been cast in the powerful and well known framework of Singular Value Decomposition (Wise, 1991). It has also been analyzed as an eigenvalue and eigenvector problem (Höskuldsson, 1988) where the mathematical and statistical properties of the PLS algorithm have been enumerated. It has been shown that the latent variables t_i and u_i (i=1,..,n) generated by the PLS algorithm form an orthogonal basis for the X and Y spaces respectively.

1.7.2 An Algorithmic Description of PLS

Of all the multivariate techniques that have been considered here, only the PLS technique is computationally iterative. In PCA and CCA, the linear combinations of the X and Y spaces were derived by *one* singular value decomposition of an appropriate matrix. In contrast, for PLS the linear combinations are generated through successive SVD's of certain residual matrices. With the *kernel* approach, the iterative techniques present *no significant computational overload* as compared to the non-iterative techniques. For some recent results on fast PLS algorithms the reader is referred to Dayal (1996).

Let X_i and Y_i denote the residual X and Y spaces after the i^{th} PLS dimension (linear combination) has been extracted. The original X and Y spaces are written as X_0 and Y_0 respectively.

- Objective : Find (at each dimension i) a linear combination pair of X_i and Y_i that have maximum covariance between them.
- Objective function : $\max \left\{ j_i^T \Sigma_{x_{i-1} y_{i-1}} l_i \right\}$
- Constraints :

  $j_i^T j_i = 1$

  $l_i^T l_i = 1$
- Solution : j_i and l_i are respectively the first left and right singular vectors of $\Sigma_{x_{i-1} y_{i-1}}$
- The complete algorithm is given below

  1. Start with $\Sigma_{x_0 y_0}$
  2. Get j_1 and l_1
  3. Iterate as follows for other PLS dimensions

     For i=2 to nx do :

     Obtain residual as follows

$$\Sigma_{x_{i-1} y_{i-1}} = \left(I - \frac{\Sigma_{x_{i-2} x_{i-2}} j_{i-1} j_{i-1}^T}{j_{i-1}^T \Sigma_{x_{i-2} x_{i-2}} j_{i-1}} \right) \Sigma_{x_{i-2} y_{i-2}}$$

j_i is the first left singular vector and l_i the first right singular vector of $\Sigma_{x_{i-1}y_{i-1}}$

End

- Remarks :

1. The J matrix from the PLS algorithm is $J_{nx,PLS} = [j_1 \mid \cdots \mid j_{nx}]$.
2. The linear combinations, $t_i = X_{i-1}j_i$ and $u_i = Y_{i-1}l_i$ (i=1, .., nx) generated by the PLS algorithm described above are defined in terms of the residual X and Y spaces. This clouds the interpretation of the linear combinations, since we do not know what the residual data matrices contain. The linear combinations t_i (i=1, .., nx), form an orthogonal basis for the X space.
3. To relate the linear combinations generated by PLS to the original X space as $T = X_0R$, we need to regress T on X_0. Doing so gives,

$$R = X_0^{\dagger}T = X_0^{\dagger}[X_0j_1 \mid X_1j_2 \mid \cdots \mid X_{nx-1}j_{nx}]$$

where $X_0^{\dagger}$ indicates the pseudoinverse of the matrix X_0. Some alternate expressions can be found in de Jong (1993).

1.7.3 PLS Estimates for the Parameters of the Linear Model

Equation (1.17) can be rewritten by defining[6] $T = XR$. In the PLS algorithm, each of the weight vectors j_i that are used to define the score vectors t_i applies to a different matrix of residuals E_i (i = 1, $\cdots$, n) as :

$$t_i = E_ij_i \tag{1.18}$$

This poses a difficulty in the interpretation of the PLS score vectors, because what is left in the residual matrix E_i at each stage is not clear. For example, some X variables dominate the first few factors and some later. Recognizing this, de Jong (1993) provided the following expression relating the score vectors in terms of the original X matrix.

$$T_{N\times n} = X_{N\times nx}R_{nx\times n} \tag{1.19}$$

The matrix R can be expressed in terms of the P and J matrices as $R = J(P^TJ)^{-1}$. Combining equations (1.17) and (1.19), the following can be obtained

$$Y = XRBQ^T + F \tag{1.20}$$

[6]Note that when all possible PLS components are extracted, the R and P matrices are related as : $R^{-1} = P^T$.

Relating equations (1.20) and (1.1), we get the PLS estimate of C as

$$\hat{C}_{PLS} = RBQ^T \tag{1.21}$$

Once the PLS model is obtained using the data obtained from normal plant operations, it can be used for predicting the quality variables in an inferential framework. The PLS matrices (scores and loadings) can be used for fault detection and isolation in exactly the same way as the PCA model matrices were used. It must be borne in mind that the predictions provided by the PLS model are reliable as long as the plant-model mismatch is insignificant. In the case of time varying plants, it may be necessary to use the recursive versions of the PLS algorithm (e.g. Dayal, 1996).

While dealing with nonlinearities in the data, two approaches are possible. The first approach is to include the nonlinear variables (such as squares, exponentials, logarithms) in the appropriate data matrices and use the standard linear PLS procedure described above. This would involve dealing with *wider* matrices (for a X matrix with 10 variables, ${}_{10}C_2 = 45$ second order variables are possible). In such circumstances, the higher order variables tend to dominate the PLS dimensions (Wold *et al.*, 1989) resulting in poor models. An attractive alternative is to move the nonlinearities to the PLS inner model. In the nonlinear PLS algorithm of Wold *et al.* (1989), the score vectors of the X and Y spaces i.e., t_i and u_i are related via a quadratic model (i.e., the inner model in Figure 1.1 is now a polynomial model instead of a linear model). It is clearly evident that this strategy can do little when the data comprises of other types of nonlinearities. With their demonstrated utility in approximating arbitrary continuous functions to any desired accuracy, neural networks can be a useful tool in nonparametric modeling studies. To this end, Qin and McAvoy (1992b) proposed an integration of neural networks with the standard PLS algorithm. Their approach preserves the outer relation in linear PLS so as to have the robust prediction property; however, neural networks are employed as the inner regressors. A direct benefit of such a strategy is that only a SISO (single-input single-output) network is trained at a time. This is not only easier than training a MIMO (multi-input multi-output) network, but also circumvents the over-parameterization and *convergence to local minima problems* one usually experiences with a MIMO network. In any case, it is clear that static nonlinearities in data are elegantly handled by incorporating either a parametric (e.g. quadratic polynomial) or a nonparametric (e.g. neural network) regression in the inner relationship of the PLS model.

Extension of the basic PLS technique (discussed above) to the domain of dynamic systems (linear and nonlinear) will be dealt with in chapter 3.

1.8 Industrial Case Studies

1.8.1 Application of PLS to the Estimation of Distillation Tower Top Composition

The partial least squares technique has been employed to obtain an estimator for the top composition in an industrial distillation column. Significant improvement was obtained in the control of the product quality by using the frequent estimates from the PLS model rather than using the measurements from an online analyzer. In this application, PLS was used in the selection of the important variables as well as in the construction of the model.

The large sampling intervals and time delays associated with online analyzers and offline laboratory procedures, make the product composition control of the distillation columns very difficult. To overcome these problems, inferential models that provide estimates for these variables based on other process measurements such as pressures, temperatures and flow rates are commonly employed. Though the most popular way is to use one temperature measurement that adequately represents the characteristics of the product composition, significantly better predictions can be obtained using multiple measurements such as several tray temperatures, steam and reflux flow rates (Weber and Brosilow, 1972). There is usually a strong collinearity amongst these measurements - use of techniques such as PLS is ideal under such conditions. In fact, the PLS technique has been applied on a pilot scale plant (Mejdell and Skogestad, 1991). In their study, Mejdell and Skogestad developed an inferential model using several temperature measurements from the column.

Process Description

The focus of our study is a rectification tower which separates a distillate product from the heavy key component. The feed to this tower is a mixture containing 45 wt % of light key product, 50-53 wt % of heavy key component and 1-3 wt % of other heavy impurities. A gas chromatograph (GC) is in operation to measure the concentration of the heavy key component in the distillate stream - however, the sampling period of the GC (90 minutes) and the process delay (roughly 70 minutes) makes composition control unsatisfactory.

The main disturbances to the operation of this tower include the feed composition, feed flow rate and the ambient temperature. The feed composition changes are small and slow and the feed flow rate remains steady (changes once in about 3 months). Therefore, these do not affect the tower operation severely. On the other hand, due to the poor performance of the pressure control loop, the ambient temperature turns out to be the major disturbance resulting in a daily oscillation of the distillate purity. The strategy of manipulating the distillate flow rate to compensate for this oscillatory behaviour often proved inadequate with the product not matching the specifications (heavy key component $\leq$ 1000 ppm in distillate). The real problem was to minimize the effects of changes in ambient temperature.

The distillation column is a packed tower consisting of three beds. A schematic diagram

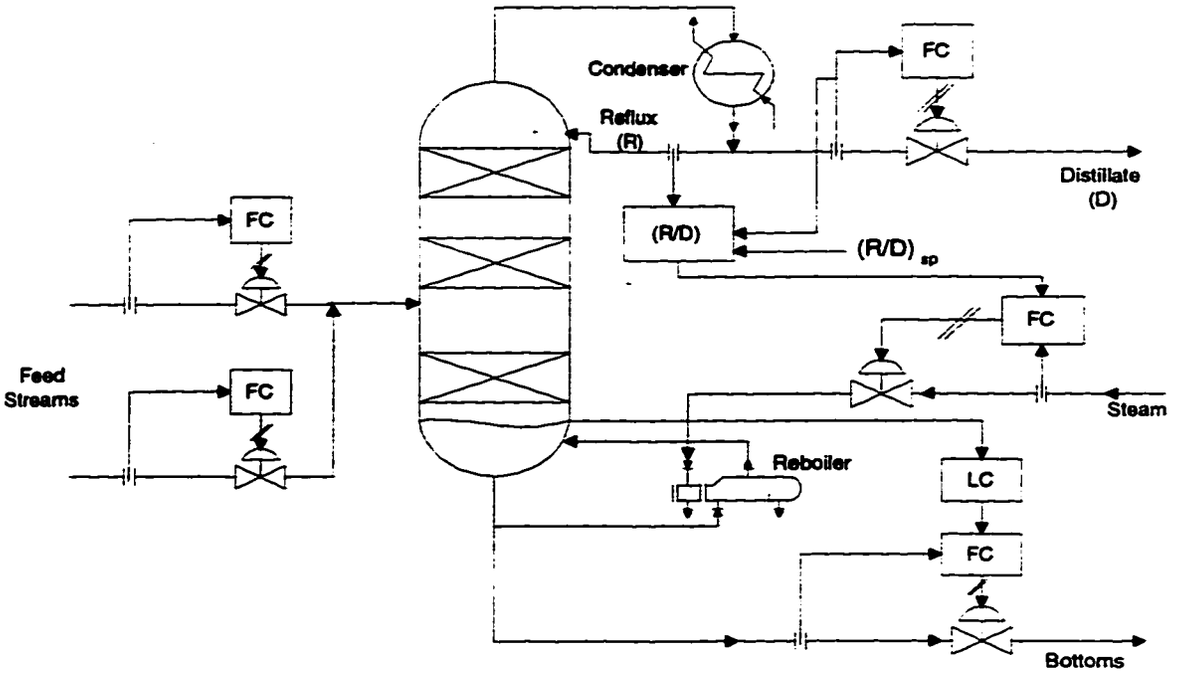

Figure 1.2: The original control strategy for the Mitsubishi distillation column

of the process with some of the control systems in place is shown in Figure 1.2. There are two reboilers for the column (only one of them is shown in the schematic). The purity of the top product was originally controlled by manipulating (manually) the distillate flow rate based on the infrequent GC output. The reflux ratio was kept constant by manipulating the steam flow to the reboiler. An empirical model was developed to estimate the composition of the heavy key component in the distillate stream using the top and bottom temperatures and reflux flow rate. The match between the output of this model and the GC readings were poor and was considered inadequate for inferential control. Since PLS can be applied to collinear data, it was decided to obtain a *soft sensor model* taking into account all the recorded variables. Subsequently, a variable selection procedure will be used to prune the variable set and select only the important variables. These variables will be used to construct the final model (the composition estimator).

In Table 1.1, the 26 variables that are available for building the PLS based inferential model are listed. Data on these 26 variables are logged on to the database every 12 minutes. In contrast to the work by Mejdell and Skogestad who used only temperature measurements. we have used pressure and flow rate measurements as well. This is in recognition of the fact that the pressure swings are a major concern for this tower. The gas chromatograph readings arrive every 90 minutes. The dead time is large but is not known exactly and was used as one of the tuning parameters in the model - the estimated dead time was the one that gave the minimum squared prediction error. The X block thus comprised of 26 variables and the Y block had one variable. All other dynamics were assumed negligible.

Table 1.1: Process variables for the Mitsubishi distillation column

Serial No.	Description	Remarks
1	Top Section Temperature	
2	Enrichment Section Temperature	
3	Mid Section Temperature	
4	Bottom Section Temperature	
5	Reboiler 1 Temperature	Gas phase
6	Reboiler 2 Temperature	Gas phase
7	Condenser Temperature	Gas phase
8	Condenser Temperature	Liquid Phase
9	Feed Flow Rate	Stream 1
10	Feed Flow Rate	Manipulated value of variable 9
11	Feed Flow Rate	Stream 2
12	Feed Flow Rate	Manipulated value of variable 11
13	Distillate Flow Rate	
14	Distillate Flow Rate	Manipulated value of variable 13
15	Steam Flow Rate	
16	Steam Flow Rate	Manipulated value of variable 15
17	Bottoms Flow Rate	
18	Bottoms Flow Rate	Manipulated value of variable 17
19	Reflux Flow Rate	
20	Inner Reflux Flow Rate	Calculated value
21	Bottom Level	
22	Bottom Level	Manipulated value of variable 21 (cascaded to bottoms flow)
23	Tower Pressure (Top)	
24	Tower Pressure (Top)	Manipulated value of variable 23
25	Tower Pressure (Bottom)	
26	Reflux Ratio	

As already indicated, proper scaling is necessary to ensure better results from the application of the PLS technique. Scaling of the variables may be performed using some a priori process information. Often, such knowledge is usually unavailable and the variables are autoscaled (mean centered and scaled to unit variance). Martens and Naes (1989) showed that autoscaling tends to amplify the noise of nearly constant variables. To overcome this problem, they suggest estimating the noise and compensating for it in the scaling procedure (the interested reader is referred to the cited reference for further details). Figure 1.3 shows the estimated noise levels for each variable (using the residuals in X variables). As expected, the noise magnitude of the top stage temperature (variable 1) is larger than those of the other temperature measurements (variables 2 through 6). The noise level in the flow rate measurements (variables 9 through 20) is even larger. Comparison of estimated model coefficients (using all the 26 variables) using auto scaling and the Martens-Naes scaling is shown in Figure 1.4. No significant differences are noticed in the model coefficients using

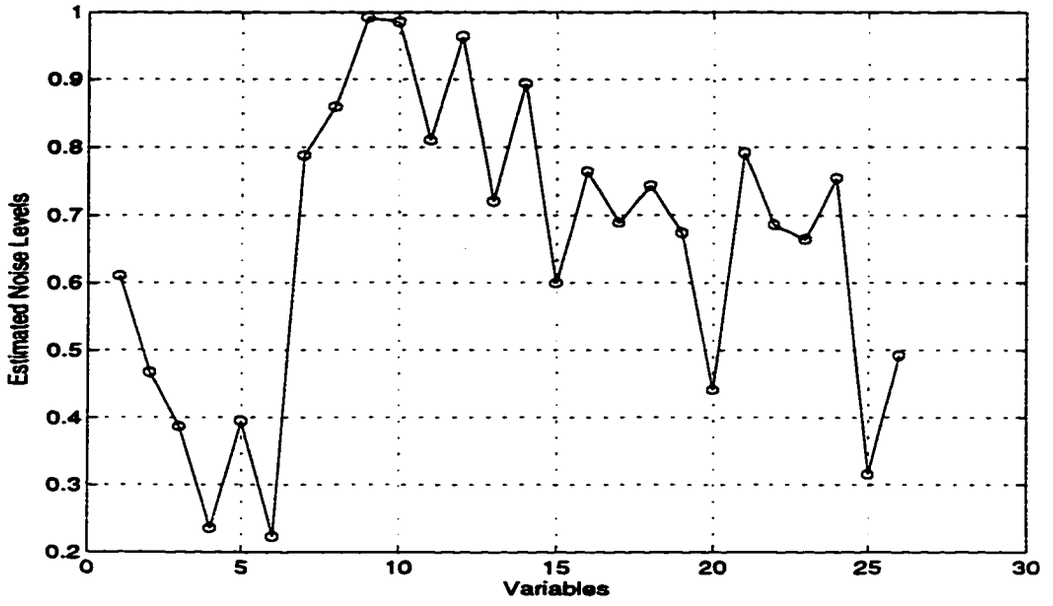

Figure 1.3: Estimated noise level for the variables

these two methods. Generally, the model coefficients for the variables tend to be larger with the Martens-Naes scaling.

Selection of Important Variables

Preliminary analysis with the data set indicated that some of the X variables have little to do in the prediction of Y. Excluding these variables may improve the predictive ability of the model. Further a model involving fewer number of variables is appropriate in an industrial setting. This was pursued as the next goal in this case study.

The engineers' delight would be to use the available process knowledge to pick the important variables. If the process characteristics are not well known, this approach may not result in models with good predictive capability. Supplementing process knowledge with statistically sound procedures would be the ideal alternative. In the statistics area, stepwise regression procedures (step-up or step-down) are used to assess the merit of each X variable in predicting Y. In doing so, any variable that contributes little is removed from the data set. For data sets with a large number of variables, this may result in a combinatorial problem. Lindgren *et al.* (1994) proposed an interactive variable selection procedure which involved re-weighting each latent variable by deleting the less influential variables. The potential problem with this approach is that some of the X variables which were considered unimportant at an earlier stage may reappear as an influential variable at a latter stage. However, this technique does not suffer from the combinatorial problem that was discussed earlier.

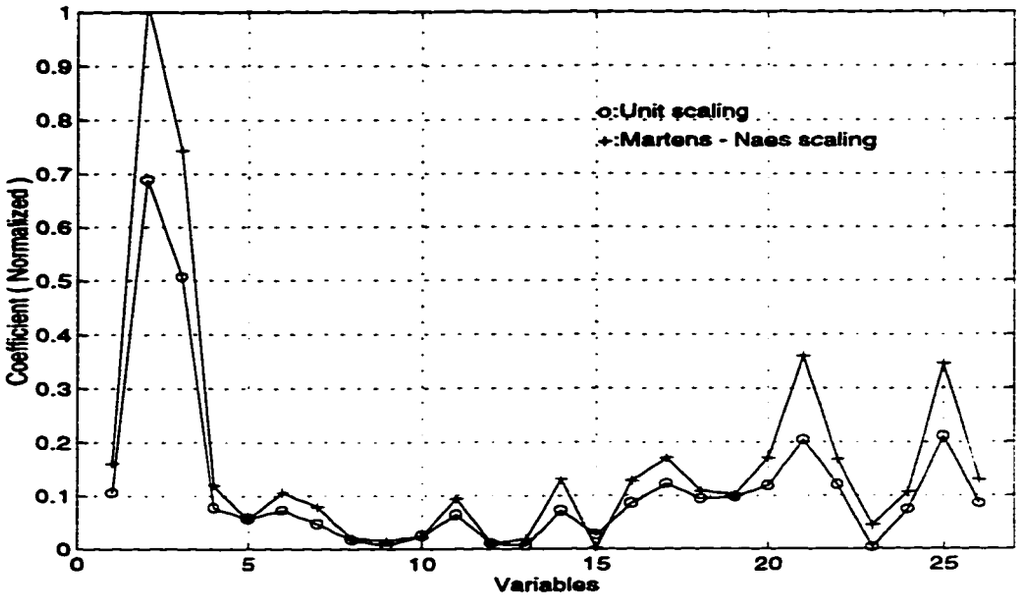

Figure 1.4: Normalized regression coefficients for the model using the two scaling procedures

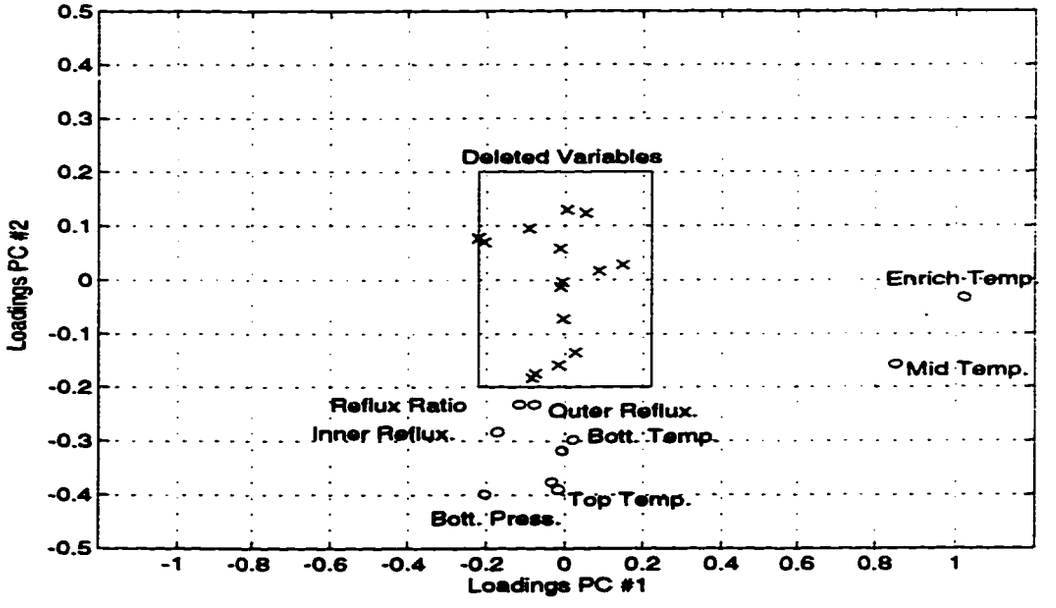

Figure 1.5: Instructive variable selection

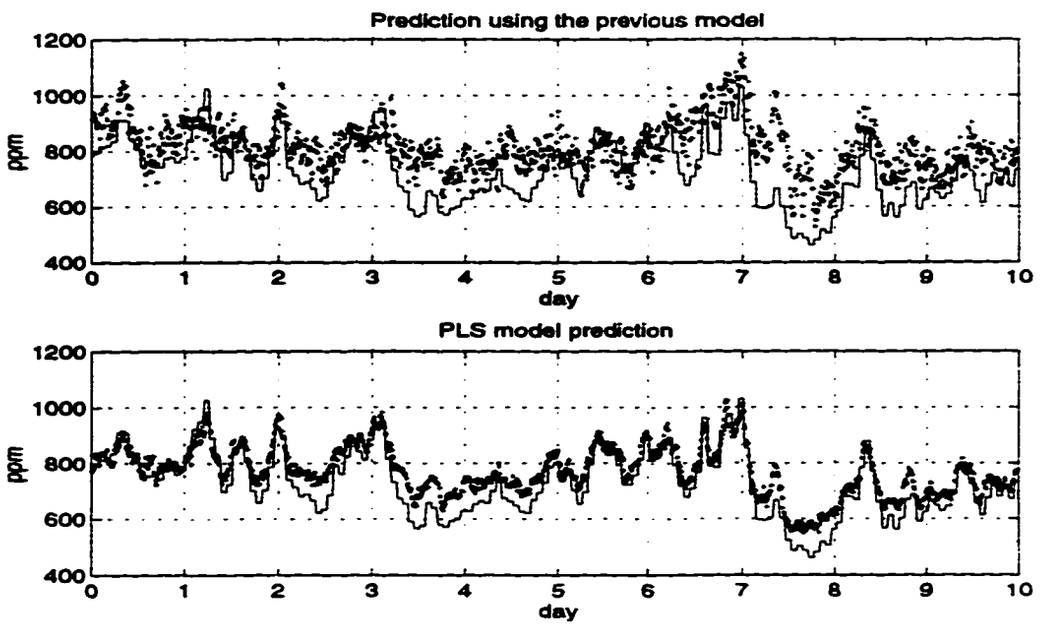

Figure 1.6: Predictions obtained using the old empirical model and the new PLS based empirical model. The solid lines are the output from the gas chromatograph and the dots represent the model predictions

For this work, a simple instructive variable selection procedure was adopted. A loadings plot involving the weights attached to the X variables in the first two PLS dimensions was used to delete the unimportant variables (see Figure 1.5). At this stage, most of the flow rate variables (feed, steam and bottom), bottom level and top pressure were deleted. This was consistent with the available process knowledge. Only 10 variables were considered for further analysis. A step wise regression procedure resulted in a final choice of 4 variables - the enrichment stage temperature, middle stage temperature, bottom pressure and the reflux flow rate.

Modelling and Control Results

A final PLS model was constructed using these 4 variables. Three PLS dimensions were sufficient to model most of the output data. The estimated time delay was 84 minutes. Figure 1.6 compares the model predictions obtained using the previously empirical model (using the top and rectification stage temperatures and reflux flow rate) and the new PLS based model. It is clearly evident that the PLS model provides a significantly better fit largely due to the inclusion of the bottom pressure in the model. The PLS modelling procedure clearly indicated the importance of the bottom pressure and provided a robust pressure compensated model for use in inferential control of the column.

The PLS model was implemented as part of the inferential control strategy to regulate

the impurity level in the distillate. The new control strategy is shown in Figure 1.7. The PLS model provides frequent estimates of the top composition based on the process measurements. The composition controller computes a setpoint for the reflux ratio which is used in conjunction with the measured reflux flow rate to provide a setpoint to the flow controller that regulates the distillate flow. This sequence of control actions help regulate the top product purity. For some operational reasons (a new control strategy is being developed to regulate the bottoms purity), the cascade control on the steam flow has been removed. Figure 1.8 indicates that significant improvement in the control was obtained with the new PLS model. The daily oscillations seem to have disappeared and the impurity levels were within acceptable values.

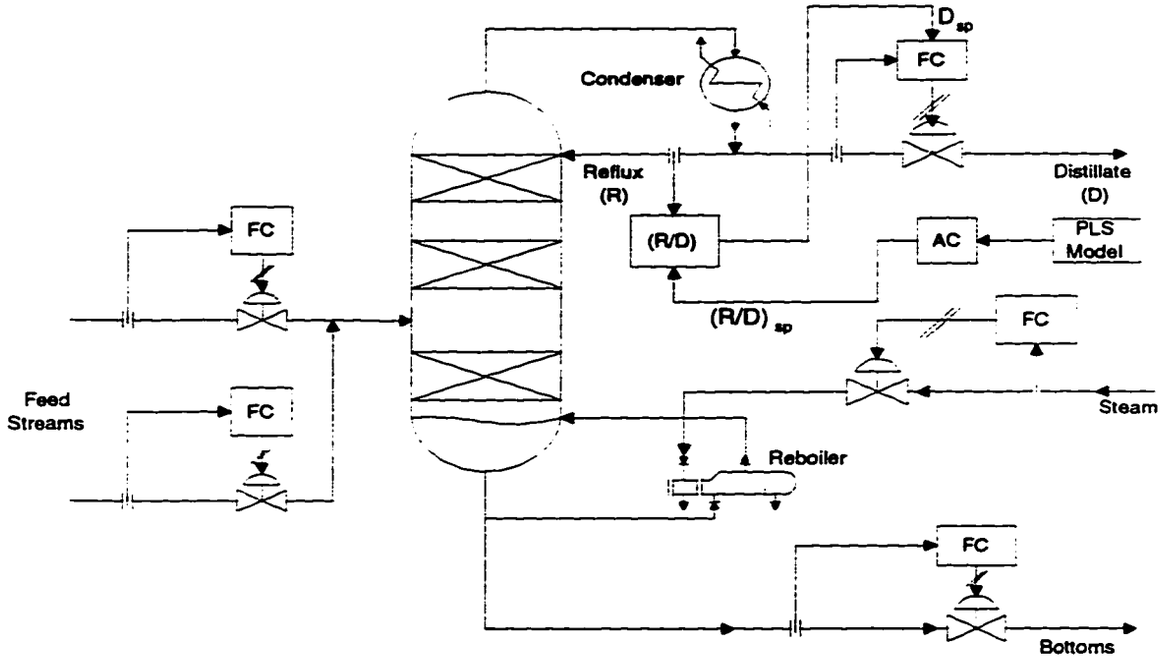

Figure 1.7: The PLS model based inferential control strategy

Final Remarks

A robust and reliable process model was obtained using the PLS technique. Inferential control with this new model improved the product quality and reduced the consumption of steam. Though the results mentioned here are specific to the distillation column studied. the methodology is fairly general and applicable to large (and possibly collinear) data sets. Extensions to multiple data blocks (comprising of primary and secondary measurements) are also available.

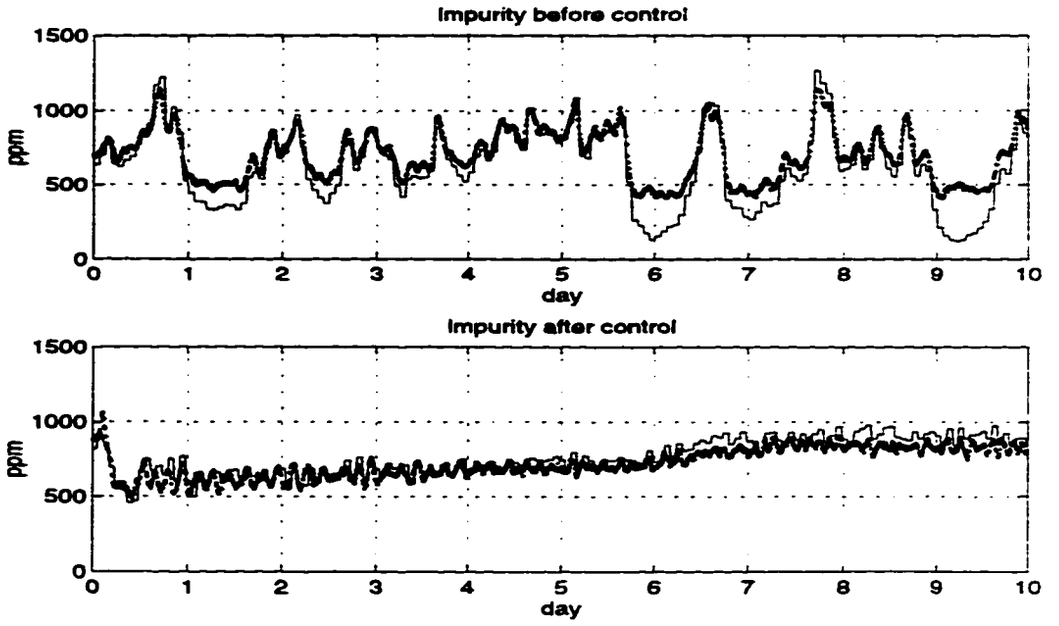

Figure 1.8: Improvement in product purity control with the new PLS based inferential model. The solid lines are the output from the gas chromatograph and the dots represent the model predictions

1.8.2 Monitoring of Process Operation using PCA

The PCA based diagnosis tools described in an earlier section are now applied to data collected from the Shell styrene unit at Scotford. Canada. A brief overview of the process and a description of the process variables is provided. Development of the PCA model and its utility in process monitoring is described in detail.

Process Description

Firstly, ethylbenzene (EB) is produced via an alkylation process involving the reaction of benzene with ethylene. The EB process involves two major steps :

- Production of crude EB by alkylation of benzene with ethylene
- Recovery and purification of pure EB

The desired alkylation reaction to produce EB is

$$C_2H_4 + C_6H_6 \overset{Mobile\,Catalyst}{\Longleftrightarrow} C_6H_5 \cdot C_2H_5 \tag{1.22}$$

though a number of undesired by-products are produced owing to side reactions. These include polyethylbenzene (PEB), xylenes, ethyltoluene and cumene.

The alkylation reactor section consists of two reactors (each with four separate catalyst beds in series) in parallel, a reactor feed heater, reactor feed effluent exchangers, a startup feed vaporizer, a prefractionator column and a vent gas scrubbing system. Catalyst regeneration facilities (to deal with deactivation caused by coking) are also included. The crude alkylate product containing benzene, EB and heavies are now processed by the distillation section (comprising of four distillation columns) to effect the separation and recovery of the EB. A benzene recovery column first separates the benzene from EB and heavier components. This benzene is recycled to the alkylation reactor section. The bottoms from the benzene recovery column is processed by the EB recovery column where EB is taken as an overhead product and sent to a EB storage tank. The column bottoms - a mixture of PEB and heavies - is passed to the PEB recovery column which separates the PEB (overhead product) from the heavier polymers (bottoms). The recovered PEB is recycled to the alkylation reactor section to control the production of PEB in the reactor. The heavy polymer residues are used as fuel in the steam boilers.

The styrene monomer is derived by the dehydrogenation of ethylbenzene in the presence of a catalyst in a steam environment. The desired reaction is :

$$C_6H_5 \cdot C_2H_5 \overset{Steam, Catalyst}{\Longleftrightarrow} C_6H_5 \cdot C_2H_3 + H_2 \tag{1.23}$$

Several other side reactions are also associated with the above dehydrogenation process. About 90% of the reacted EB is converted to styrene, the rest is converted into toluene, benzene, methane, ethylene, carbon and hydrogen. The purification of the styrene monomer and separation of the by products takes place in the distillation section which is a train of four vacuum distillation towers. Inhibitors are added in order to minimize the formation of polymers.

Hourly spot values for one month (720 samples) on process variables such as temperatures, pressures, flow rates, levels etc. were available. Composition measurements were available on an infrequent and irregular basis. The original goal of this exercise was to obtain inferential models for : (1) the xylene levels in the EB reactor effluent and (2) the xylene and EB levels in the styrene fractionation section. However, only 10 samples were available from the lab analysis and this was insufficient for building the PLS based inferential models. The study was therefore restricted to a principal components analysis of the process data.

Development of the PCA Models

The EB and the styrene units were analyzed separately. Most of the process variables available in the historical database were included in the study - a few were dropped out because they contained very little useful information (no variation whatsoever over the 720 samples).

The crucial step in building the PCA model is the selection of the reference data set

- this determines the confidence limits on the monitoring charts and hence the sensitivity and reliability of the fault detection procedure. Ideally, the PCA model should be built based on data collected from various periods of *good* plant operation. In this application, the following samples were selected as defining the normal process operation :

- EB Unit : Samples 31 - 81 and 201 - 448 involving 295 variables
- Styrene Unit : Samples 1 - 260 involving 262 variables

PCA Model for the EB unit

For the EB unit, the X matrix comprised of 299 measurements of the 295 variables. The data was first mean centered and scaled to unit variance. For this data set, the eigenvalue 1 criterion suggested using 61 principal components - these 61 components also accounted for 84% of the variance resulting in about a five fold reduction of the dimensionality. Figure 1.9 shows the eigenvalue distribution versus the principal component number.

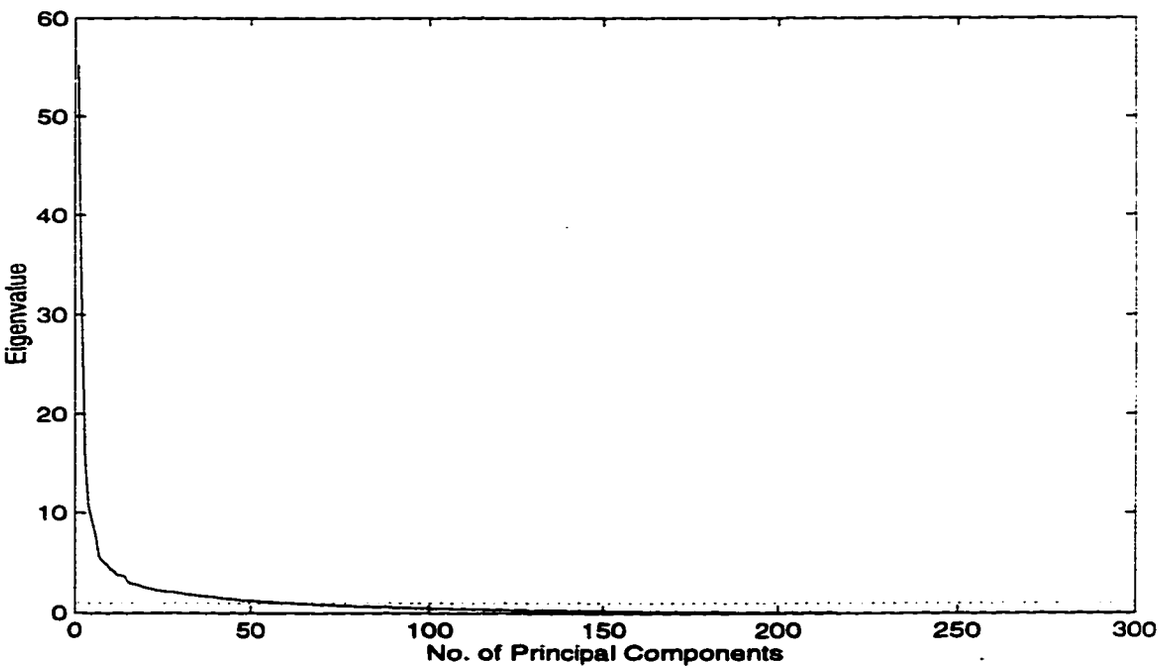

Figure 1.9: Distribution of eigenvalues for the EB unit data. The horizontal line indicates an eigenvalue of 1

The normal operating region in the principal components space is shown in Figure 1.10. Proceeding clockwise from top left portion of the Figure, we see a three dimensional view of the normal operating region - it has a near spherical shape. The scores plot (principal components 1 versus 2) indicates two clusters with the smaller one (samples 31 to 81) slightly below and to the left of the other (samples 201 to 448).

To illustrate the utility of the PCA model generated above for online monitoring purposes, the remaining samples were divided into three parts. The results are shown in Figures

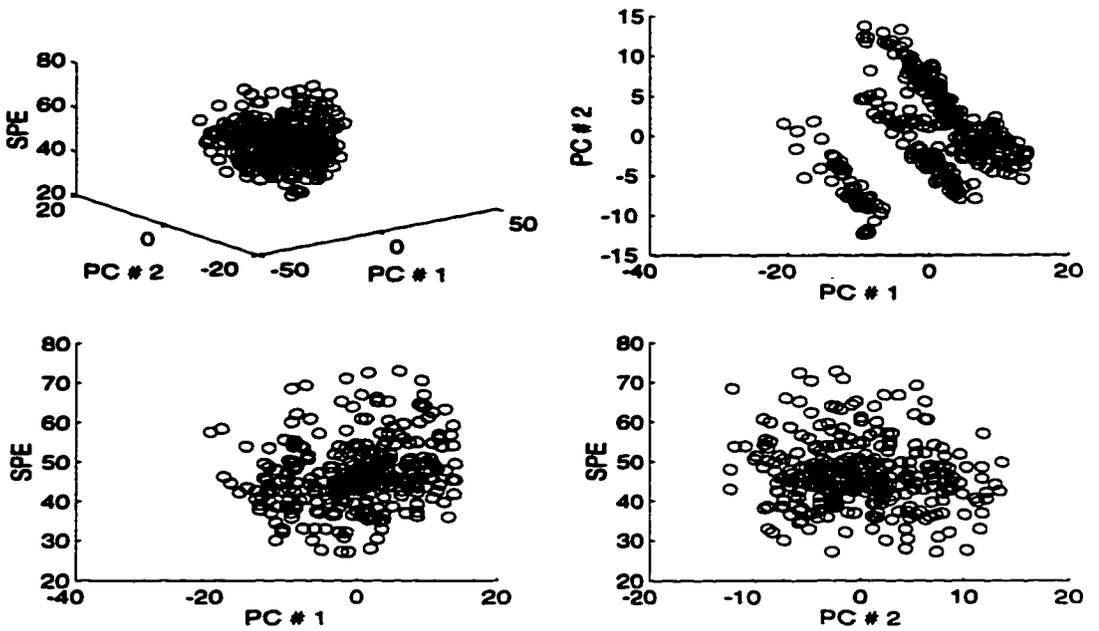

Figure 1.10: The normal operating region for the EB unit

1.11 through 1.13 (in these Figures, the normal operating data are shown as circles while the online data are represented by x, * and + respectively). In Figure 1.11, a transition is seen along principal component 2 (top and bottom right subplots show this clearly). There is also some abnormality indicated in the SPE values. These indicate that the data collected online have abnormally large values ; furthermore, there has been a breakdown of the correlational structure between the variables. The second group of samples, exhibit a similar trend (see Figure 1.12) only with larger SPE values. A different but interesting trend is shown by the third group of samples. Projection of the new data on to the principal components 1 and 2, indicates no abnormality - the samples overlap with the normal operating region almost entirely (see subplot on the top right portion of the Figure). However, the large SPE values indicate some abnormality indicating a change in the interrelationships between the process variables. This also highlights the need for including the SPE trajectories during online monitoring.

To diagnose the cause for the abnormalities, the contribution plots were inspected picking one random sample from each of the three groups. The contribution plot for a sample from group 1 (see Figure 1.14) indicate that three variables (variables 4, 17 and 207) contribute 5% or more to the overall SPE. A similar analysis for a sample from group 2 (Figure 1.15) indicates that the variables 4, 17 and 248 may be responsible for the abnormality. Low feed rate to one of the reactors appears to be the fundamental cause of the problem. In the case of group 3 (Figure 1.16), the problem seems to be associated with the variables 199 and 209. An experienced plant personnel could use this information to identify the

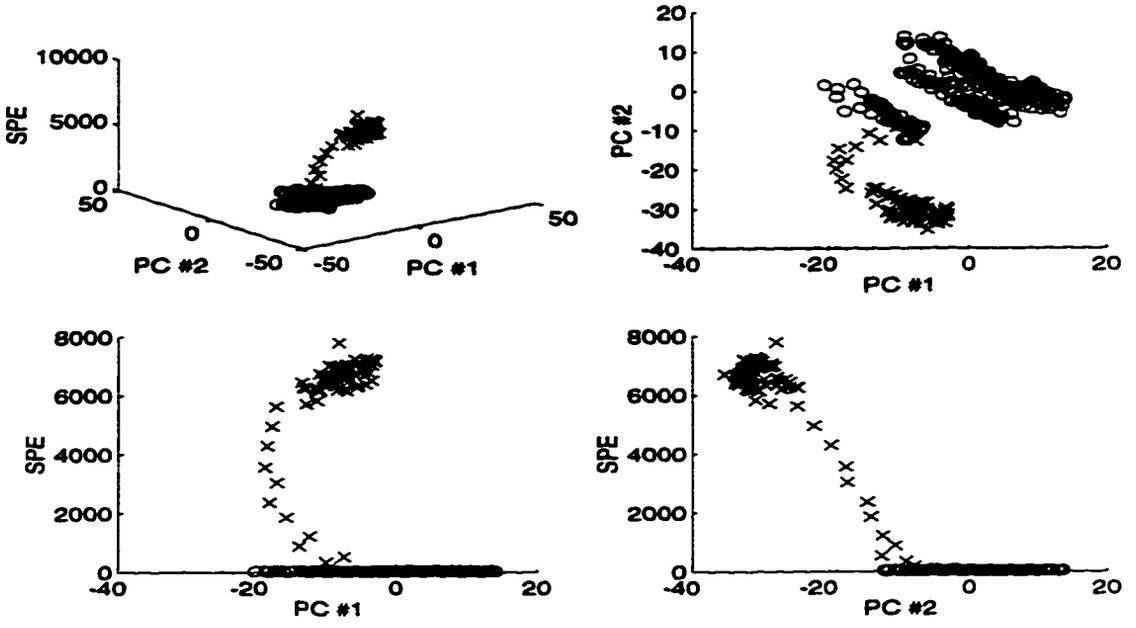

Figure 1.11: Online monitoring of the EB unit : Group 1 of data

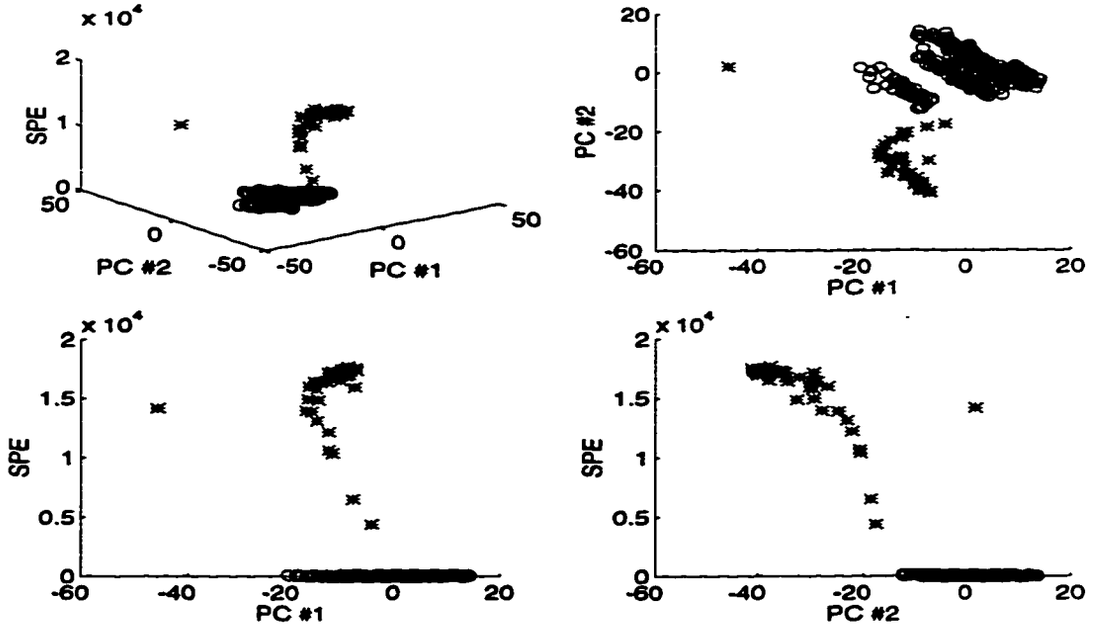

Figure 1.12: Online monitoring of the EB unit : Group 2 of data

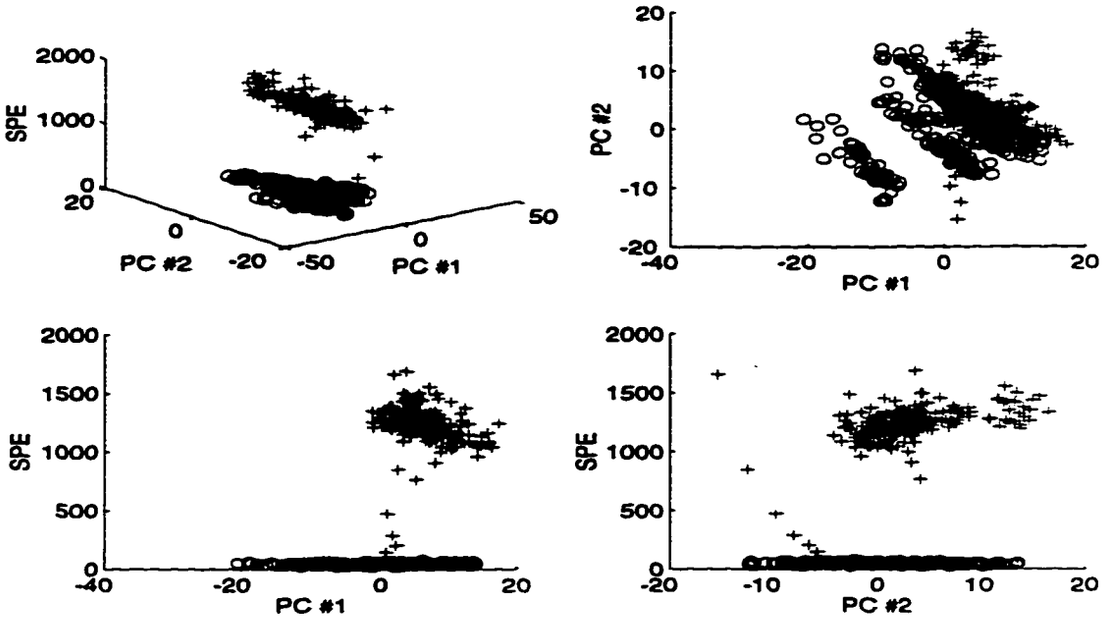

Figure 1.13: Online monitoring of the EB unit : Group 3 of data

underlying problem and effect improvements in plant operations on a continuous basis.

PCA model for the Styrene unit

The data set for the Styrene unit comprised of 260 samples and 262 variables. The PCA model was constructed using the autoscaled matrix. The eigenvalue 1 criterion suggested using 41 principal components accounting for 88% of the total variance. The distribution of the eigenvalues for this data set is shown in Figure 1.17.

Visualization of the normal operating data for the styrene unit is given in Figure 1.18. Online monitoring of the process is performed by taking this PCA model (represented by the circles in the Figure) as a reference target. Unlike the EB unit, the data from the styrene unit contained several outliers and missing data. Consequently, out of the remaining 460 samples (note that 260 out of the 720 samples have been used for model construction). some of them could not be used for online monitoring purposes. For presentation purposes. the available data was segregated into two groups and used for online monitoring.

The result of online monitoring using the first set of fresh data is shown in Figure 1.19. The current plant data are seen to conglomerate at a distance from the normal operating zone (see top right subplot). The SPE values are also large. This indicates that the process has moved to a different operating regime. The contribution plot for observations from this group indicate an abnormality due to variable 11 (the result from one such sample is presented in Figure 1.20). Variable 11 is the flow rate of the EB flush in the inhibitor section of the styrene unit.

The second set of online data portrays a different picture (Figure 1.21). This data set

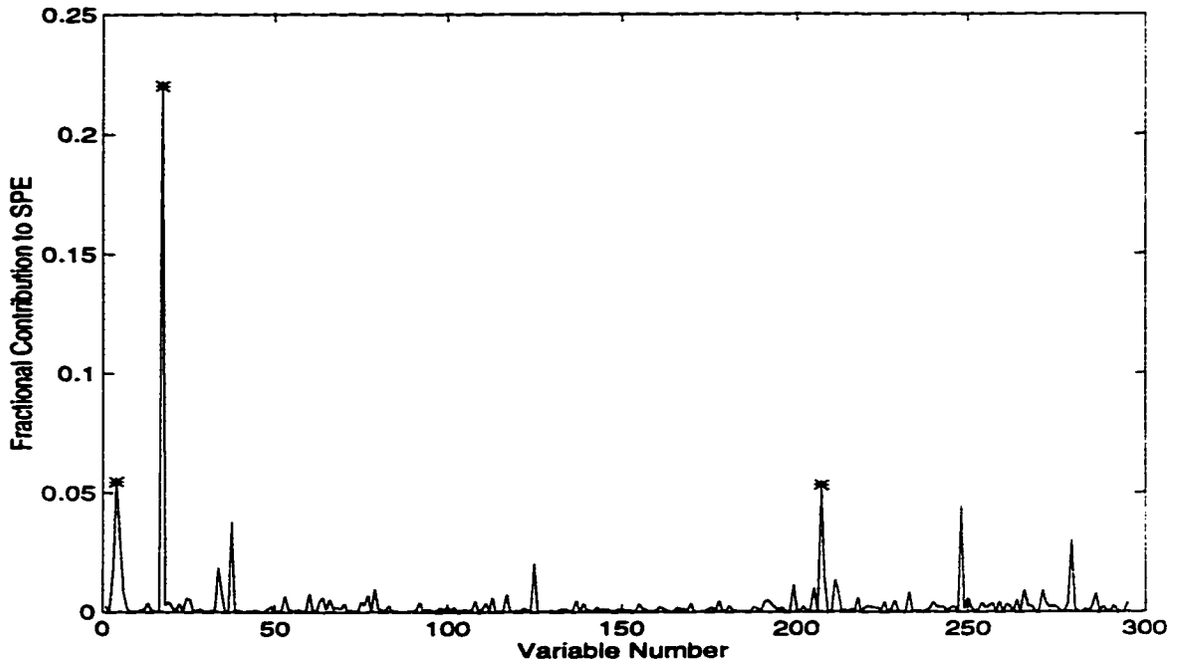

Figure 1.14: Contribution plot for a random sample from group 1 of data : EB unit

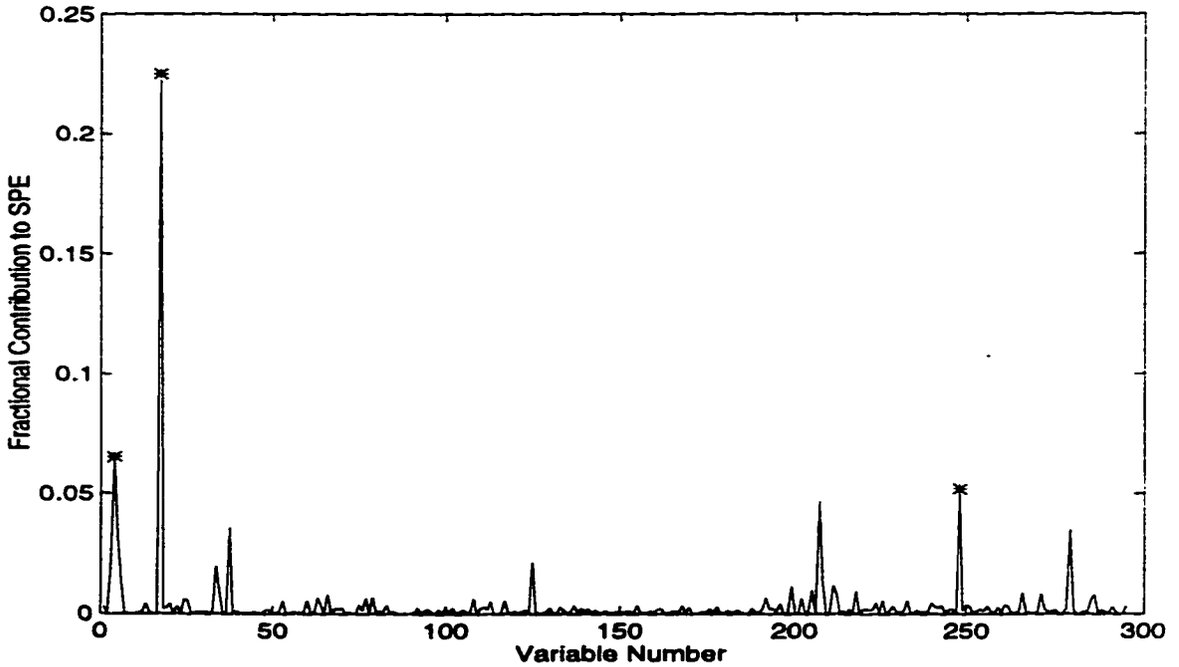

Figure 1.15: Contribution plot for a random sample from group 2 of data : EB unit

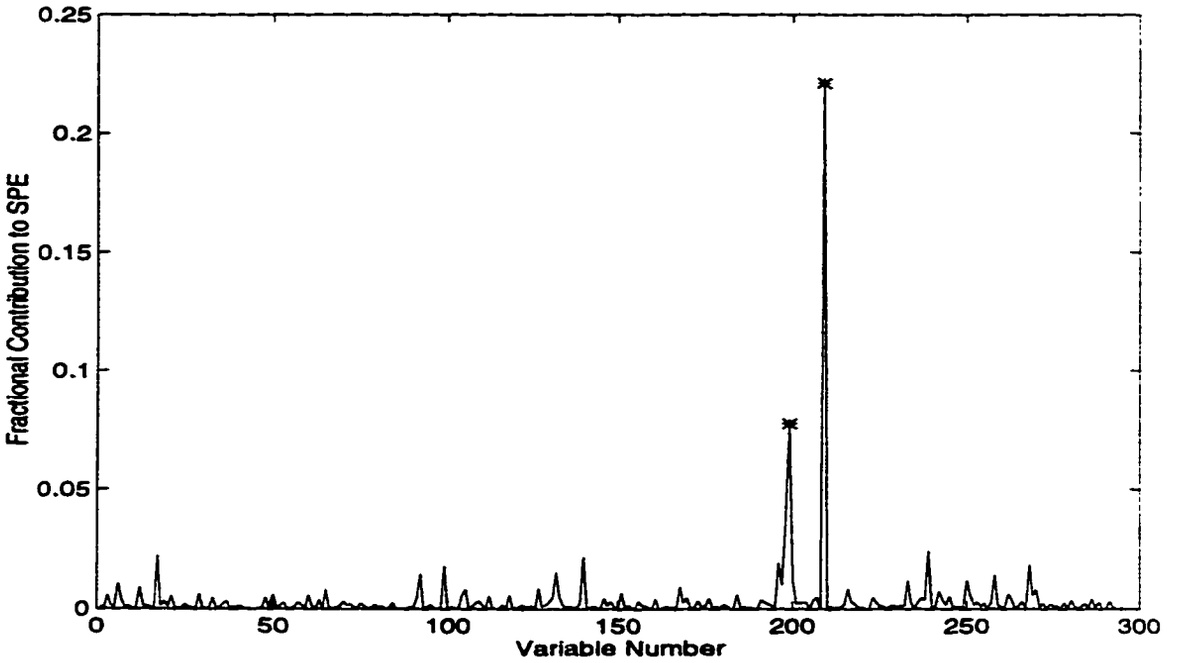

Figure 1.16: Contribution plot for a random sample from group 3 of data : EB unit

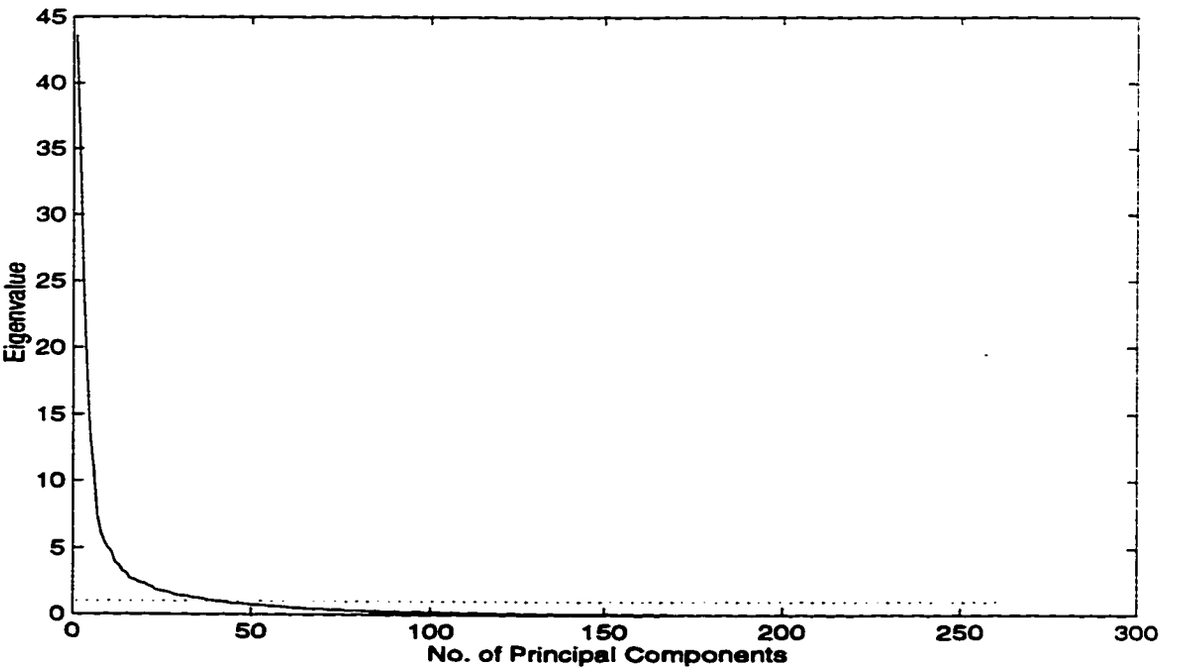

Figure 1.17: Distribution of the eigenvalues for the Styrene unit data

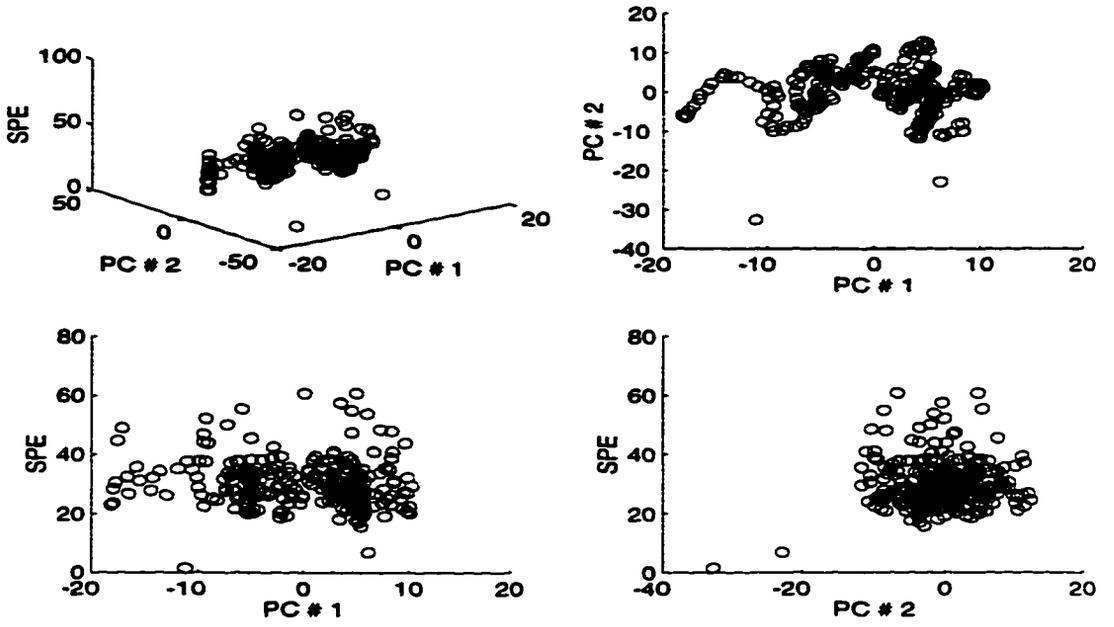

Figure 1.18: The normal operating zone for the Styrene unit

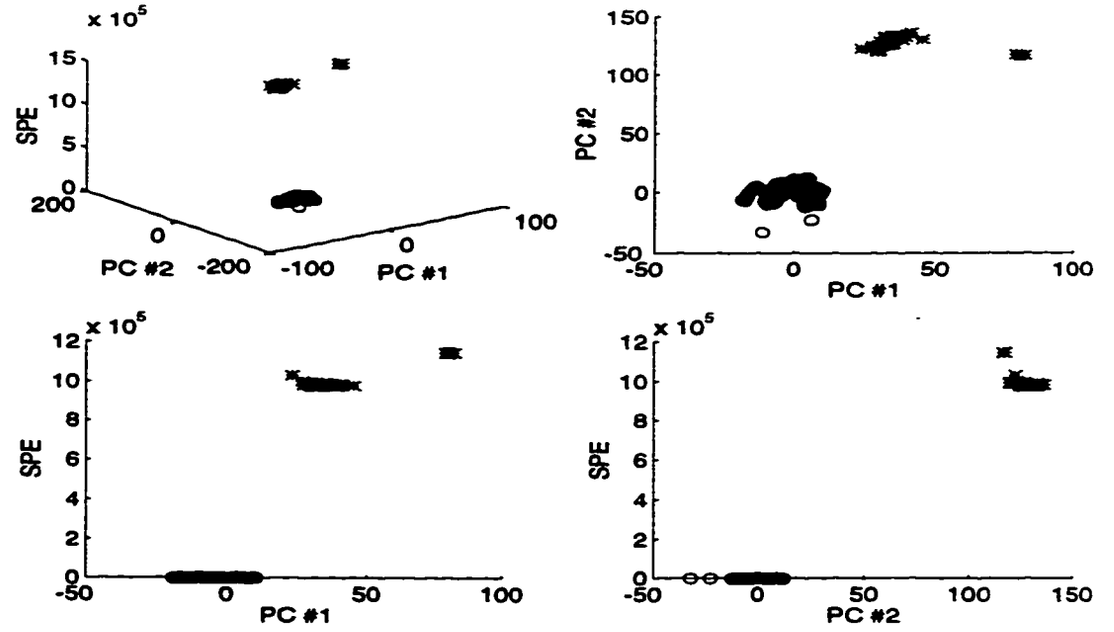

Figure 1.19: Online monitoring of the Styrene unit using the PCA model : Group 1

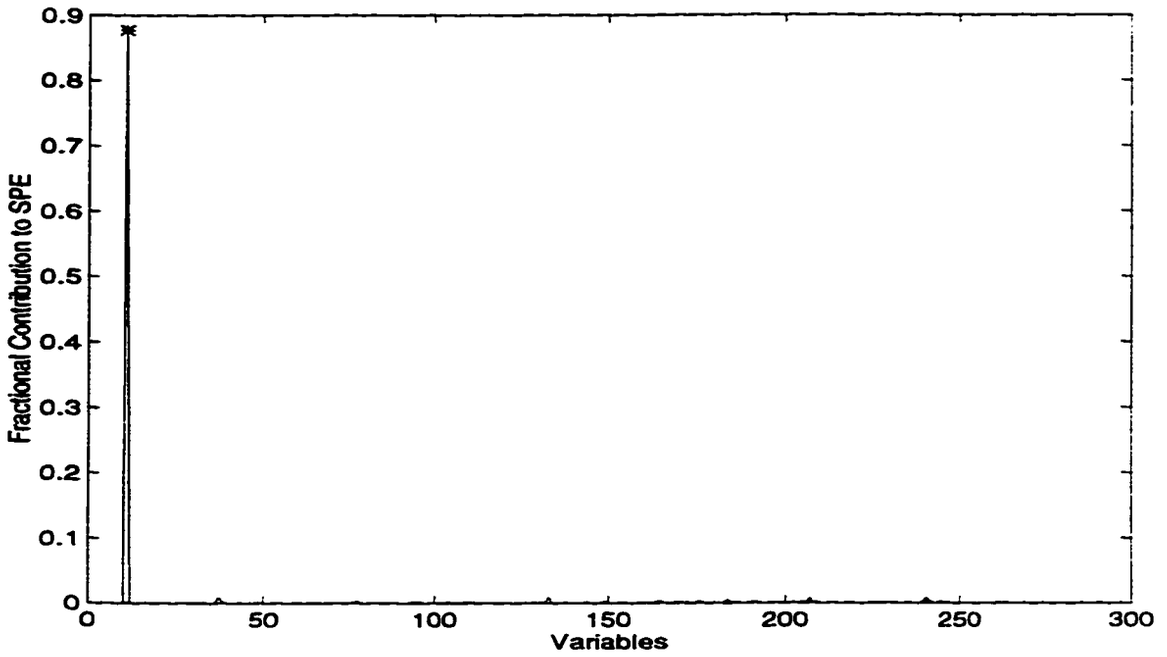

Figure 1.20: Contribution plot for a sample from Group 1 of online data : Styrene unit

represents a new operating zone - however, the normal operating zone and the new operating zone are separated along principal component 1. The SPE values for the new observations are larger in magnitude.

The contribution plot (Figure 1.22) for a sample from this group of data indicates the abnormality in variable 207.

Final Remarks

Analysis of the data from the Shell styrene unit has illustrated the utility of PCA to characterize normal process operation and to compare future runs against this reference model. In the analysis of the data, no consideration was given to the theoretical limits described by equation (1.8) - because the online data considered here violated those limits by a large margin. Loadings plot were not provided as the contribution plots were used in the isolation of faults (selection of variables using the loadings plot was demonstrated in the Mitsubishi application). As already mentioned, the inferential models based on PLS (or any other method) could not be constructed owing to insufficient data. If more data are available, future efforts could be directed in this area.

1.9 Conclusions

A tutorial introduction has been provided in this chapter to three popular multivariate techniques, namely PCA, PLS and CCA. These techniques were also analyzed in an op-

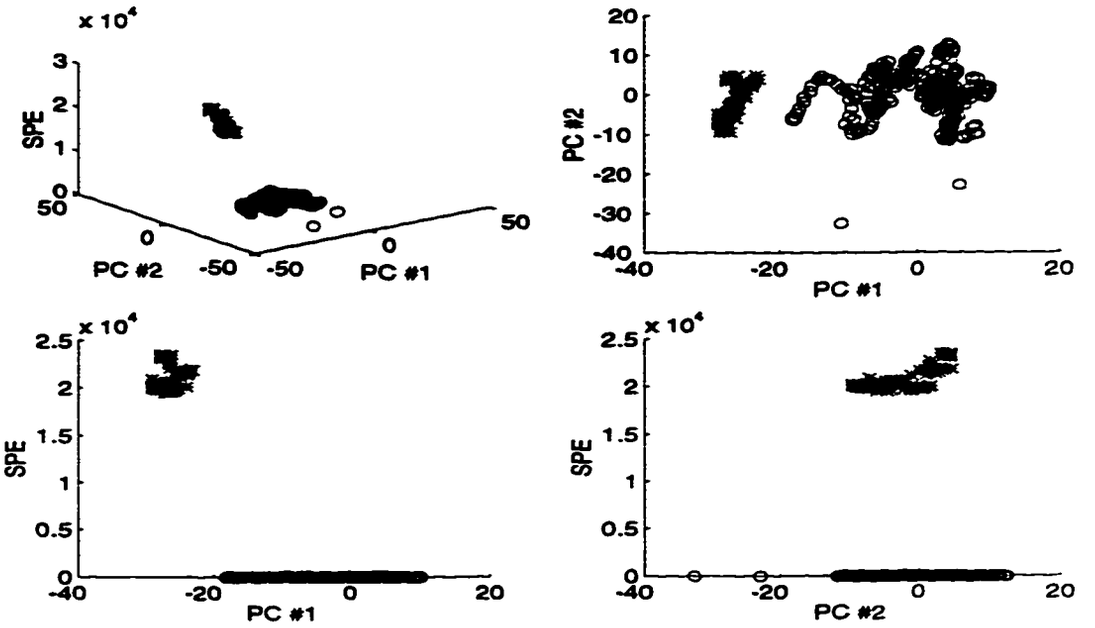

Figure 1.21: Online monitoring of the Styrene unit using the PCA model : Group 2

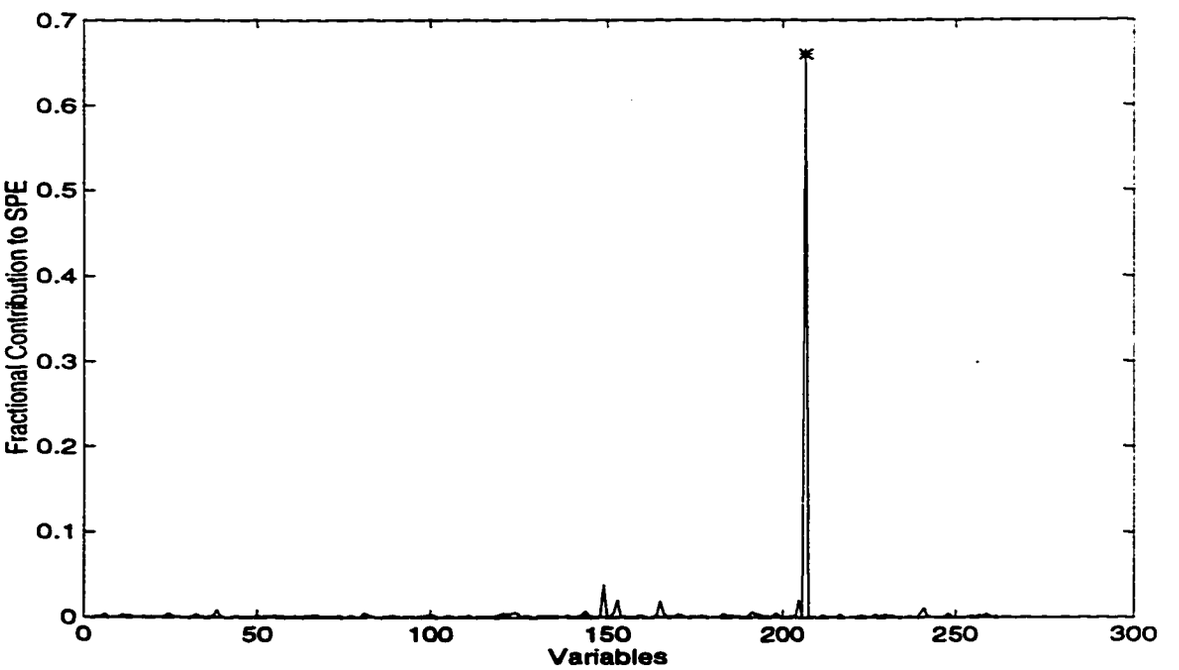

Figure 1.22: Contribution plot for a sample from Group 2 of online data : Styrene unit

timization framework. Two industrial case studies involving the algebraic PCA and PLS techniques are reported. In these application studies, the existing PCA and PLS analysis tools were used. The case study on the Mitsubishi distillation column illustrated the construction of a robust inferential model that can be implemented to perform automatic control of the column. This model was obtained by analyzing normal operating data from the column. A simple technique was presented to identify the key process variables for modelling purposes. In the case study involving the Shell styrene unit, the utility of principal components analysis in the monitoring of large scale units has been demonstrated. The importance of contribution plots for fault isolation was also highlighted.

The contribution of this thesis in extending the frontiers of these multivariate methods will become apparent when they are employed for multivariable dynamic model identification and control in the following chapters.

Chapter 2

Empirical Modelling with State Space Structures

2.1 Overview

This chapter deals with the review and description of some state space identification methods that are employed for the empirical identification of multivariable linear systems. In particular, attention is focussed on the Canonical Variate Analysis (CVA) technique of Larimore (1990) - the method is described in detail and its properties are enumerated. Using the case study of a simulated nonlinear continuous stirred tank heater (CSTR), the CVA technique is compared with the N4SID (Van Overschee and De Moor, 1994) algorithm available in the MATLAB Identification Toolbox. Identification and control of a laboratory stirred tank heater is also demonstrated. The CVA technique is also extended to the identification of systems represented by the Hammerstein structure (a nonlinear static block cascaded with a linear dynamic block). Identification of Hammerstein models is illustrated using data from a laboratory heat exchanger and the simulation example of a realistically complex acid-base neutralization tank.

---

[1] Sections of this chapter have been presented or published as

1. S. Lakshminarayanan, Sirish L. Shah and K. Nandakumar, "Identification of Hammerstein Models using Multivariate Statistical Tools", Chemical Engineering Science, 50 (22), 3599-3613 (1995).
2. S. Lakshminarayanan, Sirish L. Shah and K. Nandakumar, "Modeling a Class of Nonlinear MIMO Systems Using Multivariate Statistical Tools", Presented at the session on *Statistics and Quality Control* at the AIChE Annual Meeting, San Francisco, November 1994.
3. S. Lakshminarayanan, Sirish L. Shah and K. Nandakumar, "MIMO System Identification using Multivariate Techniques", Presented at the 44th Canadian Chemical Engineering Conference, Calgary, October 1994.

2.2 Contributions of this chapter

- Original Contributions

    1. The CVA technique has been extended to the identification of multivariable Hammerstein models. It may be worthwhile to point out the recent interest in the development of MIMO Hammerstein identification algorithms using the subspace methods. For example, Verhaegen and Westwick (1996a, 1996b) have extended the Multivariable Output-Error State Space (MOESP) algorithm (Verhaegen, 1994) to the identification of MIMO Hammerstein models.
    2. Some ideas from the AUDI algorithm of Niu and Fisher (1994) have been incorporated in the CVA approach in order to reduce the computational load.

- Applications

    1. The linear CVA technique is used to identify a model for the laboratory continuous stirred tank heater (CSTH). Control of the CSTH using the identified model is also presented.
    2. Data from an experimental heat exchanger is used to identify a Hammerstein model for the process.

- Other Contributions

    1. A tutorial introduction to the powerful CVA method that has only recently captured the attention of chemical engineers (Schaper *et al.* (1994), Lakshminarayanan *et al.* (1995) and Wang *et al.* (1996)) for process identification and monitoring.
    2. A review of the current literature concerning the identification of Hammerstein models is provided.
    3. Elucidation of some statistical properties of the N4SID (Numerical Algorithms for Subspace State Space System IDentification) and CVA algorithms via extensive simulations.

2.3 Introduction

Adequate characterization of the dynamical behavior of a plant over a wide range of operating conditions is mandatory for achieving tighter control of process variables. It has been widely acknowledged that good process identification is the most significant step towards reaping the benefits of advanced process control. Once an adequate dynamic model of the plant has been obtained, 80-90% of the implementation is complete. It is therefore clear that there exists a strong incentive to develop adequate process models.

There are three distinct approaches for process modelling :

- Phenomenological Modelling : This entails the development of models based on fundamental conservation laws - mass, momentum and energy balances. The result is usually a system of differential (ordinary or partial differential equations) and algebraic equations. Considerable engineering and laboratory knowledge goes into the construction of such models. Detailed first principles based models are often unavailable due to poor understanding of the complex physicochemical processes. Even if the model is available, it is too complicated for the synthesis of controllers.

- Empirical Modelling : Using exclusively experimental data (usually generated by an identification experiment), simple mathematical models are constructed via estimation of parameters for a specified model structure. Choice of a suitable model structure, generation of *rich* data that contains useful process information for the desired range of operation and a robust estimation procedure are necessary for the successful application of this approach. The mathematical model so constructed captures the input-output relationship - no fundamental process knowledge is represented in the model. Such a model may be adequate for the design of controllers: however, if changes occur in the process, the model has to be updated in order to reflect the current reality.

- Semi-Empirical Modelling : This method combines the advantages offered by the two identification methods listed above. Both a priori process knowledge and experimental data are used for identification, e.g. the poorly understood or complicated part of the first principles model is obtained from experimental data.

From a process control perspective, the empirical models are very appealing - the theory of controller synthesis has largely developed using this paradigm. Successful advanced industrial control schemes such as DMC as well as the commonplace PID controllers use some type of empirical model. For *time-invariant* plants, the empirical model can be identified off-line (often referred to as batch identification). In the case of *nonlinear* or *time-varying processes*, the identification is done under closed-loop conditions (employing a dither signal) so as to update the model parameters *on-line*. This chapter and other portions of this thesis deal with methods for *batch or off-line* identification of systems.

Empirical modelling of systems involves several steps (as shown in Figure 2.1), each of them crucial in its own way. Each of the blocks shown in the Figure is explained in some detail below.

1. Experimental Design

   This is the single most important step in the sequence of process identification. Without information rich data, it is not possible to obtain an adequate model of the process even with the best model structure and parameter estimation procedure. Experimental design is concerned with decisions regarding the duration of the identification experiment and the type and energy of the test signal. Special consideration needs to

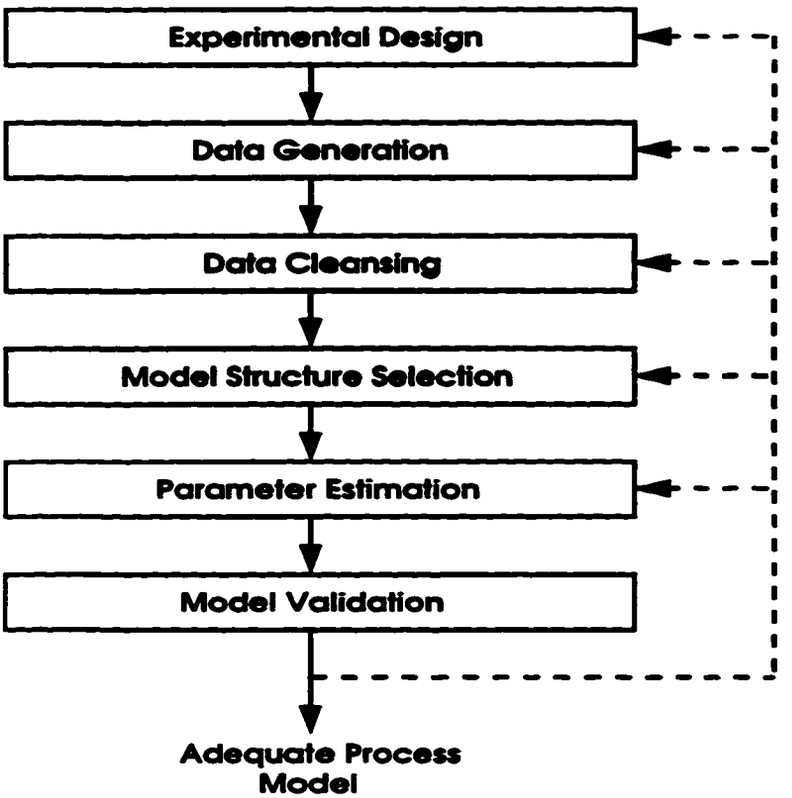

Figure 2.1: Flowchart for the model identification procedure : Empirical Modelling

be given for multivariable and nonlinear systems. The input signal must be *process friendly* and must also be acceptable to the plant management.

2. Data Generation

   This pertains to the actual identification experiment. The identification experiments are often time consuming owing to the slow nature of chemical processes. Special care and precautions should be taken to avoid undesired drifts and upsets.

3. Data Cleansing

   Data cleansing involves the treatment of data so as to serve two purposes : (1) removal of errors (outliers) which will cause errors in the estimated model and (2) enhance the frequency range in the model that is most relevant for control system design. Usually. high pass filtering (to remove trends) and low pass filtering with resampling (to avoid aliasing) must be done before the model can be identified.

4. Model Structure Selection

   A suitable model structure that can represent the collected plant data needs to be specified. The model is either in a parametric (ARX and time series models. state

space structure etc.) or non parametric (FIR/step response coefficients, frequency function etc.) form. If neural networks are used for process modelling, the number of input and output nodes, hidden layers and the shape of the activation functions need to be specified. Often, it includes a dynamical description of the process disturbances and noise. Some a priori process information can be used in fixing certain parts of the model. Specification of model structures for nonlinear processes is a particularly challenging task.

5. Parameter Estimation

   Using some mathematical/statistical procedure (often, this depends on the selected model structure), the unknown parameters of the model are obtained. Even when the true process does not belong to the assumed family of model structures, with an informative data set and a powerful statistical method, it is possible to obtain reasonable process models.

6. Model Validation

   The ultimate proof of the usefulness of the model is provided in this step. If undesired characteristics (e.g. lack of fit, improper noise structure) are seen here, it may be necessary to backtrack and rework from a previous step. In this step, the goodness of model fit is determined by cross validation, i.e. the model is used to predict using data which were not used in the estimation. The cross validation test must be favorable before the model can be used for controller design. Other tests include the examination of model residuals - if the residuals are correlated with the inputs or amongst themselves it is indicative of the inadequacy of the model structure or the incorrectness of the noise model.

From the above discussion it is clear that considerable skill and engineering judgment is required to identify mathematical models from process data. The task becomes even more difficult when empirical models are required for nonlinear or time varying process and daunting when one considers multivariable systems. Nevertheless, considerable insight has been obtained for the identification of nominal process models in each of the above classes.

Several identification methods for linear systems have emerged in the recent past. Schaper *et al.* (1994) review their advantages and disadvantages : (1) the least squares, recursive least squares (RLS) and autoregressive methods may give biased estimates; (2) the statistically accurate methods of autoregressive moving average (ARMA) modelling such as extended least squares (ELS), the Box-Jenkins approach (Box and Jenkins, 1976) and self-tuning regulator (Ljung and Soderstrom, 1983) are not always computationally or statistically well-conditioned. Gevers and Wertz (1982) have proved that the ARMA model structure is not globally identifiable and hence cannot be applied to high-order multivariable processes. Furthermore, these methods are plagued by problems such as accurate initialization and slow convergence. The Maximum Likelihood Estimation (MLE) procedure is

by far the most precise but requires an expert user. For routine and automated process identification, it is desirable to have an approach that :

- is applicable to a broad class of models without limits on the order
- is capable of identifying both linear and nonlinear process models
- automatically determines the optimum model structure and model orders but avoids overfitting the data
- is robust with respect to mild departures in the modelling assumptions such as linearity, time invariance and gaussian distribution and stationarity of the noise
- does not require an expert user to obtain models from process data

2.3.1 Organization of this Chapter

The powerful CVA technique due to Larimore and coworkers is described first. The method incorporates all of the features presented above. Besides, it has several useful mathematical and statistical properties which places it in a class of its own. The details of this algorithm will be described at length in the following sections. A brief introduction will also be provided to the N4SID algorithm. Both these algorithms identify models in the state space domain. The CVA and N4SID algorithms are evaluated using extensive simulations. The CVA technique of identification is also used to identify (a model) and control a laboratory stirred tank heater. The final portions of this chapter deals with the extension of the CVA algorithm to model a class of nonlinear systems.

2.4 The CVA Method

Given the input-output data from a process, the goal is to identify a linear model of the form,

$$X_{t+1} = \Phi X_t + G U_t + W_t \tag{2.1}$$

$$Y_t = H X_t + A U_t + B W_t + V_t \tag{2.2}$$

where X_t is the state, Y_t and U_t are the plant output and input sequences and W_t and V_t represent independent white noise processes with covariance matrices Q and R respectively. The presence of the $B W_t$ allows for a correlation between the state noise (W_t) and the measurement noise ($B W_t + V_t$) making the above structure fairly general and flexible. Larimore *et al.*, (1984) have shown that the use of the state space model structure where the measurement noise is correlated with the state noise results in a minimal order realization of the plant.

The linear identification strategy, that is being described here, borrows ideas from the canonical variate analysis technique of Larimore (1990) and the augmented upper diagonal identification (AUDI) algorithm developed recently by Niu and Fisher (1994). The CVA technique has been applied in the identification of chemical processes (Schaper *et al.* (1994)) and is a powerful identification method for linear systems. CVA is based on the Generalized Singular Value Decomposition (GSVD) theory. Several multivariate techniques such as canonical correlations analysis, canonical redundancy analysis, principal components analysis, partial least squares *etc.* can be conceptually and mathematically unified under the GSVD framework. For dynamic model identification, CVA uses the GSVD theory to obtain the pseudostates (no relation to the actual *physical* states) of the system. The pseudostates are the statistically significant optimal linear combinations of the past plant inputs and outputs with the first few pseudostates capturing most of the system dynamics. In the present study, CCA and PLS are employed as the multivariate techniques to generate the pseudostates, however only the CCA technique provides a near maximum likelihood system identification procedure (Larimore *et al.*, (1984)).

The AUDI family of algorithms mentioned earlier have been recommended to replace the traditional least squares based algorithms such as the recursive least squares (RLS). A careful rearrangement and augmentation of the data and parameter vectors normally used in the CVA and the conventional least squares based identification algorithms is found to result in a structure which provides the parameter estimates and loss functions of all orders from zero to a user specified maximum order. The L and U matrices determined by the lower and upper diagonal matrix (LU) factorization of a certain covariance matrix contains all the information on the parameters and loss functions and thus simultaneous order determination and parameter estimation is possible in a single computation step with almost no extra computational load compared with the RLS algorithm for the maximum order. This idea has been incorporated into the CCA based linear system identification procedure and significant reduction in computation time is achieved. Consequently, in this work, the data vectors are represented as in the AUDI algorithm rather than the way found in the conventional identification literature.

Consider a system for which the input-output data is available on 'p' manipulated inputs and 'q' controlled outputs. In the present approach, system identification involves the following steps :

1. Selection of optimal memory length. In simplistic terms, optimal memory length corresponds to the number of past values of plant inputs and outputs that capture the statistically significant behaviour in the future evolution of the process.

2. CCA/PLS based determination of the pseudostates of the process based on the optimal memory length.

3. Determination of the optimal number of pseudostates (optimal state order) to be retained in the final model.

4. Computation of the matrices in the state space model.

The Akaike Information Criterion (AIC) is used to handle the crucial issues of determining the optimal memory length and state order. Philosophically, we are concerned with the relationship between the complexity of a model and its performance on a given data set. The desired model is one whose *information distance* from the true system is a minimum (Ljung (1987)) with its complexity being as low as possible. From a practical viewpoint. AIC can be considered as a joint criterion for the determination of model structure and parameter values within the structure. Conceptually, the AIC can be regarded as follows :

$$AIC = \begin{pmatrix} Model\ fit \\ error \end{pmatrix} + \begin{pmatrix} Penalty\ for \\ model \\ complexity \end{pmatrix} \tag{2.3}$$

The penalty term is added to ensure that a model of increased complexity is chosen only when it offers a significantly better fit of the observed data. Thus, AIC represents a balance between the model fit and the number of parameters estimated. Beyond the true order of the model, there is no significant improvement in the model fit but the number of parameters keep increasing linearly with model order. This means that the plot of AIC versus model order will indicate a minima and then increase almost linearly. The AIC thus provides a definite and optimal procedure for the comparison of different models given a fixed set of observations.

2.4.1 Selection of optimal memory length

Based on the user specified value for the maximum memory length (MML). the 'past' of the process at any time t is defined as

$$\underline{p}(t) = \left[\underline{y}(t-MML)\,\underline{u}(t-MML)\cdots\ \underline{y}(t-2)\,\underline{u}(t-2)\ \underline{y}(t-1)\ \underline{u}(t-1)\right] \tag{2.4}$$

where

$$\underline{y}(t) = [y_1(t)\ y_2(t)\cdots\ y_q(t)] \tag{2.5}$$

and

$$\underline{u}(t) = [u_1(t)\ u_2(t)\cdots\ u_p(t)] \tag{2.6}$$

are the output and input sequences respectively.

An important point to note in the above definition of the 'past' vector is that the output and input variables alternate. This presents a contrast with the conventional identification literature where the inputs and outputs are blocked separately. The intertwining of the y's and the u's in the expression for the data vector is the fundamental reason for the superior performance of the AUDI algorithm in terms of obtaining much more information from the plant data (multiple model identification) than is possible with the conventional least squares estimator.

Now, construct the Augmented Covariance Matrix (ACM) as follows

$$ACM = \begin{bmatrix} \Sigma_{p(t)p(t)} & \Sigma^T_{y(t)p(t)} \\ \Sigma_{y(t)p(t)} & \Sigma_{y(t)y(t)} \end{bmatrix} \tag{2.7}$$

where

$$\Sigma_{p(t)p(t)} = \frac{1}{N - MML} \sum_{t=MML+1}^{N} \underline{p}^T(t)\underline{p}(t) \tag{2.8}$$

$$\Sigma_{y(t)p(t)} = \frac{1}{N - MML} \sum_{t=MML+1}^{N} \underline{y}^T(t)\underline{p}(t) \tag{2.9}$$

$$\Sigma_{y(t)y(t)} = \frac{1}{N - MML} \sum_{t=MML+1}^{N} \underline{y}^T(t)\underline{y}(t) \tag{2.10}$$

are the sample covariance matrices.

The loss functions for all subsystems 1 through q and for all memory lengths from 0 through MML are found from the diagonal elements of the U matrix obtained from the LU factorization of the ACM. Niu and Fisher (1994) use a similar strategy for model order determination in their AUDI family of algorithms. Interpretation of the structure of the U matrix can be found in Niu (1994).

AIC for all memory lengths 'k' (k = 1. ... .MML) is computed from the loss functions as follows

$$AIC_k = [(N - MML)\ln(\det(U_k))] + 2kq(p+q) \tag{2.11}$$

where

- U_k corresponds to the diagonal loss function matrix for memory length 'k'. The diagonal elements of U_k are the loss functions for subsystems 1 through q.
- The second term in the right hand side of the AIC expression. is the number of parameters estimated for a k^{th} order ARX model.

The optimal memory length (OML) is the value of 'k' for which AIC is a minimum. If the optimal memory length is close to MML, it is necessary to repeat the computations using a larger value for MML and then arrive at the optimal value for the OML of the system.

2.4.2 Determination of Pseudostates

Once OML is fixed, the 'past' and 'future' of the process at time t is defined as

$$\underline{p}(t) = \left[\underline{y}(t-OML)\,\underline{u}(t-OML)\ \cdots\ \underline{y}(t-2)\,\underline{u}(t-2)\,\underline{y}(t-1)\,\underline{u}(t-1)\right] \tag{2.12}$$

$$\underline{f}(t) = \left[\underline{y}(t)\,\underline{y}(t+1)\ \cdots\ \underline{y}(t+OML-1)\right] \tag{2.13}$$

from which the sample covariance matrices $\Sigma_{p(t)p(t)}$, $\Sigma_{p(t)f(t)}$ and $\Sigma_{f(t)f(t)}$ are determined.

The pseudostates are then computed as linear combinations of the past space using canonical correlations analysis or partial least squares. To relate back to the CCA and PLS overview presented earlier, we have the past (P) and future (F) spaces instead of the X and Y spaces respectively. If the CCA technique is use, a total of OML*q pseudostates are generated. With the PLS technique, a total of OML*(p+q) pseudostates are obtained. If J is the matrix of weight vectors then the pseudostates are defined as X_t=$\underline{p}(t)$J.

2.4.3 State Order Selection

Having generated all the possible pseudostates of the system based on the input-output data, the next decision to be made concerns the number of these states that need to be retained in the final model. The AIC is again used for this purpose. For a system with 'p' manipulated inputs, 'q' outputs and plant order 'k' we can express AIC as

$$AIC_k = (N - 2 \times OML + 1)\left[q(1 + \ln 2\pi) + \ln \mid \Sigma^k_{\epsilon\epsilon} \mid\right] + \delta_k M_k \tag{2.14}$$

where N is the number of data points, OML the optimal memory length, δ_k the small sample correction factor (defined below) as proposed by Hurvich and Tsai (1990) and M_k is twice the number of free parameters for the k-order state space model (see Candy *et al.* (1979)). The expressions for δ_k and M_k are given below.

$$\delta_k = \frac{N}{N - \left(\frac{M_k}{q} + \frac{q+1}{2}\right)} \tag{2.15}$$

$$M_k = 4kq + q(q+1) + 2kp + 2qp \tag{2.16}$$

The error covariance matrix for plant order 'k' is given by

$$\Sigma^k_{\epsilon\epsilon} = \frac{1}{N - 2 \times OML + 1} \sum_{t=OML+1}^{N-OML+1} \left(\underline{y}(t) - \underline{\hat{y}}^k(t)\right)^T \left(\underline{y}(t) - \underline{\hat{y}}^k(t)\right) \tag{2.17}$$

The difference between the actual plant output measurement vector $\underline{y}(t)$ and the one step ahead prediction vector (assuming plant order k) $\underline{\hat{y}}^k(t)$ is used to determine the error covariance matrix. The one step ahead prediction is expressed as

$$\underline{\hat{y}}^k(t) = \Sigma_{y(t)p(t)} J_k \left(J_k^T \Sigma_{p(t)p(t)} J_k\right)^{-1} J_k^T \underline{p}^T(t) \tag{2.18}$$

J_k is a matrix comprising of the first 'k' weight vectors of the past space as columns. AIC is computed for all values of 'k' from 0 to the maximum possible state order (maximum possible state order equals OML*q for CCA and OML*(p+q) for PLS). The value of 'k' that gives a minimum for AIC is selected as the optimal plant order.

2.4.4 Estimation of the State Space Model

Having identified the optimal pseudostates of the plant, we are now in a position to estimate the various matrices in the state space model equations (2.1) and (2.2). The states of this model are the first 'k' (optimal) pseudostates that have been computed. The state space model matrices, computed using multiple linear regression, can be obtained using equations (2.19) through (2.26).

$$\begin{bmatrix} \Phi & G \\ H & A \end{bmatrix} = \begin{bmatrix} J_k^T \Sigma_{p(t+1)p(t)} J_k & J_k^T \Sigma_{p(t+1)u(t)} \\ \Sigma_{y(t)p(t)} J_k & \Sigma_{y(t)u(t)} \end{bmatrix} \Delta^{-1} \tag{2.19}$$

where

$$\Delta = \begin{bmatrix} J_k^T \Sigma_{p(t)p(t)} J_k & J_k^T \Sigma_{p(t)u(t)} \\ \Sigma_{u(t)p(t)} J_k & \Sigma_{y(t)u(t)} \end{bmatrix} \tag{2.20}$$

The B. Q and R matrices are given by

$$B = S_{21} S_{11}^{\dagger} \tag{2.21}$$

$$Q = S_{11} \tag{2.22}$$

$$R = S_{22} - S_{21} S_{11}^{\dagger} S_{12} \tag{2.23}$$

Here, † indicates the pseudoinverse operation. The submatrices, S_{11} through S_{22}. are obtained from the covariance matrix of the prediction error, S.

$$S = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} = \begin{bmatrix} J_k^T \Sigma_{p(t+1)p(t+1)} J_k & J_k^T \Sigma_{p(t+1)y(t)} \\ \Sigma_{y(t)p(t+1)} J_k & \Sigma_{y(t)y(t)} \end{bmatrix} - \Psi \tag{2.24}$$

with

$$\Psi = \begin{bmatrix} \Phi & G \\ H & A \end{bmatrix} \begin{bmatrix} J_k^T \Sigma_{p(t)p(t+1)} J_k & J_k^T \Sigma_{p(t)y(t)} \\ \Sigma_{u(t)p(t+1)} J_k & \Sigma_{u(t)y(t)} \end{bmatrix} \tag{2.25}$$

The covariance matrices in the above expressions are obtained in the same way as shown in equations (2.8) through (2.10). Definitions for $\underline{y}(t)$, $\underline{u}(t)$ and $\underline{p}(t)$ are found in equations (2.5), (2.6) and (2.12) respectively. $\underline{p}(t+1)$ is defined as follows :

$$\underline{p}(t+1) = \left[\underline{y}(t+1-OML)\, \underline{u}(t+1-OML) \cdots \underline{y}(t-1)\, \underline{u}(t-1)\, \underline{y}(t)\, \underline{u}(t) \right] \tag{2.26}$$

2.5 Overview of the N4SID algorithm

While the CVA algorithm presented above relies heavily on statistical arguments, the MOESP and N4SID algorithms are based on geometrical and linear algebra concepts. However, as shown in Van Overschee and De Moor (1995), it is possible to consider these algo-

rithms as special cases of a general unifying theory - the difference being in the choice of the weighting matrices used. The exposition of the unifying theory is beyond the scope of this thesis - the user is referred to the cited paper for further details.

The N4SID algorithm consist of two steps. The first step involves the computation of a certain characteristic subspace, from the process input-output data, that coincides with the subspace generated by the columns of the extended observability matrix of the system. The dimension of this subspace is equal to the order of the system to be identified. In the second step, the extended observability matrix and the model order computed above is used to identify the system matrices (such as Φ, G, H and A).

In N4SID, an optimal linear combination of the past and future inputs is used to predict the future outputs. The space generated by the optimal linear combinations of the past inputs is called the oblique projection, $\mathcal{O}$. The singular value decomposition of the oblique projection can be expressed as :

$$\mathcal{O} = (\mathcal{U}_1 \quad \mathcal{U}_2) \begin{pmatrix} \mathcal{S}_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \mathcal{V}_1^T \\ \mathcal{V}_2^T \end{pmatrix} \tag{2.27}$$

The order of the model is equal to the number of non-zero singular values in $\mathcal{S}_1$. The extended observability matrix is $\mathcal{U}_1 \ (\mathcal{S}_1)^{1/2}$. The states $\tilde{X}_i$ are determined from the above SVD matrices as $\tilde{X}_i = (\mathcal{S}_1)^{1/2} \ \mathcal{V}_1^T$. Once the extended observability matrix and the states have been computed, the system matrices are obtained by solving a system of equations (as in CVA) using the least squares technique.

2.6 Evaluation of the CVA and N4SID algorithms

The statistical and optimality properties of the CVA and other subspace algorithms have been elucidated in the identification literature. A brief review follows :

- Larimore (1994) illustrated the optimality of CVA in quite small samples of 100 data points while estimating the 43 parameters of a 6-state, 2-input, 2-output process when the state order is known. When the state order is estimated using AIC corrected for small sample size, the CVA achieves optimality provided the condition of persistent excitation holds true. The rich input excitation facilitates reliable model order selection and hence the optimality.

- Deistler *et al.* (1995) investigated the statistical properties of two algorithms - CVA versus a subspace algorithm proposed by Akaike (1976). Using simulation studies, they conclude that the CVA estimates are as asymptotically efficient as the Maximum Likelihood (ML) estimates. This implies that CVA combines the numerical simplicity of the subspace methods and the statistical optimality of the ML or the PEM (prediction error method) estimator. Note that the PEM estimator is asymptotically equivalent to the ML estimator.

- Van Overschee and De Moor (1995) provide a common framework for the three popular subspace identification algorithms - CVA, MOESP and N4SID. It is interesting to note that all the subspace identification algorithms calculate the same result (up to within a similarity transform) for very large sample sizes *when the correct state order is chosen.* When lower model orders are selected, CVA was shown to have the smallest variance for the eigenvalue estimates of the state matrix Φ. Further, while N4SID and MOESP are sensitive to scaling of the inputs and/or outputs, the CVA method is not sensitive.

The statistical optimality (estimation of model order, variance of the estimated pole locations etc. for simulation examples) and the numerical stability of the CVA algorithm have been observed and reported in several studies such as the ones mentioned above. Though no formal proof has been provided to substantiate these claims, no single counter-example is available as of now.

In this study, using the simulation example of a typical process system - the nonisothermal CSTR, the CVA approach with the states computed using (1) CCA (2) PLS and the N4SID algorithm from the MATLAB Toolbox were evaluated.

Making the usual assumptions, Seborg *et al.* (1989) derive the following model of the reactor where a simple, irreversible, first order exothermic reaction $A \rightarrow B$ takes place.

$$\dot{C}_A = \frac{Q_f}{V}(C_{Af} - C_A) - k_o \exp\left(-\frac{E}{RT}\right) C_A$$

$$\dot{T} = \frac{Q_f}{V}(T_f - T) + \frac{(-\Delta H)}{\rho C_p} k_o \exp\left(-\frac{E}{RT}\right) C_A + \frac{UA_h}{V\rho C_p}(T_c - T)$$

The outputs are the concentration (C_A) and temperature (T) of the reactor contents and the manipulated variables are the feed flow rate (Q_f) and the temperature of the coolant (T_c). More details can be found in Schaper *et al.* (1994) or Seborg *et al.* (1989). The steady state values of the output, for the parameter values specified in Table 2.1, is given by $\bar{C}_A = 0.077$ (lb mol/ft^3) and $\bar{T} = 571.35\ ^oR$. For this set of operating conditions, the linear description of the CSTR results in a second order model with poles located at $0.8524 \pm 0.0868i$.

The nonlinear CSTR model was perturbed simultaneously by random binary sequences of magnitude 10 ft^3/hr in the feed flow rate and 10 R in the coolant temperature. Using a sampling frequency of 20 h^{-1}, 500 data sets each consisting of 1500 samples were simulated with signal to noise ratios of 1, 10 and 100. From these, data sets comprising of the first 300, 600, 900, 1200 and 1500 points were constructed and used for identification. 500 data sets each containing 5000 data points were also generated. Consequently, each row in the following tables represent results based on 500 identification runs. The identified order is the median value of the 500 runs. Some of the identification runs returned unstable models and MSPE values for such cases are denoted by an asterisk (*).

Table 2.1: Nominal operating values of the CSTR

Parameter	*Value*
k_o	$2 \times 10^8 h^{-1}$
$\frac{E}{R}$	$1 \times 10^4\ ^oR$
T_f	$530\ ^oR$
T_c	$530\ ^oR$
C_{Af}	$0.27\ lbmol/ft^3$
-ΔH	$1.5 \times 10^4 BTU/lbmol$
UA_h	2000 BTU $/h^oR$
Q_f	$100\ ft^3/h$
V	$50\ ft^3$
ρC_p	$50\ BTU/ft^3\ ^oR$

Table 2.2: Results of the CVA (CCA) identification

Sample Size	SNR	Identified Model Order
300	1	2
300	10	2
300	100	2
600	1	2
600	10	2
600	100	2
900	1	2
900	10	2
900	100	2
1200	1	3
1200	10	2
1200	100	2
1500	1	2
1500	10	2
1500	100	2
5000	1	2
5000	100	2

Table 2.3: Results of the CVA (PLS) identification

Sample Size	SNR	Identified Model Order
300	1	9
300	10	27
300	100	26
600	1	8
600	10	30
600	100	31
900	1	8
900	10	34
900	100	43
1200	1	5
1200	10	37
1200	100	41
1500	1	3
1500	10	47
1500	100	33
5000	1	2
5000	100	40

Table 2.4: Results of the N4SID identification

Sample Size	SNR	Identified Model Order
300	1	15 (*)
300	10	15 (*)
300	100	15 (*)
600	1	15 (*)
600	10	15 (*)
600	100	9 (*)
900	1	15 (*)
900	10	15 (*)
900	100	10 (*)
1200	1	15 (*)
1200	10	15 (*)
1200	100	10 (*)
1500	1	15 (*)
1500	10	15
1500	100	8
5000	1	15 (*)
5000	100	5

Some conclusions can be arrived at by examining tables 2.2 through 2.4. They are summarized below :

- The CVA algorithm that used the CCA technique to construct the *states* identified the exact model order in all cases except one. This included cases with short data records and significant noise levels.
- If the PLS technique is used to generate the *states*, the resulting model is found to be considerably overparameterized (more states are found necessary to fit the plant data). The results obtained using PLS did not appear to conform to any set pattern.
- For large data sets with high noise levels, the CCA and PLS based CVA identified the true plant order.
- All models that were identified (various sample sizes, signal-to-noise ratios) using the CCA/PLS based CVA were stable (all poles were inside the unit circle).
- The N4SID algorithm almost always suggested model orders that were significantly different from the true plant order. When the SNR was small, the correct order was not identified even with large samples. Higher noise levels seem to cause problems for this identification procedure. The method appears to be more suited to analyzing large data sets with high SNR.
- The most disturbing aspect of the performance of the N4SID algorithm was that it identified unstable models even for this stable plant. Reliability of the models identified by this algorithm is therefore suspect.

More details of the identification procedures are now presented. A data set containing 1500 samples with a signal to noise ratio of 10 was simulated using the nonlinear CSTR model. The three techniques were used to identify models from this data set.

- CVA (CCA) Model : The plot of the AIC measure against the state order is presented in Figure 2.2. The sharp minima in the plot (the sharp minima is typical in the identification using CVA) suggests a second order model with pole locations at $0.8506 \pm 0.0675i$. The model fit is shown as a scatter plot in Figure 2.3. The observed and predicted values fall on the 45^o line indicating a good fit.

  The goodness of fit is also supported by the whiteness test performed on the residuals of both output variables (see Figure 2.4) with the autocorrelation function (ACF) and the partial autocorrelation function (PACF) plots following the theoretical patterns observed for white noise.

  The cross validation run (shown in Figure 2.5) performed using data from a different *identification experiment* indicates that the model captures the dynamics of the plant very well.

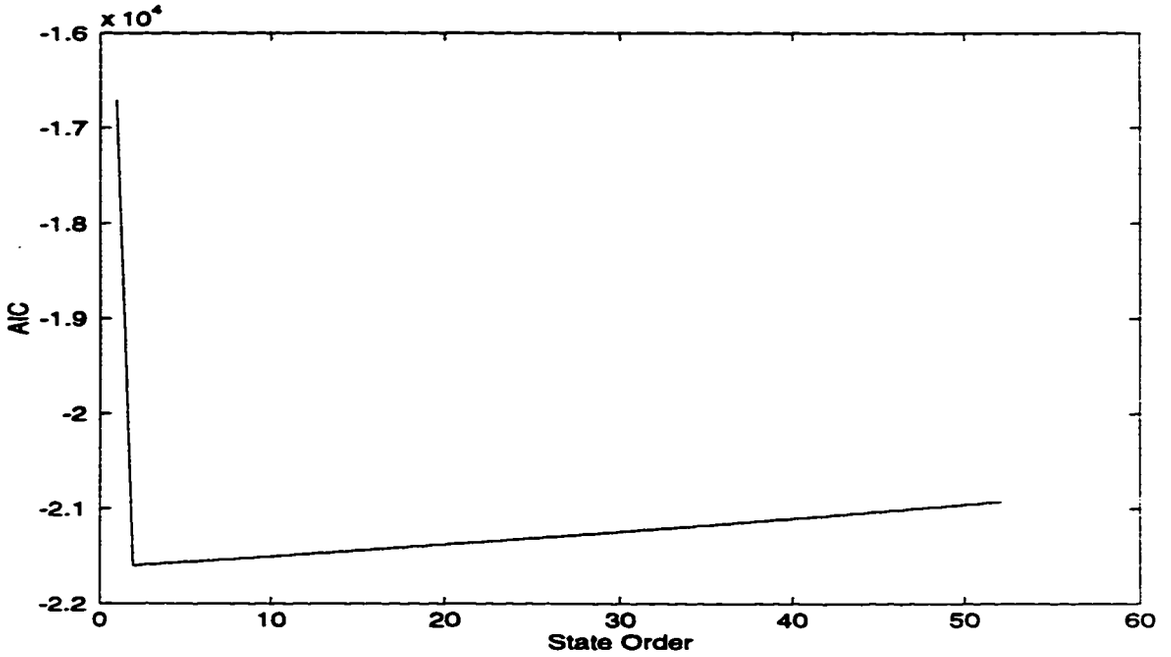

Figure 2.2: The AIC for selecting optimal model order

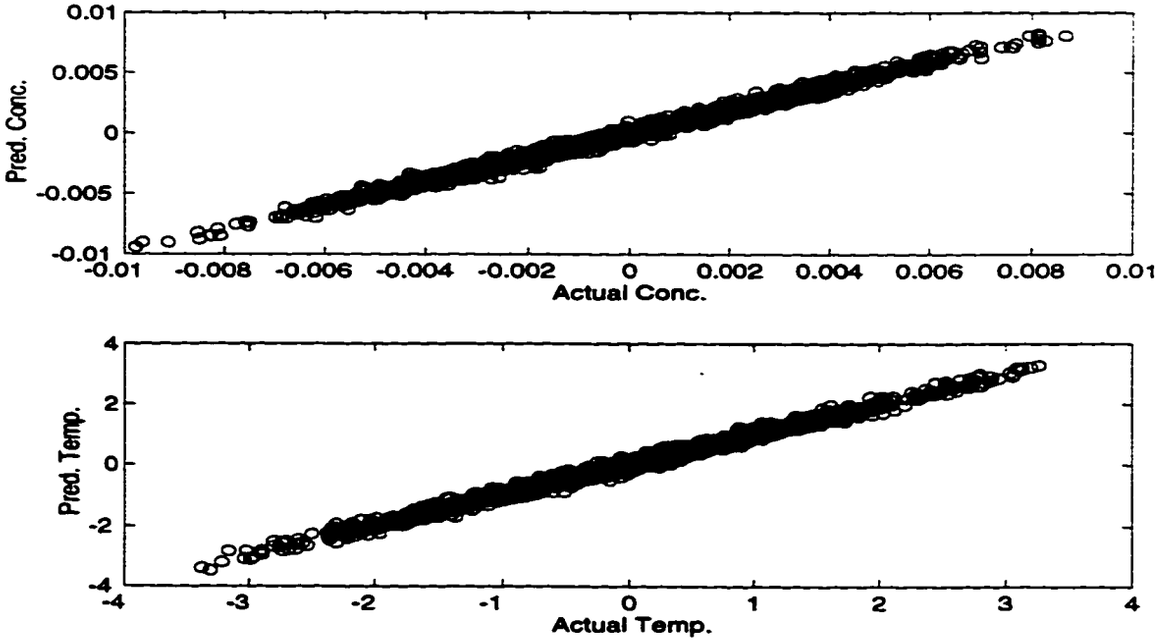

Figure 2.3: Scatter plot depicting the model fit : Second order CVA (CCA) model

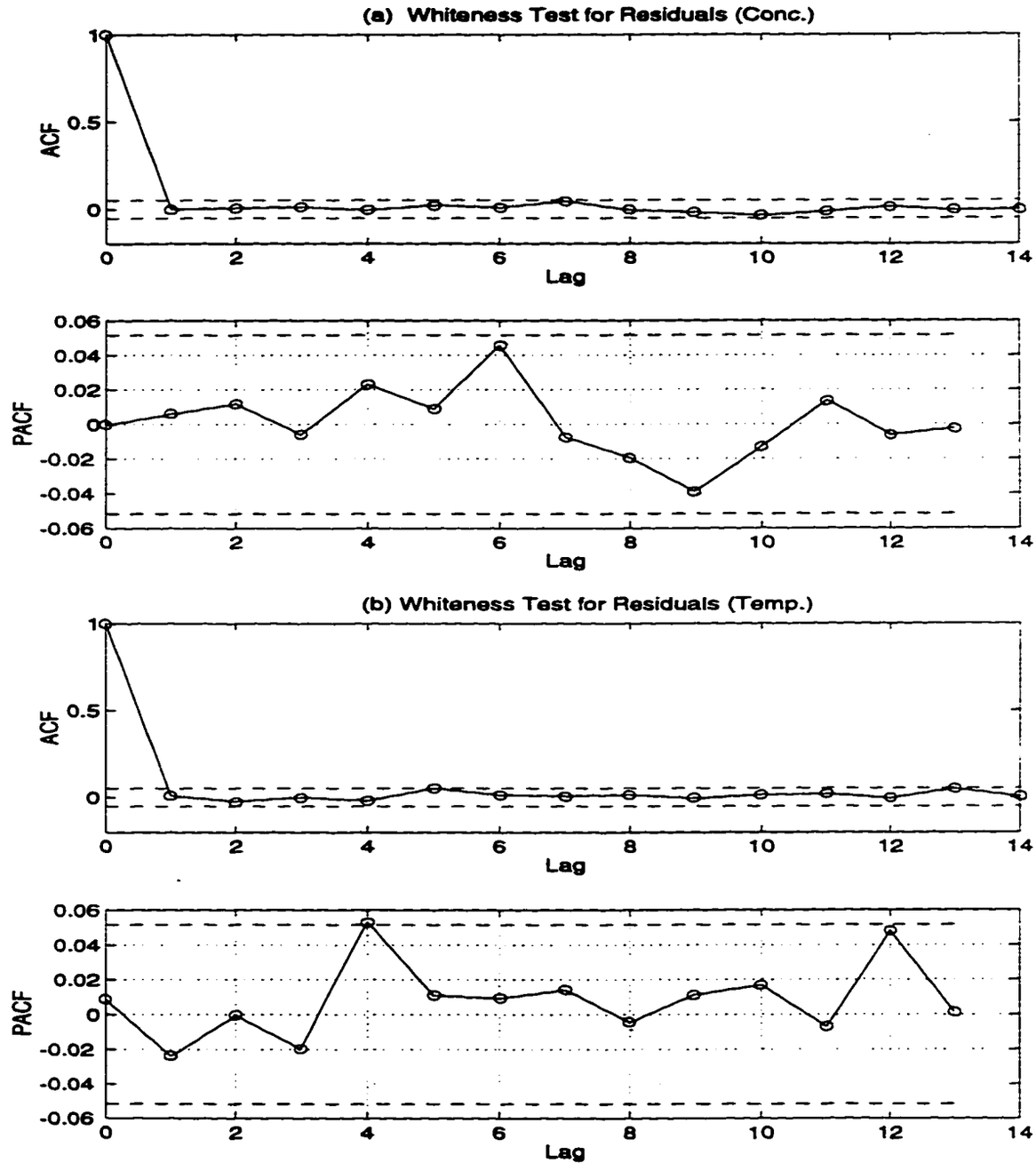

Figure 2.4: The ACF and PACF plots for the CVA (CCA) model : Test for whiteness in residuals

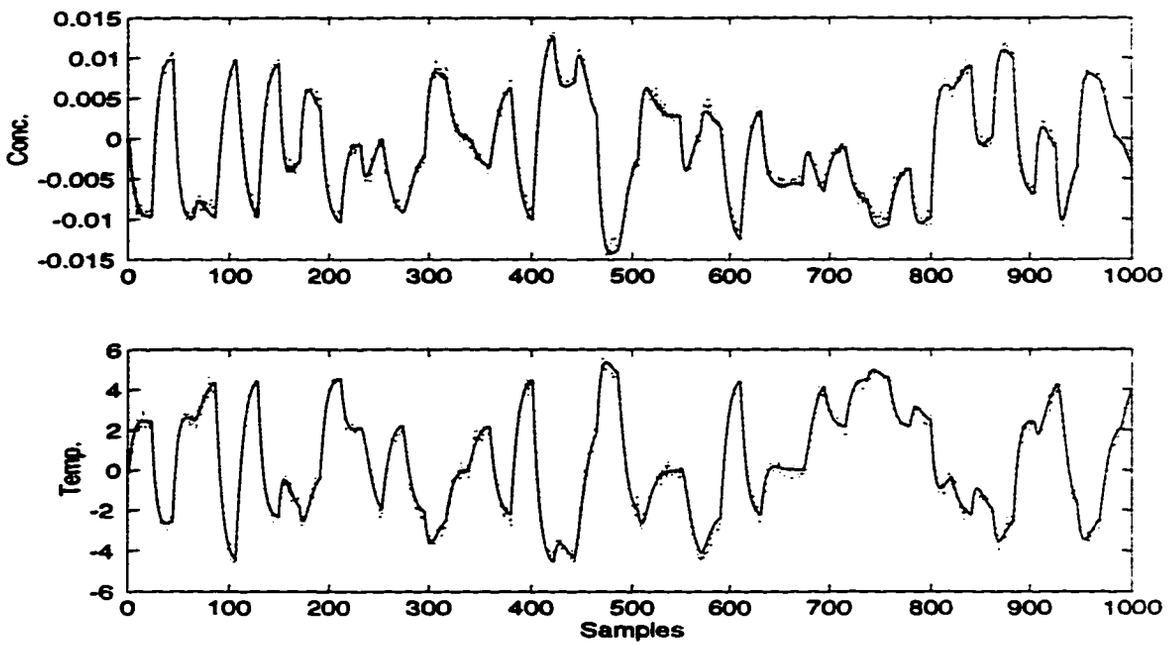

Figure 2.5: Cross validation run : Second order CVA (CCA) model. The dots represent the actual process outputs and the solid lines indicate the model predictions

- CVA (PLS) Model : A stable model of order 20 was suggested by the identification procedure based on the AIC. The mean squared prediction errors (MSPE) were similar to that obtained with the CVA (CCA) method. Furthermore, the residuals of this model *passed* the whiteness test. When the model order was restricted to 2, the MSPE values were significantly larger (see Table 2.5). The pole locations of this second order model were $0.8632 \pm 0.0894i$. The scatter plot depicting the model fit (Figure 2.6) shows a larger spread around the 45^o line compared to the results of the CVA (CCA) method. The residuals had noticeable structure in them (refer Figure 2.7).

  The cross validation results shown in Figure 2.8 also indicates the inadequacy of this second order model.

  The results seem to indicate that the *states* generated by the PLS technique are not as *strong* as those generated by CCA. To obtain comparable MSPE values and whiteness in residuals, more states are required by the PLS based CVA procedure.

- N4SID Model : This method suggested a model order of 15 - the MSPE values were closer to those obtained with the CVA (CCA) technique. Residual analysis indicated white residuals for this model. If the model order is fixed at 2 owing to theoretical considerations, the model fit was as good as those obtained with a 15th order model. The residuals were also white. This seems to indicate that the 13 remaining states were fitting either the noise or nothing at all.

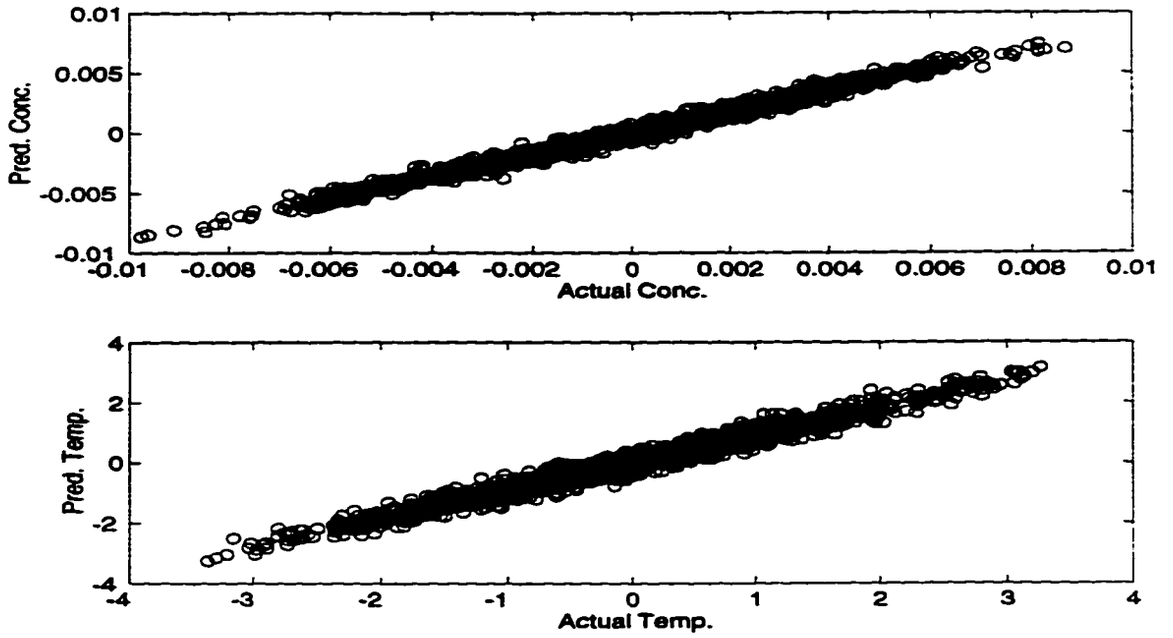

Figure 2.6: Scatter plot depicting the model fit : Second order CVA (PLS) model

Table 2.5: MSPE values for the identification and cross validation runs

Method	Identification Data Set		Cross Validation Data Set	
	$MSPE_{y_1}$	$MSPE_{y_2}$	$MSPE_{y_1}$	$MSPE_{y_2}$
CVA (CCA)	1.04×10^{-7}	0.0149	4.82×10^{-7}	0.07
N4SID	1.04×10^{-7}	0.0148	4.80×10^{-7}	0.07
CVA (PLS)	2.47×10^{-7}	0.0522	1.47×10^{-6}	0.40

The model fit, whiteness test for the residuals and cross validation are presented in Figures 2.9. 2.10 and 2.11 respectively. The pole locations were $0.8526 \pm 0.0699i$.

The MSPE values obtained for the three second order models for both the identification and cross validation data sets are summarized in Table 2.5. It is seen that the MSPE values for the CVA (CCA) and the N4SID procedure are almost identical (with N4SID results being marginally (though not significantly better). The lower order CVA (PLS) model appears to be poor in comparison. From this it can be suggested that the model order selection criterion used in the N4SID procedure has to be improved and made robust with respect to noise levels and sample sizes. Selection of the model order close to the true plant order even under unfriendly circumstances (low SNR, small sample sizes etc.) is the distinguishing characteristic of the CVA (CCA) procedure and makes it suitable as a fully automated system identification procedure.

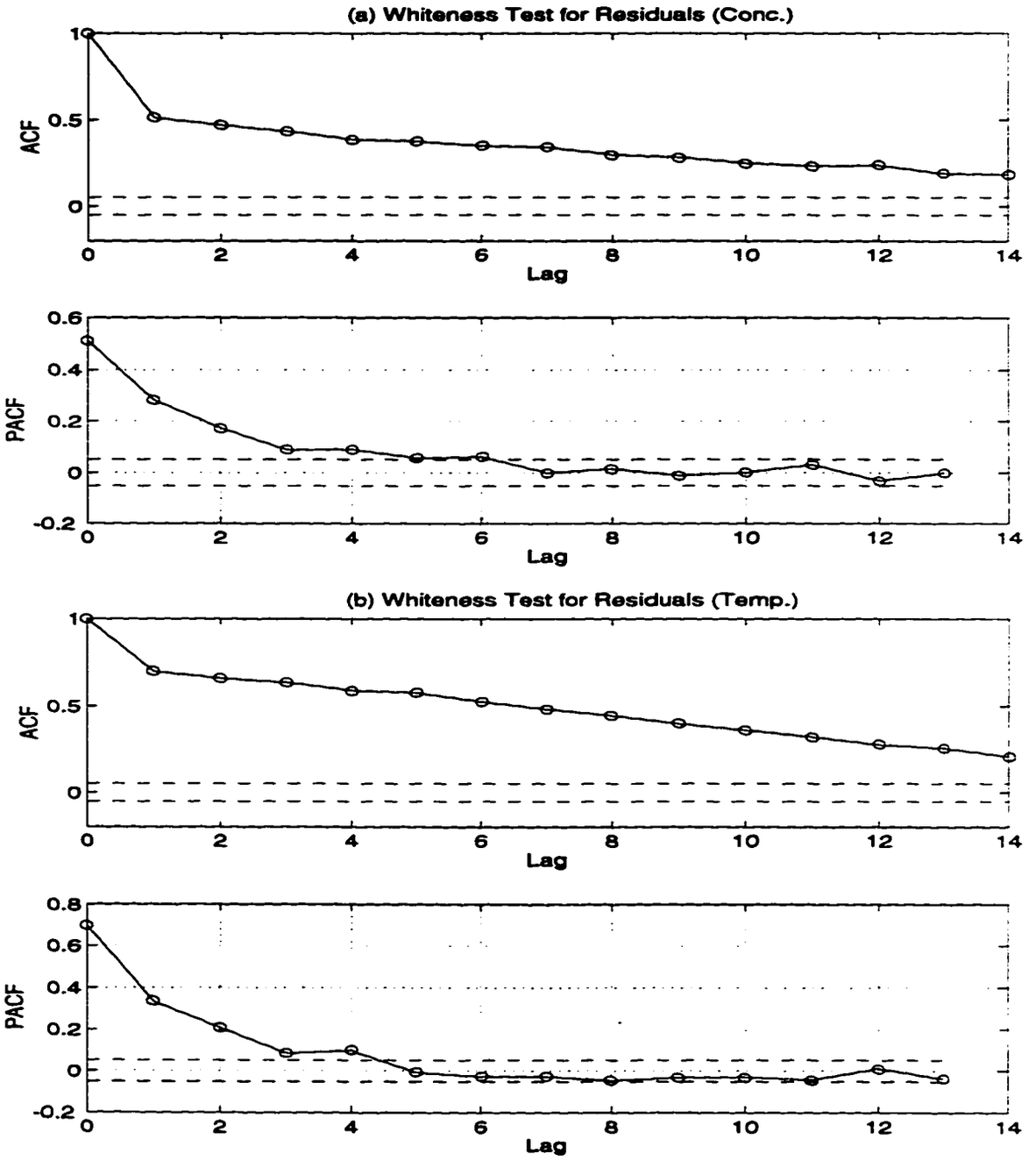

Figure 2.7: The ACF and PACF plots for the CVA (PLS) model : Test for whiteness in residuals

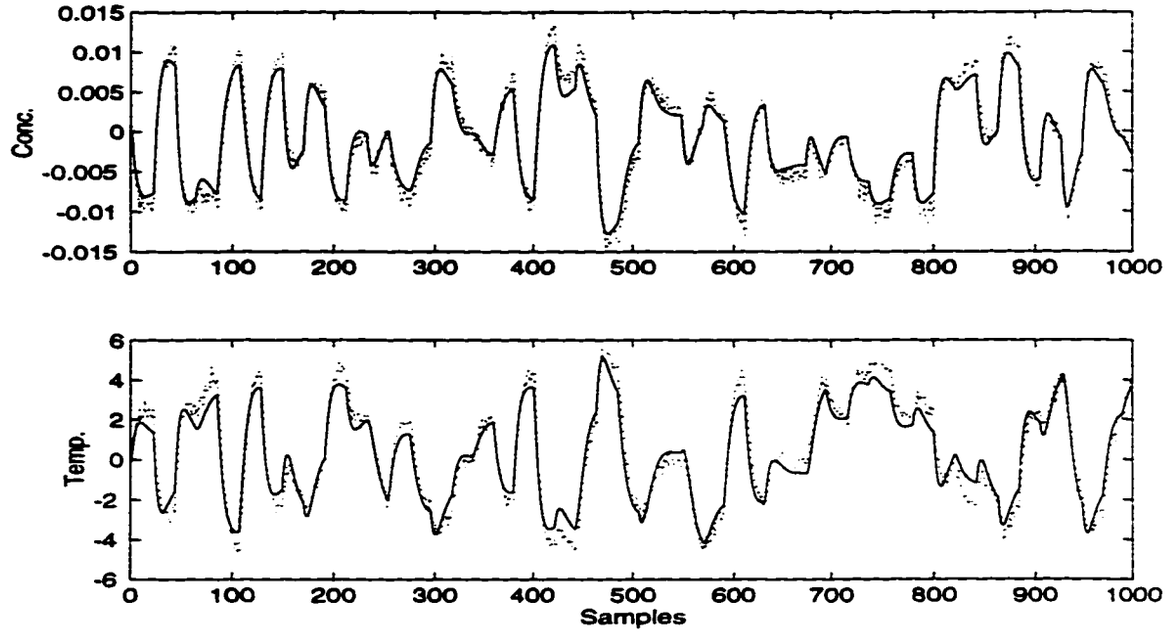

Figure 2.8: Cross validation run : Second order CVA (PLS) model. The dots represent the actual process outputs and the solid lines indicate the model predictions

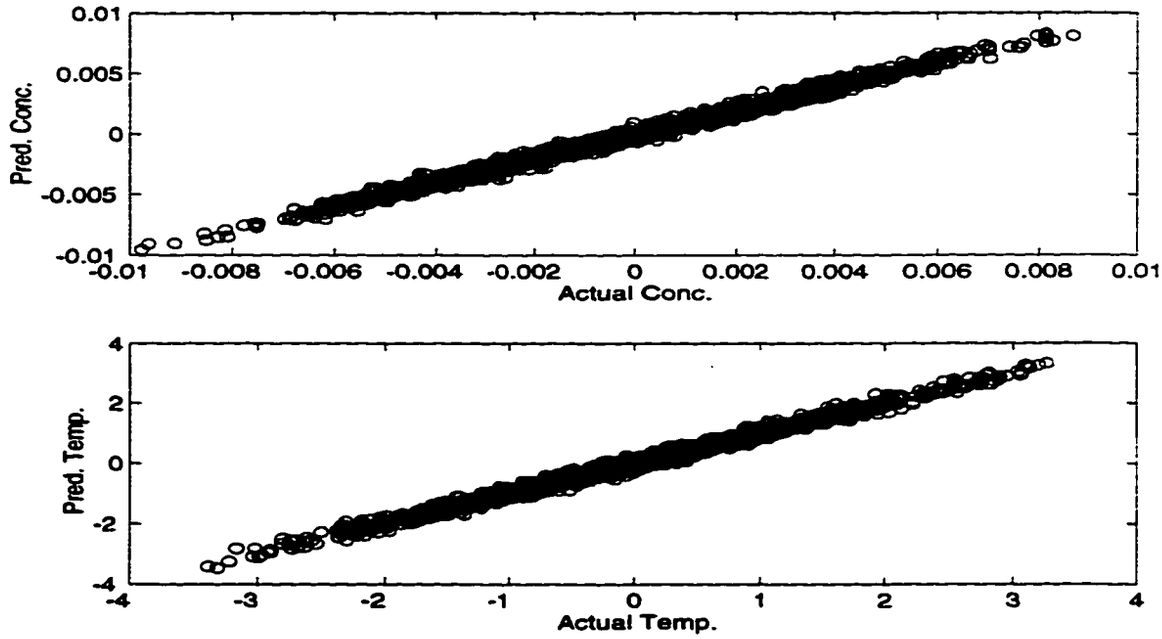

Figure 2.9: Scatter plot depicting the model fit : Second order N4SID model

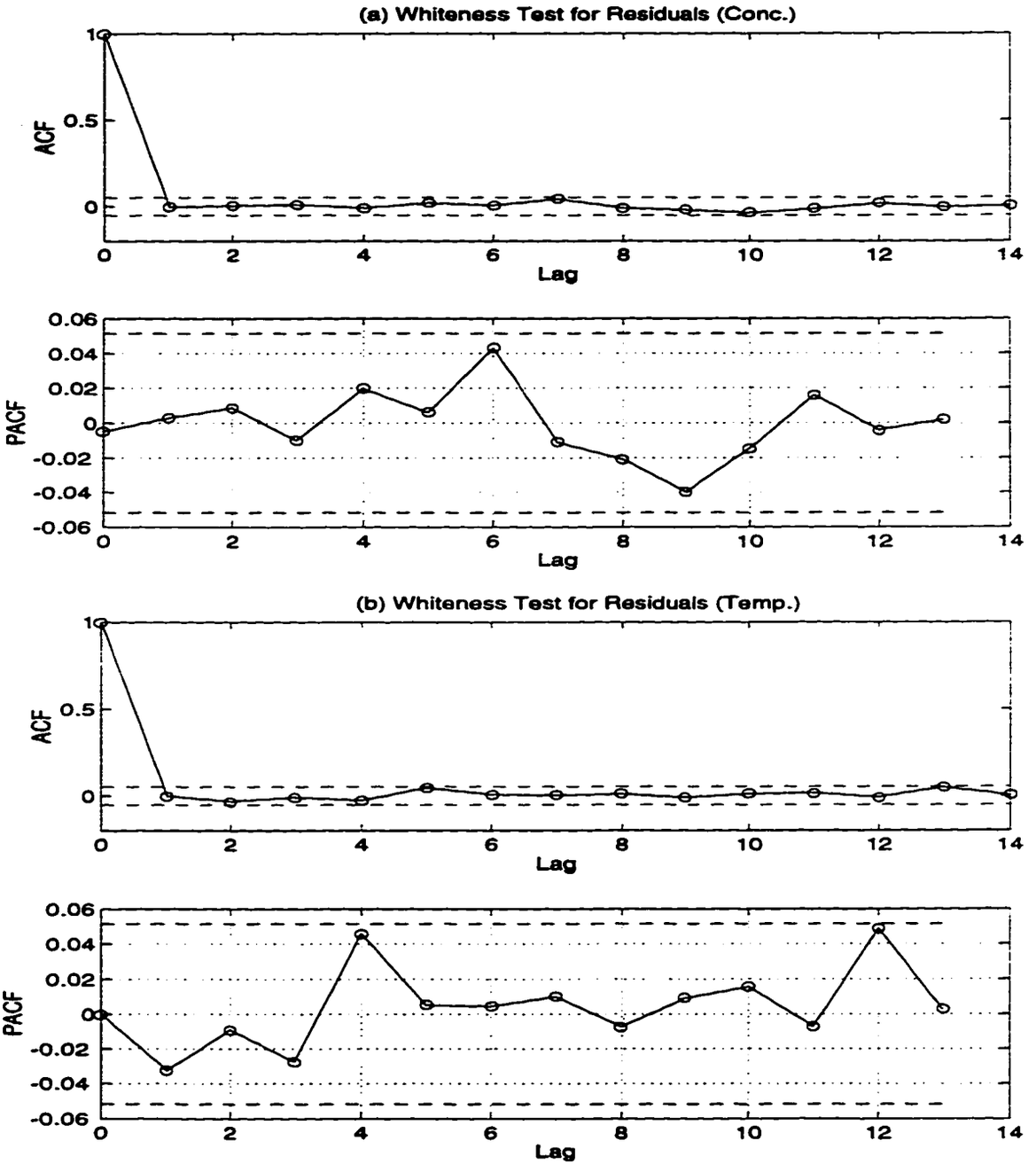

Figure 2.10: The ACF and PACF plots for the N4SID model : Test for whiteness in residuals

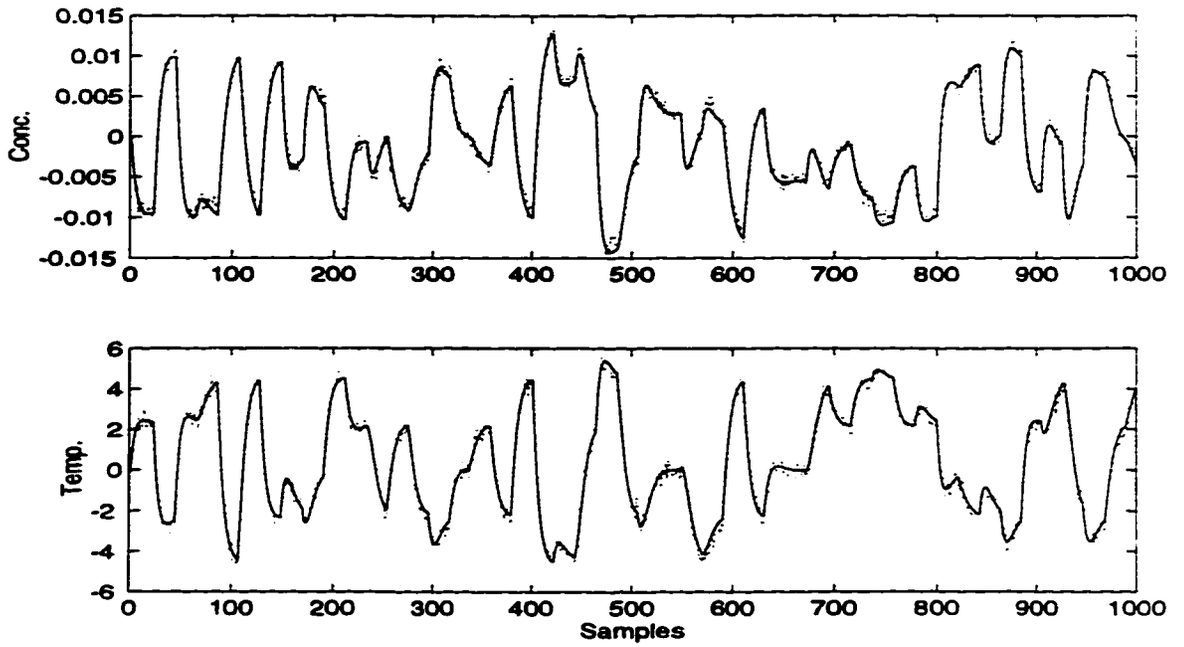

Figure 2.11: Cross validation run : Second order N4SID model. The dots represent the actual process outputs and the solid lines indicate the model predictions

2.7 Identification and Control of the Laboratory CSTH

The laboratory CSTH system (see Figure 2.12) is a cylindrical tank of uniform cross section where two streams of water (one hot and the other cold) are mixed. The contents of the tank exit through a long and winding copper tube. The flow rates of the hot (u_1) and cold water (u_2) streams serve as manipulated variables to control the temperature of the exit stream (y_1) as well as the level of water in the tank (y_2). A thermocouple was located at about 4.75 m downstream (from the tank exit) to provide data on this variable. Facilities exist to introduce disturbances in the steam flow through the heater coil (not shown in Figure 2.12), the inlet temperature of the hot water stream, etc. The data acquisition and control algorithms were implemented using a personal computer (PC486/33MHz/1.2GB HDD) running real-time MATLAB/SIMULINK.

The manipulated inputs were perturbed by a sequence of step type signals of varying amplitude in order to collect plant data. A data set consisting of 715 samples were collected at intervals of five seconds. Linear trends in the data were removed by detrending. The detrended data were then processed by the CVA and N4SID algorithms.

The CVA algorithm identified a fifth-order state space model. The model fit is depicted as a time series plot in Figure 2.13 and as a scatter plot in Figure 2.14. The mean squared errors (MSE) for the model fit were 0.1046 and 1.4156 for the temperature and the level channels respectively. The correlation coefficients between the actual measurements and the model predictions were 0.9919 (for temperature) and 0.9648 (for level). Both these

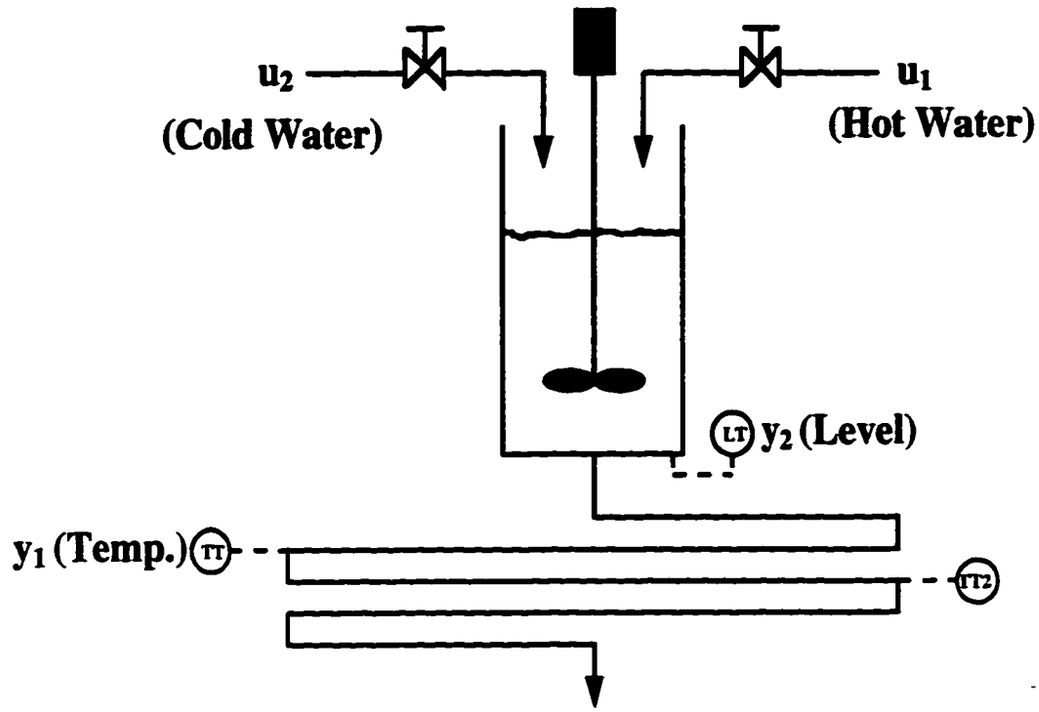

Figure 2.12: Schematic of the Laboratory CSTH

plots indicate that the temperature is modelled better than the level. This may be due to : (1) slow dynamics and (2) relatively higher noise levels for this channel. The step responses shown in Figure 2.15 indicate that the hot and cold water flow rates have a very similar effect on the water level (as expected). The steady state effects of the manipulated inputs on the exit stream temperature are different due the operating and the environmental (temperature of the hot and cold streams) conditions.

A state space model with state order 3 was identified by the MATLAB-N4SID algorithm. The model fit, scatter plot of the actual and predicted values as well as the step responses identified by the model are shown in Figures 2.16, 2.17 and 2.18. The MSE values are 0.1370 and 1.7291 for temperature and level respectively. The correlations between the actual and predicted values were 0.9893 and 0.9537 respectively. The model fit is marginally inferior to that provided by the CVA approach. Again, the level is poorly modelled compared to the temperature. The step responses for the level are again very similar. The hot and cold water flow rates appear to have qualitatively and quantitatively different effects on the temperature (in contrast to the results from the CVA model).

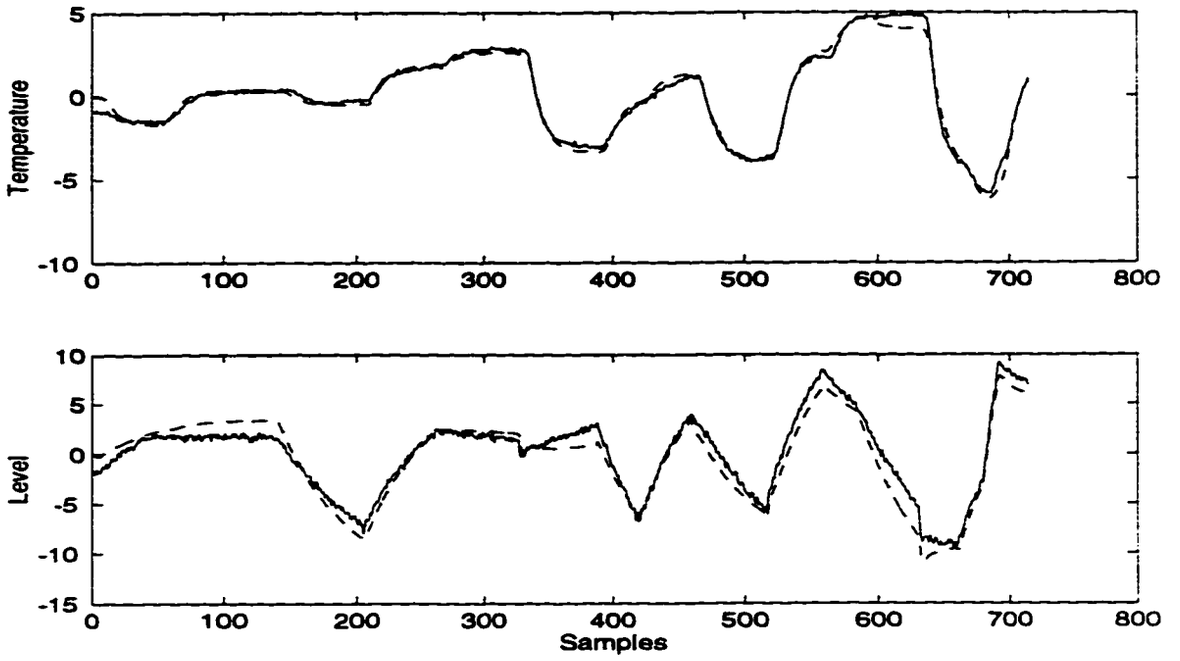

Figure 2.13: Model fit for the CSTH data using the CVA (CCA) approach. The solid lines represent the actual measurements and the dashed lines indicate the model predictions

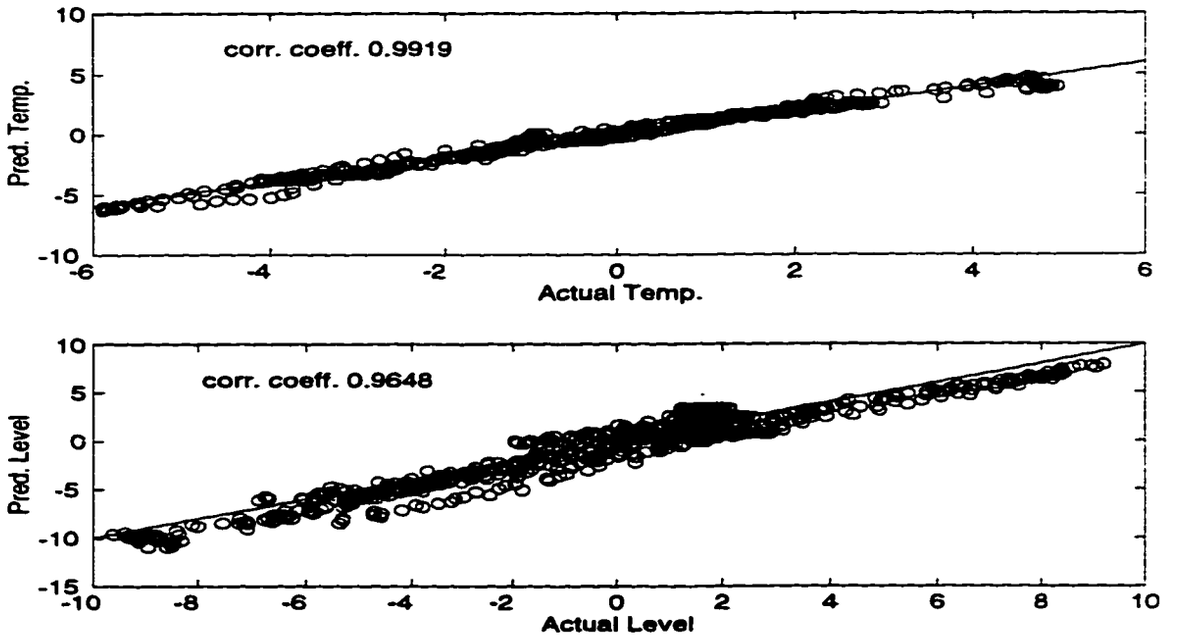

Figure 2.14: Scatter plot showing the model fit for the CSTH data using the CVA (CCA) technique

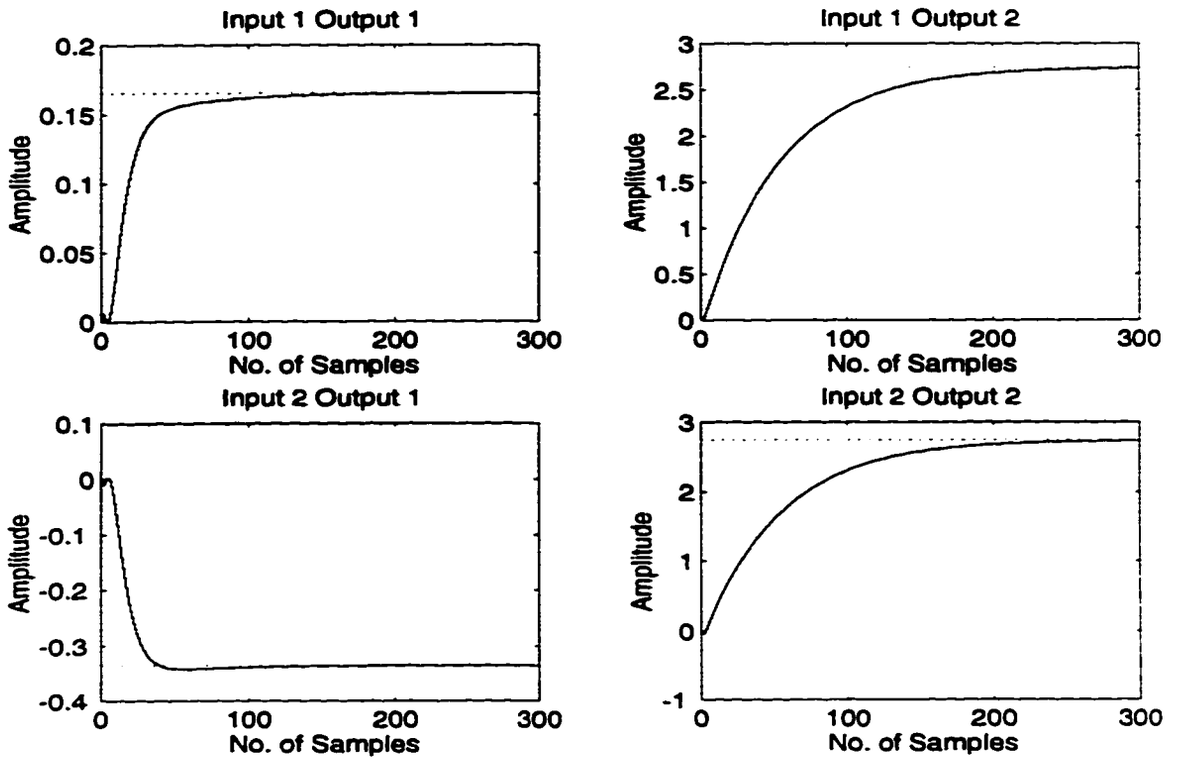

Figure 2.15: Step responses using the CVA (CCA) model

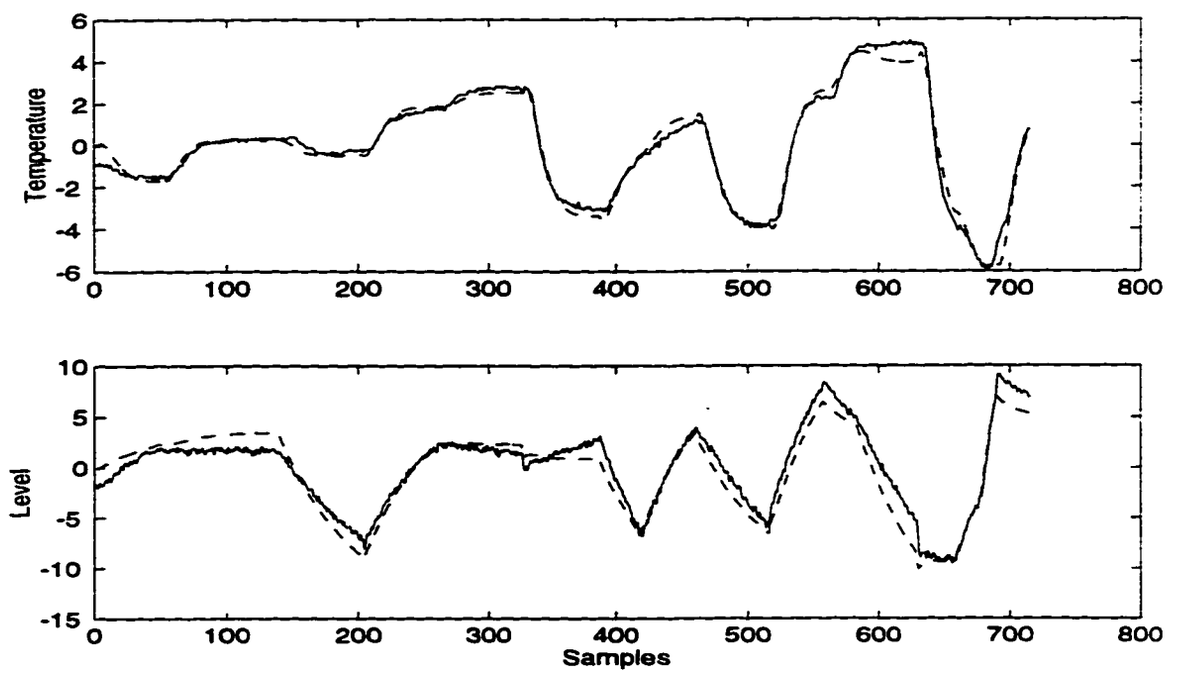

Figure 2.16: Model fit for the CSTH data using the N4SID approach. The solid lines represent the actual measurements and the dashed lines indicate the model predictions

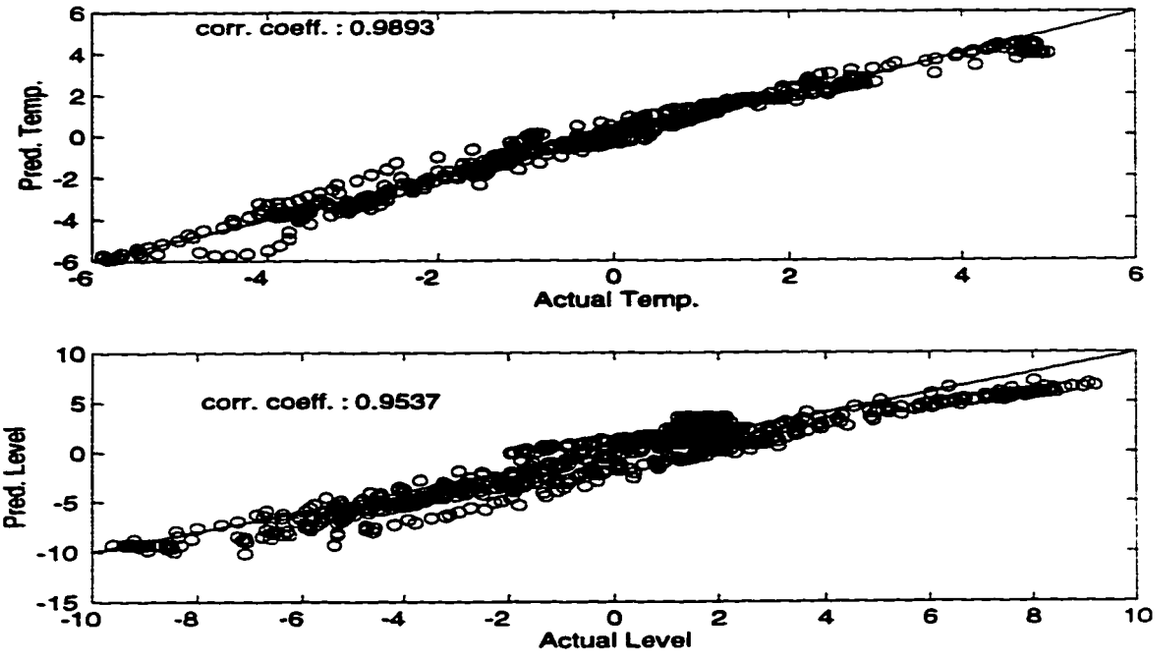

Figure 2.17: Scatter plot showing the model fit for the CSTH data using N4SID

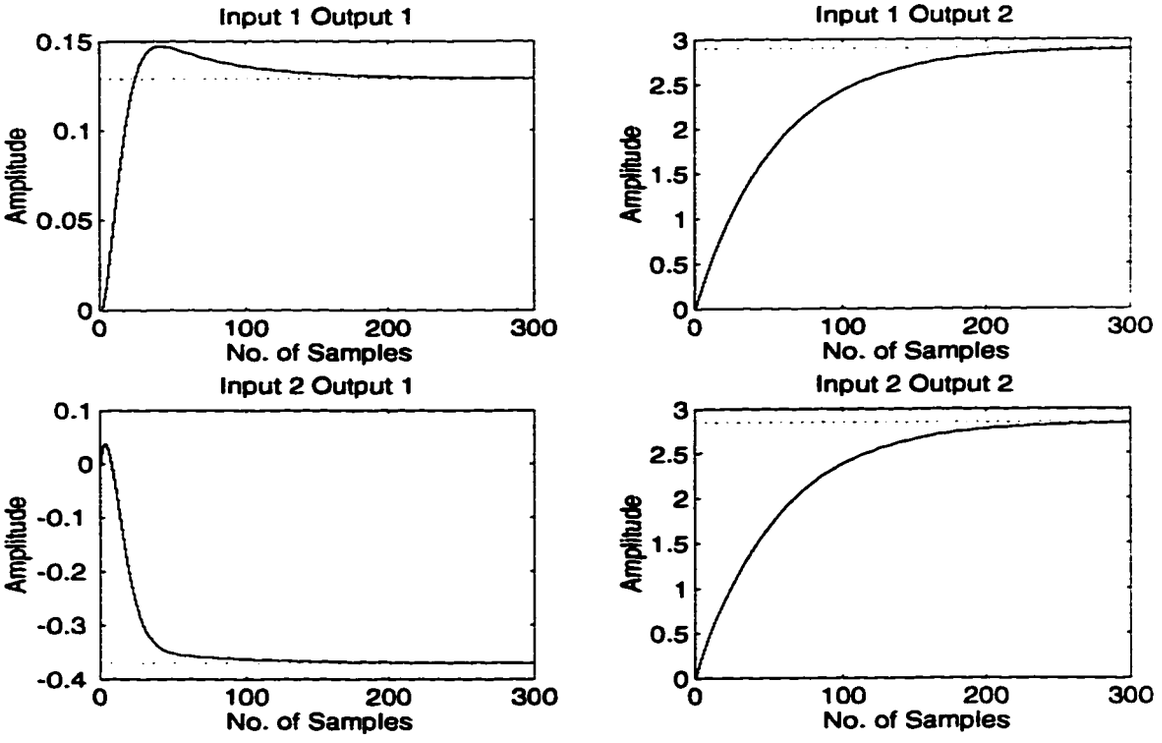

Figure 2.18: Step responses using the N4SID model

Analysis of residuals from the CVA and N4SID models indicate a certain structure in the residuals. By suitably filtering the plant data (the filter can be designed based on the autocorrelation and partial autocorrelation plots) a more accurate model could be obtained. The ultimate goal of this modelling exercise was to implement a multivariable constrained model based predictive controller on the process. Therefore, the fit obtained from the models were considered adequate. The CVA model (system matrices and the noise covariance matrices) is :

$$\Phi = \begin{bmatrix} 0.9805 & -0.0103 & -0.0223 & 0.0077 & -0.0234 \\ 0.0260 & 0.9835 & -0.0201 & -0.0215 & 0 \\ 0.3001 & -0.0779 & 0.7642 & 0.0154 & -0.0394 \\ 0.5770 & 0.1574 & -0.3620 & 0.3438 & -0.0210 \\ -1.1402 & 0.5951 & 1.2442 & -0.2715 & 0.5102 \end{bmatrix} \tag{2.28}$$

$$G = \begin{bmatrix} -0.0012 & 0.0035 \\ 0.0090 & 0.0049 \\ 0.0101 & -0.0541 \\ -0.1437 & -0.2374 \\ -0.1510 & 0.1666 \end{bmatrix} \tag{2.29}$$

$$H = \begin{bmatrix} -2.5563 & -0.0522 & -0.0087 & -0.0052 & -0.0135 \\ -0.7528 & 3.7872 & 0.1220 & 0.2128 & 0.0074 \end{bmatrix} \tag{2.30}$$

$$A = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{2.31}$$

$$B = \begin{bmatrix} -3.0419 & -0.3327 & 0.5141 & -0.0060 & 0.1458 \\ -0.5742 & 7.1318 & 0.9544 & -0.0197 & -0.2303 \end{bmatrix} \tag{2.32}$$

$$Q = \begin{bmatrix} 0.0006 & -0.0000 & 0.0000 & 0.0007 & -0.0002 \\ -0.0000 & 0.0007 & 0.0004 & -0.0012 & -0.0007 \\ 0.0000 & 0.0004 & 0.0030 & 0.0001 & 0.0010 \\ 0.0007 & -0.0012 & 0.0001 & 0.0299 & -0.0031 \\ -0.0002 & -0.0007 & 0.0010 & -0.0031 & 0.0461 \end{bmatrix} \tag{2.33}$$

$$R = \begin{bmatrix} 0.0001 & 0.0006 \\ -0.0008 & 0.0237 \end{bmatrix} \tag{2.34}$$

A constrained DMC controller (with amplitude and rate constraints on the inputs) was implemented using the real-time MATLAB / SIMULINK platform. The CVA model was used to provide the step response coefficients for use in the DMC algorithm (details of the DMC control algorithm will appear in Chapter 4). The DMC controller parameters were : N_1=1, N_2=10, N_u=2 and λ=1. Constraints were posed on the rate (0 - 100%)

and the amplitudes (± 10%) of the input signal. The results of the control experiment is presented in Figure 2.19. Both servo as well as regulatory runs are shown. The initial 250 samples represent the startup period. This is followed by two setpoint changes (one positive and the other negative) in level. The setpoint changes are tracked perfectly and with minimal effect on temperature. Between samples 700 and 900 two setpoint changes were made on the temperature. The new setpoints are tracked well albeit less aggressively compared to the level channel. Some interaction is noticed for the large setpoint change made in level around sample time 1100. At sample time 1400, a step disturbance was introduced in the hot water stream (by mixing it with another cold stream) - this large disturbance affected the temperature loop more than the level loop (level loop appears to be very tightly tuned). The disturbance was quickly rejected by increasing the hot water flow and decreasing the cold water flow. Now, simultaneous setpoint changes were made in the level and temperature. The time delay for the temperature channel was increased by measuring the exit temperature at a distance downstream (TT2) from the usual location. The temperature setpoint is reached after a significant time delay - oscillations are seen in the water level too. However, the control system is able to cope up with this increase in time delay without becoming unstable. The temperature measurement was reverted back to the original location and a setpoint change was made in the temperature at the 1780^{th} sampling instant. This setpoint could not be reached even though the cold water flow was completely shut off. This is because the temperature setpoint was higher than the temperature of the hot inlet stream. A steady state offset was noticed in the temperature but the level was maintained very close to its setpoint. When the temperature of the hot inlet flow was restored to the original value and the setpoint of the level channel was increased at sample 2000, both the setpoints were reached. Finally, an unmeasured amount of cold water was quickly dumped into the tank to simulate a pulse type disturbance resulting in a decrease in the temperature and an increase in the level. These effects were quickly removed and the system was restored to the desired state.

2.8 Identification of Hammerstein Models

Most chemical processes are nonlinear. Capturing the process nonlinearities using empirical models is a crucial yet challenging task in nonlinear system identification. Many model structures have been proposed for the identification of nonlinear systems. Significant among these are the classical Volterra series expansion models, block oriented models (Hammerstein and Wiener structures), polynomial ARMA models (NARMAX), state-affine representations and neural networks. A recent survey of the models available for nonlinear input-output modelling can be found in Cinar (1994) .

Hammerstein models provide the simplest and useful representations of typical chemical engineering processes like high purity distillation columns, heat exchangers (Eskinat *et al.* (1991), Luyben and Eskinat (1994)) and pH systems (Zhu and Seborg (1994)). Well

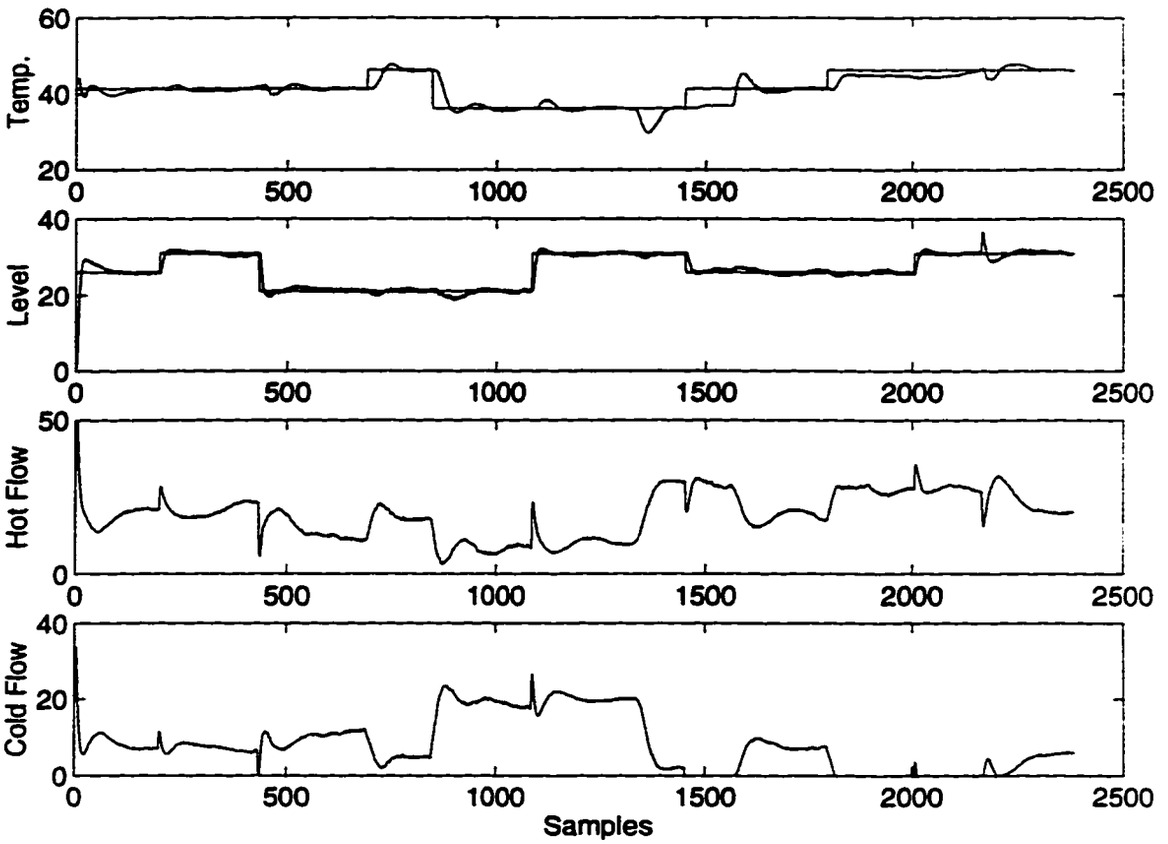

Figure 2.19: Constrained DMC implementation on the Laboratory CSTH using the CVA model

established linear controller design methods can be employed once the Hammerstein model of the system becomes available. Zhu and Seborg (1994) present the nonlinear predictive control (unconstrained) of a neutralization system using a Hammerstein representation of the process. The extension of the CVA technique for the identification of multivariable nonlinear systems is the goal of this section.

The Hammerstein model (see Figure 2.20) consists of a nonlinear static element followed by a linear dynamic element. The pioneering work of Narendra and Gallman (1966) provided the initial impetus to this type of modelling. Their iterative algorithm extended the technique presented for linear systems by Steiglitz and McBride (1965). The Narendra-Gallman algorithm (NGA) updates the linear dynamic element and the nonlinear gain polynomial separately and sequentially. This prompted a flurry of research activity in this area and techniques that considered the Hammerstein models as linear MISO models were put forth. Hsia (1968) used a noniterative strategy to estimate the Hammerstein model parameters for the case where the linear dynamic part has no zeros. This restriction was relaxed in the work of Chang and Luus (1971) who claimed that their algorithm was superior to the NGA in terms of computation time. Gallman (1976) proved that, on the contrary, the

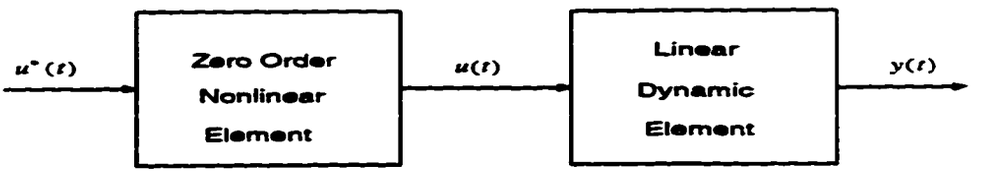

Figure 2.20: Basic Hammerstein Model

NGA provides accurate parameter estimates and is actually faster for higher order systems. The above methods worked well for systems with additive white noise. Hammerstein model identification methods that handle coloured noise were presented by Haist *et al.* (1973) and Hsia (1976). Nonparametric models based on correlation analysis was used by Billings and Fakhouri (1979). Online identification of MISO Hammerstein models based on the well known recursive least squares (RLS) algorithm and the recursive prediction error method (RPEM) was examined by Kortmann and Unbehauen (1987). They conclude that the RPEM algorithm provides better results compared to the RLS algorithm. Eskinat *et al.* (1991) established the robustness of NGA, to various levels of noise. in comparison to other well known identification methods such as the prediction error method (PEM) and the recursive prediction error method (RPEM). Extension of the NGA to include MISO systems was also done in their work. A remarkable feature of this work is the use of Hammerstein models to represent the dynamics of real physical systems such as the distillation column and heat exchanger. In a recent paper, Lang Zi-Qiang (1994) describes a new method based on the results of nonparametric statistics and best approximation theory. However. this method is applicable only to plants whose linear dynamics are open loop stable. Luyben and Eskinat (1994) outline an experimental procedure for the determination of the SISO Hammerstein model using nonlinear auto-tuning.

A major drawback with all the existing parametric methods is that the order of the dynamic part is assumed to be known *a priori* even for SISO systems. On the other hand. the nonparametric methods are not useful in the case of asymptotically unstable plants. The compromise solution under such a scenario would be to use a linear identification method that can provide reliable model order and parameter estimates - the CVA technique emerges as the natural choice. In this work, the iterative NGA technique for the identification of a nonlinear system representable by the Hammerstein structure is extended to perform simultaneous structure determination and parameter estimation of multivariable chemical process systems. The parameters of the linear system obtained in state space form using the CVA method and the coefficients of the polynomial type nonlinear elements are alternately adjusted, until convergence, to obtain the model (see Lakshminarayanan *et al.* (1995)).

First, some possible parameterizations of the Hammerstein model are illustrated. This is followed by a description of the NGA along with the mathematical formulation of the Hammerstein model identification procedure. Finally, application of the developed theory to model typical chemical process systems is presented followed by concluding remarks.

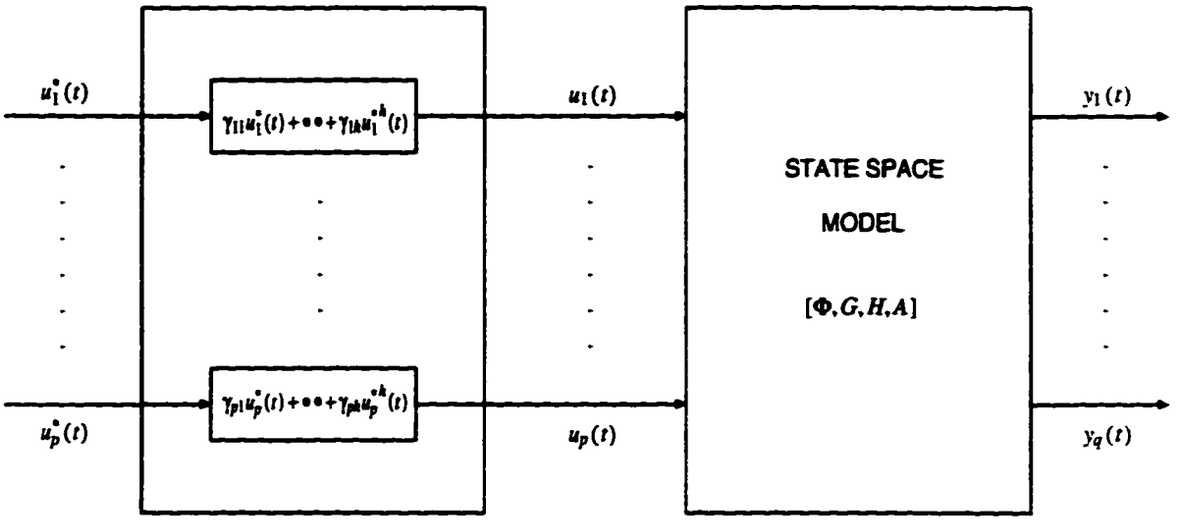

Figure 2.21: Separate Parameterization

2.8.1 Hammerstein Model Parameterization

Parameterization of the Hammerstein structure for the case of single input-single output (SISO) is unique and straightforward. However, structures for the MIMO Hammerstein model are reasonably complex. This is due to the fact that several model structures are possible depending on the way the static nonlinearities are realized. Two such parameterization schemes are illustrated in Figures 2.21 and 2.22. Separate parameterization involves transforming each of the plant inputs individually to get the inputs to the linear part of the model. However, the separate parameterization is somewhat restrictive and often not found to be adequate for modelling systems such as those considered in this communication. The combined parameterization involves a complex transformation where the inputs to the linear part of the model are obtained by considering the powers and products of the plant inputs as will be seen in the pH example considered later. As a consequence, relatively more parameters are to be estimated in the latter parameterization.

Note that the linear dynamic part of the model is represented in the state space form given by equations (2.1) and (2.2). The matrices Φ, G, H, A and B as well as the noise covariance matrices will be determined using the CVA method with the states generated by canonical correlations analysis between the *past* and *future* spaces.

2.8.2 The Narendra-Gallman Algorithm

The iterative algorithm of Narendra and Gallman (1966) updates the linear dynamic element and the nonlinear gain polynomials separately and sequentially. This algorithm is appealing for two reasons : (1) Robustness to high noise levels and (2) Ease of adaptation for the case where no *a priori* assumption regarding model order is made. The NGA based Hammerstein model identification procedure is formulated in a formal manner below.

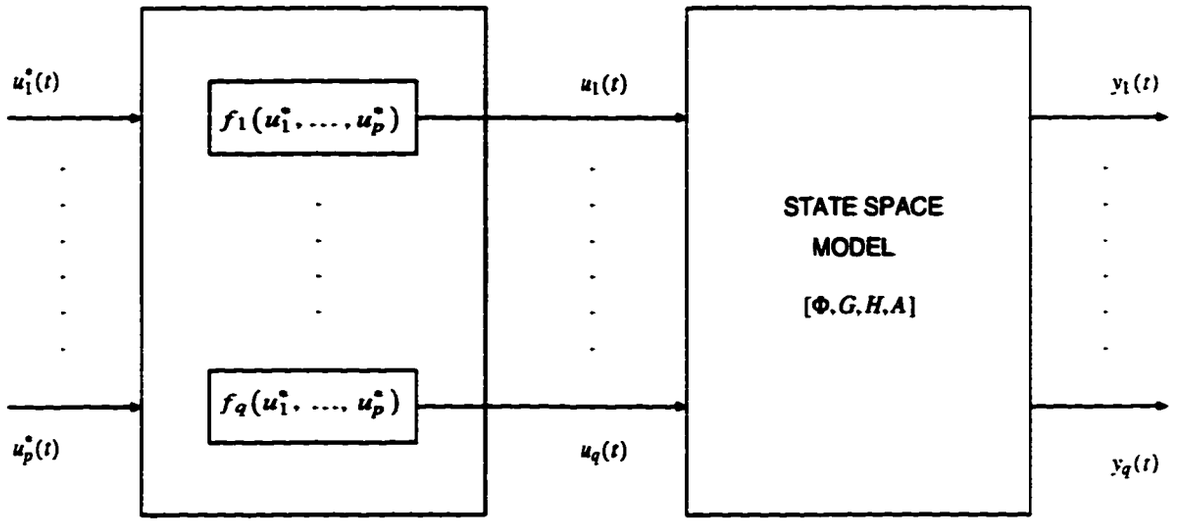

Figure 2.22: Combined Parameterization

The mathematical development will be based on the separate parameterization structure of the MIMO Hammerstein model but the analysis holds for the combined representation as well.

Consider, once again, the MIMO Hammerstein model of a nonlinear system as shown in Figure 2.21. The physical inputs to the plant are denoted by $\underline{u}^*(t) = \left[u_1^*(t), \cdots, u_p^*(t)\right]$ and the inputs to the linear part of the model are represented as $\underline{u}(t) = [u_1(t), \cdots, u_p(t)]$. The nonlinearities are all assumed to be static and confined to the plant inputs, $\underline{u}^*(t)$, and hence removed upon their transformation to the intermediate variables, $\underline{u}(t)$. The dynamics are modelled by the state space model relating $\underline{u}(t)$ to $\underline{y}(t)$. Suppose the parameters (Φ, G, H, A) of the dynamic element are known. The nonlinear part Γ can be obtained as follows :

Let the coefficients of the polynomial nonlinearities for each of the inputs u_i^* (the index i in this and the following mathematical expressions run from 1 to p unless otherwise stated) be expressed as $\Gamma_i = [\gamma_{i1}, \ldots, \gamma_{ih}]$. All of these can then be incorporated in a single vector of coefficients, $\Gamma = [\Gamma_1, \ldots, \Gamma_p]^T$. The variables formed by considering the powers of each of the plant inputs can be put together in the vector, $\Lambda_i(t) = \left[u_i^*(t)\ u_i^{*2}(t)\ \ldots\ u_i^{*h}(t)\right]$. This leads to the following relationship in the Hammerstein model :

$$u_i(t) = \Lambda_i(t)\Gamma_i^T \tag{2.35}$$

In the MIMO scenario, equation (2.35) can be compactly represented as :

$$\underline{u}^T(t) = \begin{bmatrix} \Lambda_1(t) & 0 & 0 & \cdots & 0 \\ 0 & \Lambda_2(t) & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & \Lambda_p(t) \end{bmatrix} \Gamma = \Lambda(t)\Gamma \tag{2.36}$$

which when combined with equations (2.1) and (2.2) gives :

$$\underline{y}^T(t) = \left\{\left[H(zI-\Phi)^{-1}G + A\right]\Lambda(t)\right\}\Gamma \tag{2.37}$$

Rewrite equation (2.37) as :

$$\underline{y}^T(t) = C(t)\Gamma \tag{2.38}$$

where

$$C(t) = \left[H(zI-\Phi)^{-1}G + A\right]\Lambda(t) \tag{2.39}$$

Equation (2.38) represents a system of linear equations with the well known least squares solution

$$\Gamma = \Sigma^{-1}_{c(t)c(t)}\Sigma_{c(t)y^T(t)} \tag{2.40}$$

Again, when the nonlinear part is "known," the input to the linear part $\underline{u}(t)$ can be computed as

$$\underline{u}^T(t) = \Lambda(t)\Gamma \tag{2.41}$$

This input signal is then used to get better estimates for the linear part (Φ,G,H,A) of the model.

The main steps of the multivariable Hammerstein model identification procedure can be summarized as follows :

1. Get the best linear model relating $\underline{u}^*$ and $\underline{y}$ using the CCA method.
2. Using the values of (Φ,G,H,A), calculate Γ using equations 2.37 through 2.40. Normalize Γ_i as, $\Gamma_i = \Gamma_i / \| \Gamma_i \|_\infty$. Stack up the Γ_i's to get Γ. (The infinity norm of the vector Γ_i is defined as, $\| \Gamma_i \|_\infty = \max(abs(\Gamma_i))$).
3. Recalculate, using equation 2.41, input $\underline{u}(t)$ for the linear part with the value of Γ obtained in Step 2.
4. Improve the estimates of the dynamic linear part by building a CCA model relating $\underline{u}$ to $\underline{y}$.
5. Check for convergence of Γ. If converged, Stop. Else, go to Step 2.

2.8.3 Illustrative Examples

Case Study 1 : Heat Exchanger

In this section, experimental data obtained from a heat exchanger by Eskinat *et al* (1991) are analyzed[2]. The experimental setup with the details of the hardware equipment are available in the cited reference. The nonlinearity in the system is caused by the presence of two distinct operating regions corresponding to the high and low process water flow rates. A data set containing 334 input-output samples was made available. Of this data set, 75% of the samples were used to construct the model and the rest to validate it. The input is the voltage signal to the valve and the output is the temperature in degrees Celsius (see Figure

[2]I like to thank Dr. Eskinat and Prof. Luyben for providing this data

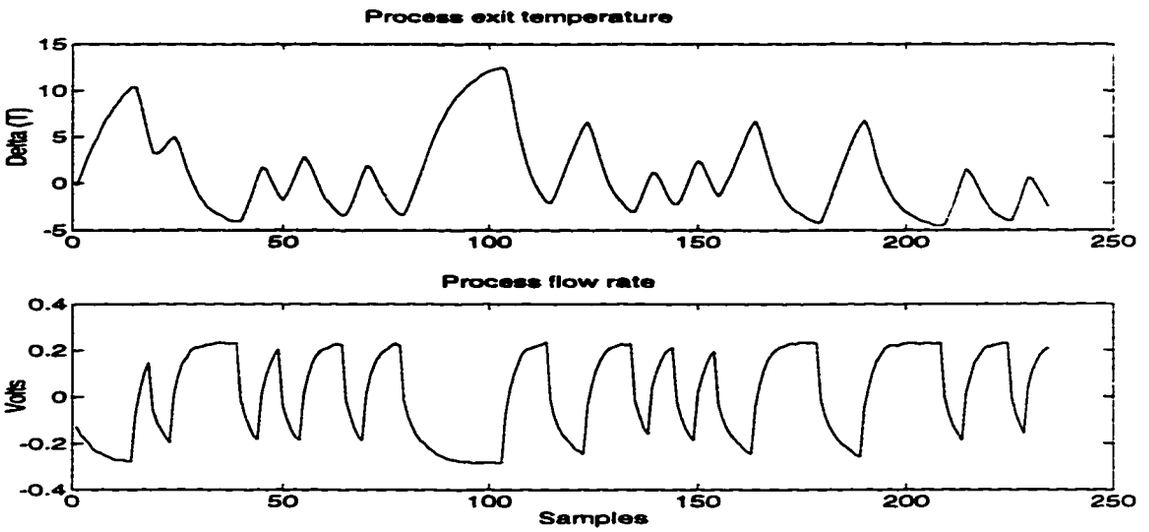

Figure 2.23: Plant Data for SISO Analysis : Heat Exchanger

Table 2.6: Summary of identification results for the heat exchanger data

Maximum memory length	10
Optimal memory length	4 (using AIC)
Optimal Plant Order	1 (using AIC)
Γ	$[0.1138 \ -0.1006 \ -0.1976 \ -1.0000]^T$
Φ	0.8074
G	-12.7827
H	4.3668
A	-10.5059
B	3.6207
Q	4.2126e-04
R	0.0029

2.23). A fourth order polynomial was chosen to represent the nonlinearity. The results of this identification exercise are shown in Table 2.6.

Inadequacy of a linear model to characterize such behavior is clearly demonstrated in Figure 2.24 where the actual process response is compared with some other models. Hammerstein models identified using the proposed method and that of Eskinat *et al.* (1991) predict the nonlinear response in the heat exchanger very closely. The quality of both these models is virtually the same ; however, in the proposed method the model order for the linear dynamic part is not fixed *a priori*.

Case Study 2 : Acid-Base Neutralization Process

Control of pH is of crucial importance in many chemical and biochemical processes. First principles modelling gives highly nonlinear equations which involve the equilibrium constants that are often unavailable. An empirical modelling approach is ideal in such a scenario.

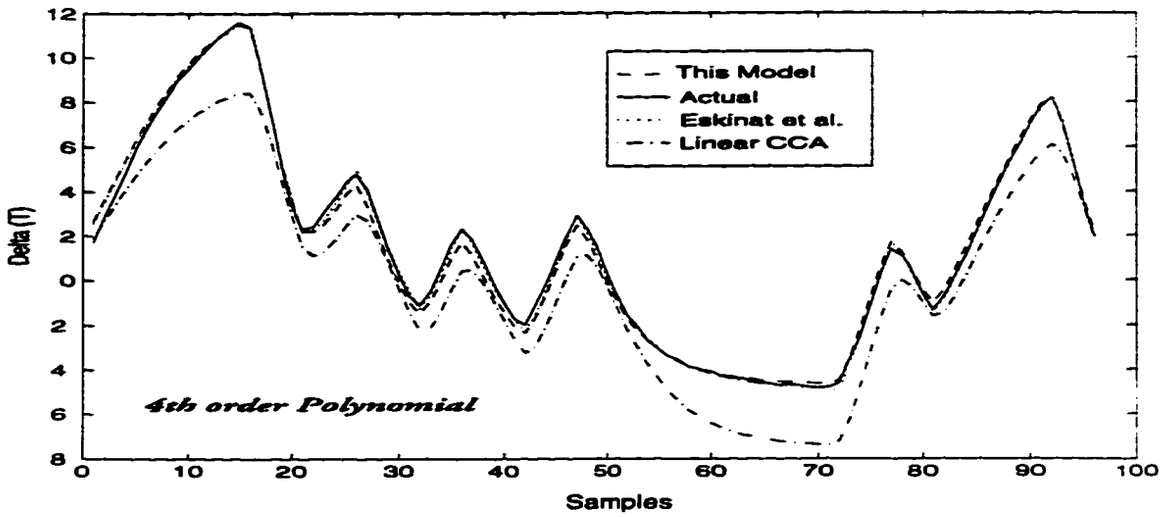

Figure 2.24: Cross Validation for SISO Model : Heat Exchanger data

The example considered is an acid-base neutralization process performed in a single tank. The system description, the nonlinear process model and the operating conditions can be found in Henson and Seborg (1994). The level and pH of the liquid in the well stirred neutralization tank are the two outputs that are manipulated by the acid and base flow rates. Data were collected by perturbing the system inputs by ±10% of their nominal values using specially designed random signals (Hernández and Arkun (1993)) that enable good nonlinear identification. Signal to noise ratio was kept at 10 for identification purposes. The convergence tolerance on Γ was set at 10^{-8} for the SISO, MISO and MIMO cases. In all the instances, the algorithm converged within 7 iterations. The AIC was used to obtain the optimal memory length and model order.

As a first step in the identification of this process, the SISO model that relates the acid flow rate to the pH was obtained. The input-output data used to obtain the model are shown in Figure 2.25. The model was validated by comparing the actual and predicted output of the data obtained from a different input-output sequence. This comparison is presented in Figure 2.26. It is seen that a Hammerstein model with a fourth order polynomial nonlinearity gives a good input-output mapping compared to the linear model.

The next step was to examine the application of this technique to the MISO system with pH as the system output that was to be related to the acid and base flow rates. See Figures 2.27 and 2.28 for the data used in model building and the results of the cross validation run respectively. In this case, it was found that the separate parameterization model yielded poor models and the combined parameterization with 3rd order nonlinearity worked adequately. The linear model is found to be totally unacceptable.

The true test of the theory presented in this work lies in its ability to identify a MIMO Hammerstein model of the process. Using the input-output sequences shown in Figure

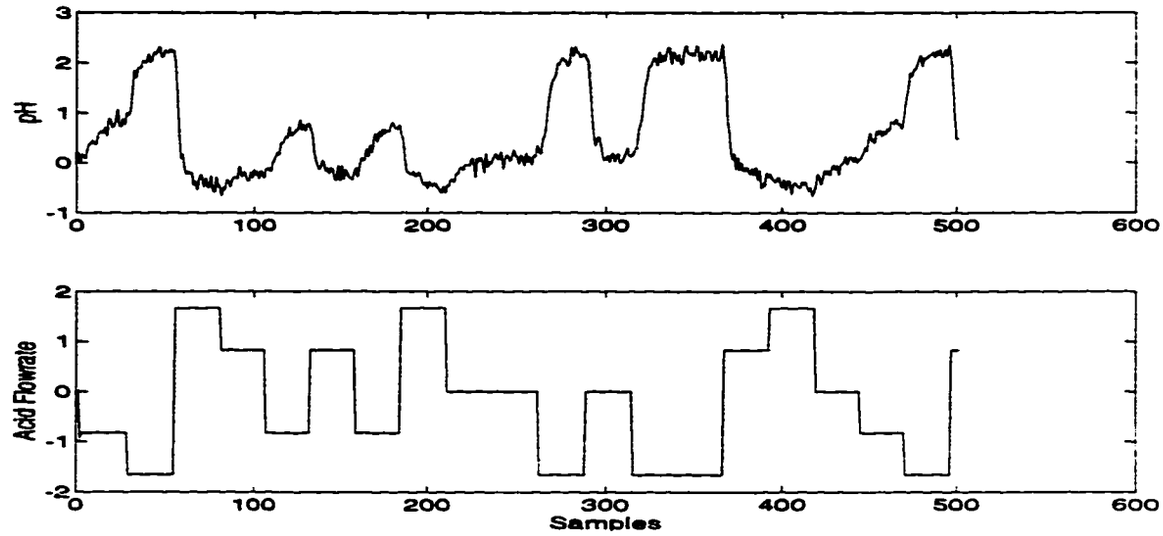

Figure 2.25: Plant Data for SISO Analysis : Acid-Base Neutralization Process

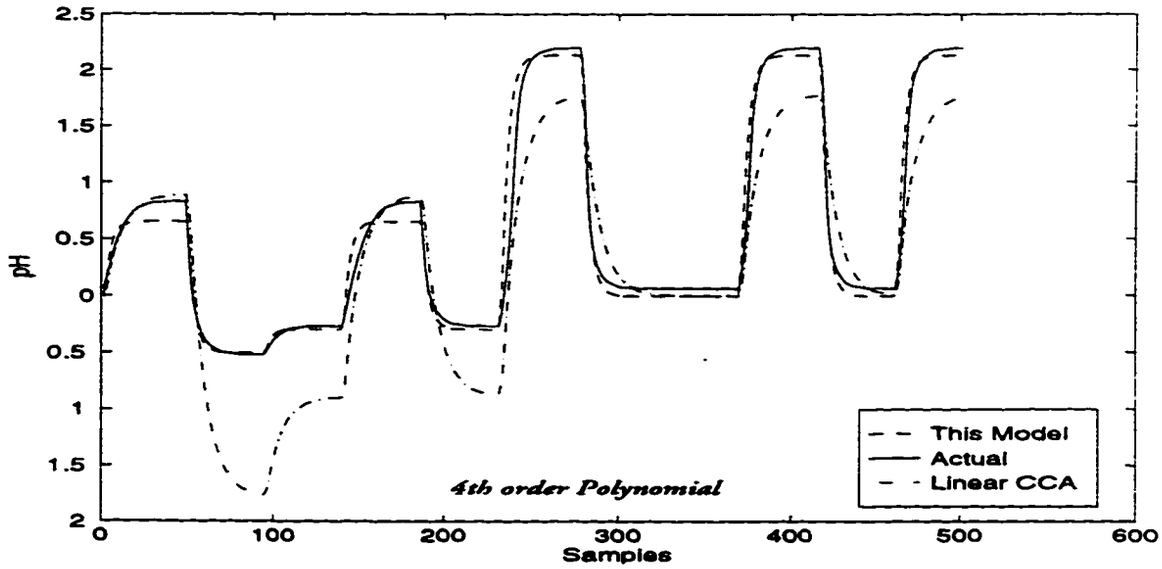

Figure 2.26: Cross Validation for SISO Model : Acid-Base Neutralization Process

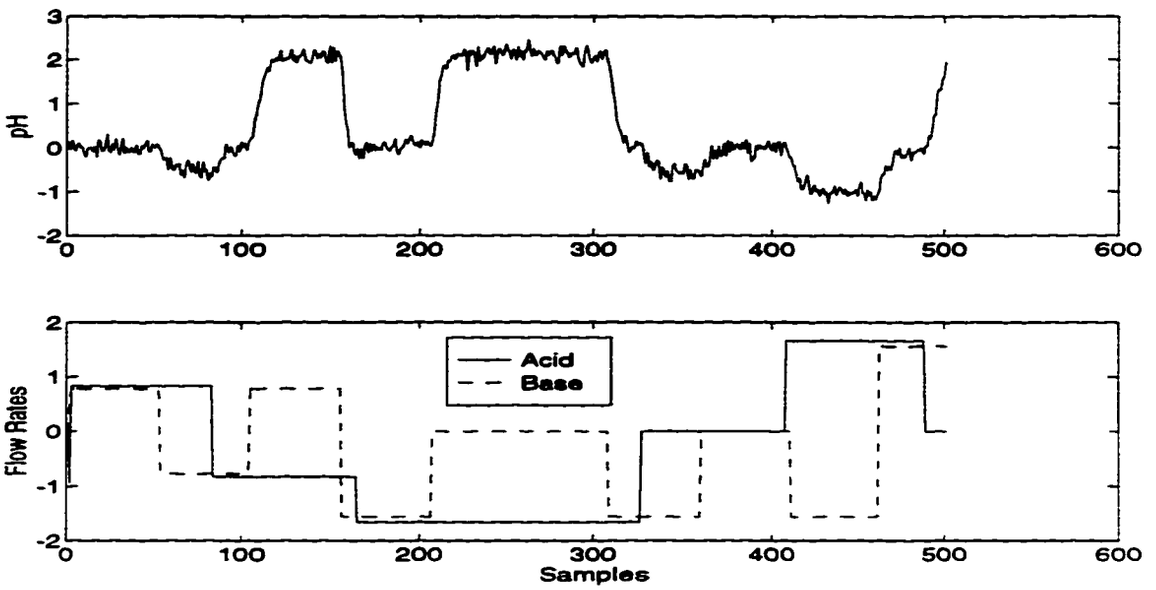

Figure 2.27: Plant Data for MISO Analysis : Acid-Base Neutralization Process

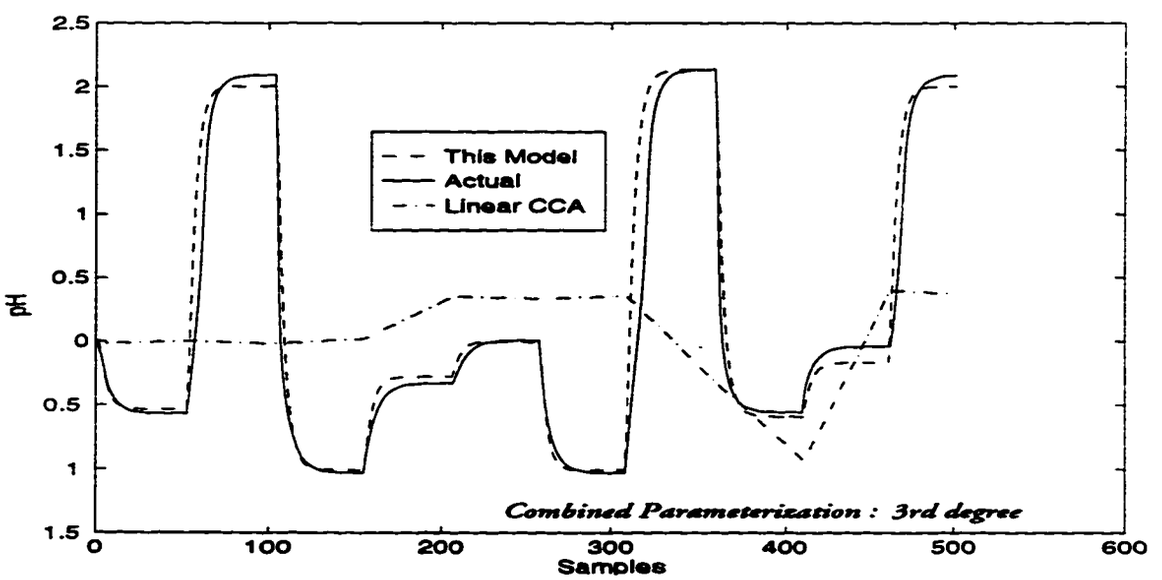

Figure 2.28: Cross Validation for MISO Model : Acid-Base Neutralization Process

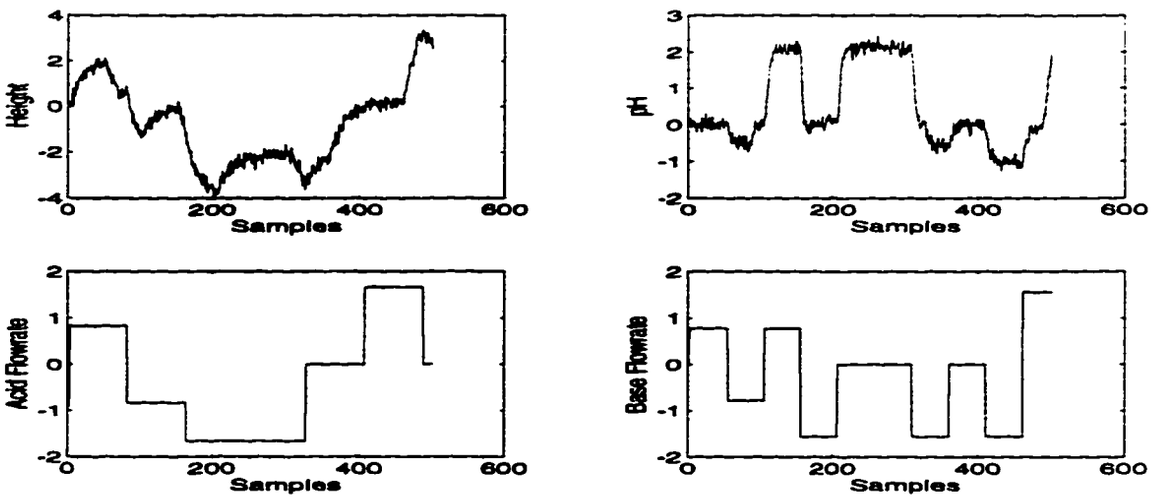

Figure 2.29: Plant Data for MIMO Analysis : Acid-Base Neutralization Process

Table 2.7: Summary of identification results for acid-base neutralization process : SISO case

Φ	$\begin{pmatrix} 0.7406 & -0.0053 \\ -2.7562 & -0.1664 \end{pmatrix}$
G	$\begin{pmatrix} -0.1185 \\ -1.2442 \end{pmatrix}$
H	$(\,1.0059 \quad -0.0569\,)$
A	$(\,-0.0371\,)$
Nonlinearity	$u = u^* - 0.5001\,u^{*2} + 0.2148\,u^{*3} - 0.0338\,u^{*4}$

2.29, a jointly parameterized 3rd order polynomial Hammerstein model was identified. The output predictions of this model for two different input sequences are presented in Figures 2.30 and 2.31. It is observed that the model predictions and the true process outputs are in close agreement. Unacceptable models were obtained using the separate parameterization structure. The linear models capture the dynamic relationship for the output variable height accurately while they fail to do so for the pH. This is expected because the nonlinearity in the system is associated with the pH measure in the system. Tables 2.7 through 2.9 summarize the identified models for the SISO, MISO and MIMO cases respectively.

Comparison of SISO and MISO models based on other existing algorithms (*e.g.* Eskinat *et al.* (1991), Kortmann and Unbehauen (1987)) are not included here. If a Hammerstein model is used to explain the plant behavior, all identification methods will very likely produce equivalent models *as long as* the model order of the dynamic part is picked carefully. By including the SISO and MISO examples, it is shown that the proposed strategy encompasses other methods in terms of producing acceptable models. In addition, this algorithm has the capability of modelling MIMO systems and also the flexibility of not having to fix the dynamic model structure *a priori*.

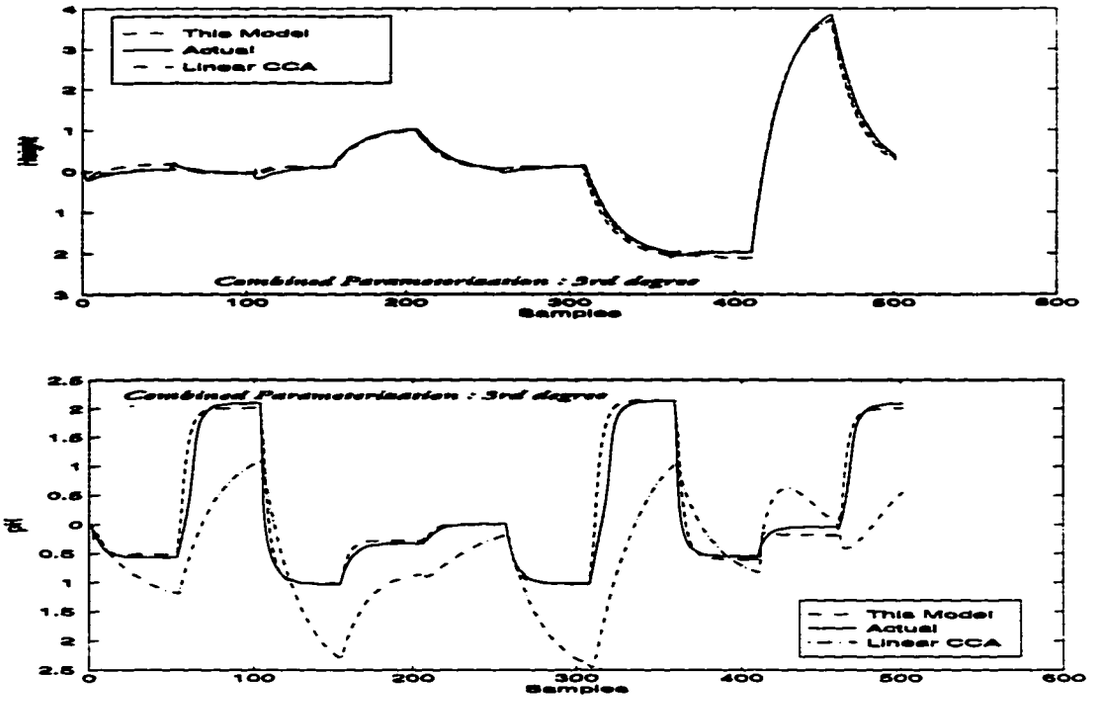

Figure 2.30: Cross Validation for Input Sequence 1 : MIMO Model (Acid-Base Neutralization Process)

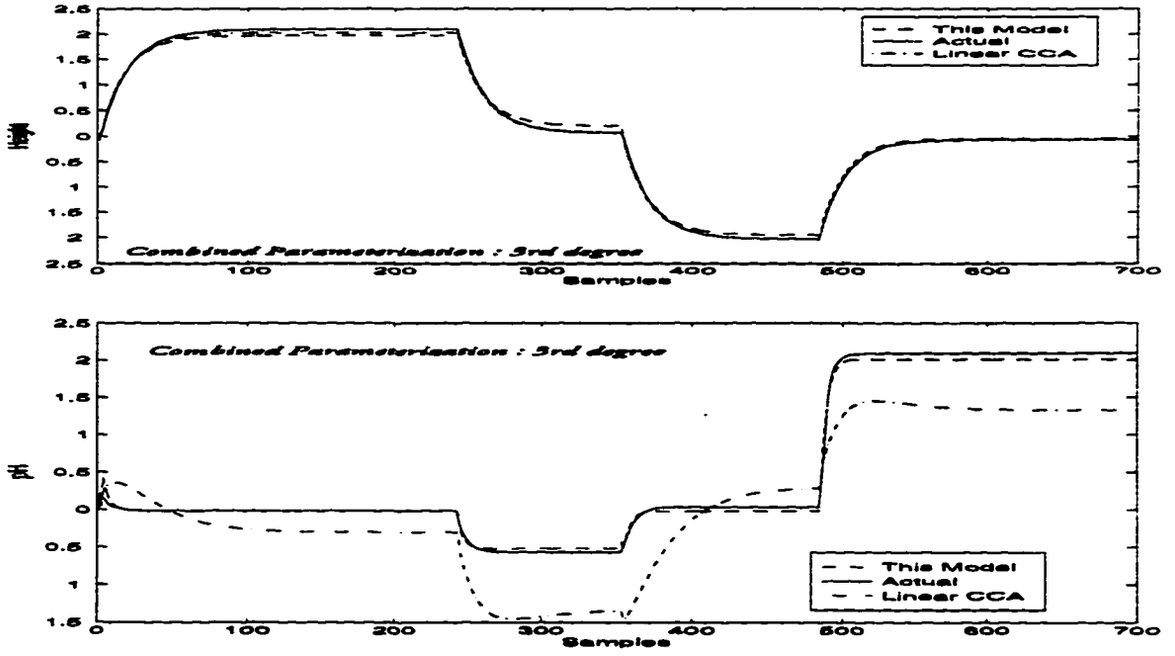

Figure 2.31: Cross Validation for Input Sequence 2 : MIMO Model (Acid-Base Neutralization Process)

Table 2.8: Summary of identification results for acid-base neutralization process : MISO case

Φ	$\begin{pmatrix} 0.6521 & -0.0214 \\ -3.0716 & 0.2376 \end{pmatrix}$
G	$\begin{pmatrix} 0.1973 \\ 1.7144 \end{pmatrix}$
H	$(1.0646 \quad -0.0648)$
A	(0.0764)
Nonlinearity	$u = 0.8781u_1^* + u_2^* + 0.5196u_1^{*2} + 0.2696u_2^{*2} - 0.8467u_1^*u_2^*$ $-0.0479u_1^{*3} - 0.0125u_2^{*3} - 0.3131u_1^*u_2^{*2} + 0.3014u_1^{*2}u_2^*$

Table 2.9: Summary of identification results for acid-base neutralization process : MIMO case

Φ	$\begin{pmatrix} 0.7903 & -0.0415 & -0.0309 & 0.0033 \\ -0.0575 & 0.7042 & -0.0475 & 0.0244 \\ 2.0911 & -0.0062 & 0.3594 & -0.0421 \\ 3.1760 & 2.3832 & -0.2177 & 0.1914 \end{pmatrix}$
G	$\begin{pmatrix} 0.1467 & 0.0569 & -1.4198 & -2.3193 \\ 0.0636 & -0.2557 & -0.9676 & 0.6817 \end{pmatrix}^T$
H	$\begin{pmatrix} 1.7389 & -0.4585 & 0.1167 & 0.0053 \\ -0.6846 & -0.7915 & -0.0177 & -0.0637 \end{pmatrix}$
A	$\begin{pmatrix} 0.0823 & 0.0623 \\ -0.0896 & 0.0402 \end{pmatrix}$
Nonlinearities	$u_1 = u_1^* - 0.3056u_2^* - 0.3490u_1^{*2} - 0.1719u_2^{*2} + 0.5663u_1^*u_2^*$ $+0.0322u_1^{*3} + 0.0326u_2^{*3} + 0.1987u_1^*u_2^{*2} - 0.2144u_1^{*2}u_2^*$
	$u_2 = 0.0031u_1^* + u_2^* + 0.2896u_1^{*2} + 0.1983u_2^{*2} - 0.5149u_1^*u_2^*$ $-0.0356u_1^{*3} + 0.0677u_2^{*3} - 0.2032u_1^*u_2^{*2} + 0.1452u_1^{*2}u_2^*$

2.9 Conclusions

Besides the tutorial introduction of the CVA technique for the identification of linear empirical state space models, this chapter examined several important aspects concerning the identification of empirical process models from experimental data. The important conclusions are presented below.

- Extensive simulations were performed to illustrate properties of the CVA algorithm. The CVA algorithm is found to be very reliable and robust compared to the N4SID algorithm. Other studies that compare the various subspace identification approaches corraborate to this fact.
- A constrained model based predictive control algorithm (DMC) was implemented on the laboratory stirred tank heater. The CVA algorithm was used to develop a state space model from plant data and used in the DMC algorithm.
- An automated multi-input multi-output model identification procedure that does not presuppose any model structure for the linear part of the Hammerstein model is

presented.

- The Narendra-Gallman algorithm has been adapted and extended for performing multivariable nonlinear model identification.
- Using experimental data from a heat exchanger and a simulated acid-base neutralization system, it is shown that a significant improvement in modelling is achieved with the use of Hammerstein models rather than linear models.
- For multivariable systems, the nonlinear process dynamics are modelled well by the jointly parameterized MIMO Hammerstein models rather than the separately parameterized structure.

Chapter 3

Modelling and Control of Multivariable Processes : The Dynamic Projection to Latent Structures Approach

3.1 Overview

This chapter addresses the issue of modelling and control of multivariable chemical process systems using the dynamic version of a popular multivariate statistical technique. namely. Projection to Latent Structures (Partial Least Squares or PLS). Discrete input-output data is utilized to construct a *projection based* dynamic model that captures the dominant features of the process under study. The structure of the resulting model enables the synthesis of a multi-loop control system. In addition, the design of feedforward control for multivariable systems using the dynamic PLS framework is also presented. Three case studies will be used to illustrate the modelling and control of multivariable linear and nonlinear systems using the suggested approach.

[1] Sections of this chapter have been submitted for possible publication as : S. Lakshminarayanan. Sirish L. Shah and K. Nandakumar,"Modelling and Control of Multivariable Systems : The Dynamic Projection to Latent Structures Approach". Submitted to the AIChE Journal. July 1996.

3.2 Contributions of this chapter

- A new modelling and control approach is demonstrated for MIMO (multiple input. multiple output) systems. The key idea is to cast the MIMO problem as a series of SISO (single input, single output) problems.
- A novel strategy for multivariable feedforward control design is presented. The elements of this multivariable controller have a simple representation (like the SISO case) and are easily computed.
- Synthesis of nonlinear feedforward controllers is illustrated using the example of an acid-base neutralization process.
- In the process of synthesizing PLS based multivariable feedforward controllers, the Cramer's rule for the solution of a system of linear equations has been extended to include nonsquare systems. Details of this can be found in Appendix B.

3.3 Introduction

Advanced control algorithms such as Model Predictive Controllers (DMC. GPC etc.) are gaining increasingly wide acceptance in the chemical process industries mainly due to their ability to deal with (a) multivariable (square or nonsquare) systems and (b) systems with hard and soft constraints. These control algorithms employ simple and intuitive process descriptions such as step/impulse response coefficients and discrete transfer functions. However. the design of such controllers is possible only after the development of a complete model describing the effect of all the process inputs on all the process outputs. First principles based models are either difficult to obtain or too unwieldy to use for controller design. The multivariable process model is usually obtained empirically by performing an identification experiment and analyzing the recorded plant input-output data. The presence of several time scales and different delays MIMO processes presents a challenging problem in the identification of such systems. If the system were to exhibit nonlinear characteristics over the desired range of operation, the tasks of identification and control can become even more formidable. Even if an adequate plant model was available, the issue of control structure selection (centralized/decentralized) needs to be addressed. In model predictive control (MPC), the controller is centralized and reliability is achieved by performing online optimization. Morari (1990) points out that there are many cases where the modelling and design effort necessary for MPC is either impossible or not economically justifiable. In practice, a decentralized (multi-loop) control structure is preferred for ease of startup. bumpless automatic/manual transfer, and fault tolerance in the event of actuator or sensor failures and is readily designed using recently developed control algorithms (Seborg *et al.* (1989), Ogunnaike and Ray (1994)) or other methodologies (Morari and Zafiriou. 1989).

The decision on loop pairing is critical - the Relative Gain Array (RGA) method and its extensions as well as physical arguments are the key tools to screen potential alternatives.

Identification of SISO systems is an extremely well researched topic even for nonlinear systems (Ljung (1987), Cinar (1994)). For linear SISO systems, least squares based techniques have proven to be handy in the recursive as well as nonrecursive identification schemes. In addition, it is also possible to identify the parameters of all orders, from zero to a user specified maximum, using an efficient implementation of the least squares algorithm (Niu and Fisher, 1994). Several commercial identification and control packages (e.g. MATLAB System Identification Toolbox (1992), CONSYD (1989), ADAPTx (1992)) are capable of estimating linear SISO and/or MIMO dynamic models from observed plant data. In the identification of MIMO processes, a high degree of correlation is often observed between process variables. In such cases, use of identification software based on the ordinary least squares technique will result in parameter estimates with large variances owing to the ill-conditioned nature of the problem. One way to circumvent the ill-conditioned nature of the MIMO identification problem is to resort to alternatives other than the ordinary least squares. Very recently, multivariate statistical techniques such as PCA (Principal Components Analysis) and PLS have been applied to chemical engineering problems involving process monitoring, fault detection and modelling (Kresta (1992), Wise (1991). Qin and McAvoy (1992a), Qin (1993), Nomikos and MacGregor (1994), Ricker (1988)). However. very few attempts have been made to exploit the potential advantages that PLS has to offer in the domain of dynamic modeling and control. Some possible methods along with their benefits/drawbacks are described in Kaspar and Ray (1992, 1993). Modeling of nonlinear *static* data using an integrated PLS-neural net model has been demonstrated by Qin and McAvoy (1992b). The last three papers cited above provided the motivation for the current investigation in which identification and control are performed in the PLS framework.

The key contribution of this work involves the development of a modelling and control approach for MIMO processes that is cast as a series of SISO identification and control problems. Thus one can, by employing the proposed strategy, utilize the wealth of identification and control algorithms that have been developed for SISO systems. Linear systems are easily handled using standard time series representations (e.g. ARX models); the Hammerstein structure provides a framework for handling nonlinear systems. Although it can be argued that the Hammerstein structure cannot handle every type of nonlinearity, their utility in modelling typical processes (heat exchangers, high purity distillation columns, acid-base neutralization systems etc.) has been shown in earlier work (Eskinat *et al.* (1991), Lakshminarayanan *et al.* (1995)). Subsequent *decentralized* controller design is based on the estimated SISO dynamic models which provides an automatic selection of loop pairing. The control structure involves the use of pre- and post-compensators along with provisions for annulling the nonlinearities that are identified from the plant data. Finally, a strategy for the design of multivariable feedforward controllers in this PLS framework is proposed.

The subject matter of this chapter is outlined as follows. First an extension of the PLS

technique (covered in Chapter 1) to handle dynamic linear and nonlinear process data in a manner that facilitates easier control system design is presented. This is followed by a section describing the synthesis of multivariable feedforward controllers. The theoretical matter is supplemented by including several semi-realistic examples involving modelling and control of multivariable chemical process systems.

3.4 Dynamic Extension of the PLS algorithm

In Chapter 1, the PLS algorithm was presented to build static linear models. Some possible modifications to handle static nonlinearities were also outlined. In this section, the basic PLS algorithm is extended in order to model dynamic multivariable processes.

The dynamic analog of equation (1.1) can be written as follows :

$$Y = X C_{dyn} + Noise \tag{3.1}$$

where Y represents the output or controlled variables and X the manipulated variables (inputs). The important difference between equations (1.1) and (3.1) is that while C represents a static map in equation (1.1), the matrix C_{dyn} in equation (3.1) is a dynamic mapping relating the manipulated inputs to the controlled outputs.

An obvious way to model dynamic processes with PLS is to include past values of the input and/or output variables in the input data matrix **X**; the algebraic PLS algorithm still forms the computational machinery and does model reduction in a statistically sound manner. This would mean that we need to deal with huge matrices particularly for MIMO systems. More importantly, the use of the resulting model may only be limited to providing a good input-output mapping of the process rather than aiding the synthesis of a control system (particularly for nonlinear systems). With this approach, C_{dyn} is a matrix of constants whose elements can be interpreted as finite impulse response (FIR) coefficients (Ricker, 1988) or as a multivariate autoregressive moving average (ARMA) model (Qin and McAvoy, 1992a).

A dynamic PLS modelling procedure that can be directly utilized for control system design has been reported in the literature (Kaspar and Ray, (1992,1993)). Their method does not involve the use of lagged variables but is based on the filtering of input data. In this way, they argue, the major dynamic component in the data is removed and the filtered data can be analyzed using the standard PLS procedure. The dynamic filter is designed either by using some *a priori* knowledge of the process (in the form of an *average* dynamics) or by minimizing the sum of squares of the output residuals, F_{n+1}. In the former case, all the dynamic filters are identical and equal to the assumed *average* dynamics. In the latter case, the dynamic filters are determined using the optimization objective stated above and hence are generally distinct from one another. Using several simulation examples, Kaspar and Ray (1992, 1993) have demonstrated the utility of their approach for the identification

and control of systems described by linear models.

A dynamic extension of the PLS algorithm that is based on the direct modification of the PLS inner relation is proposed. Instead of relating the input and output scores (i.e. t_i and u_i) using a static linear or nonlinear model, a dynamic component such as the ARX or the Hammerstein model may be used. In Kaspar and Ray (1992, 1993), this approach was quickly dismissed as being suboptimal in terms of the PLS outer relationship. This suboptimality problem comes into prominence only when no attention is placed on the design of the plant probing signals. Employing input signals with sufficient low frequency content, the proposed method identifies *adequate* plant models by utilizing the techniques developed for SISO systems. The proposed strategy will be particularly convenient in the modelling of nonlinear multivariable systems - for example, instead of using tedious multivariable Hammerstein models one can piece together several univariate Hammerstein models to obtain an overall model.

For linear systems, although the PLS model matrices and the dynamic inner relationships identified using the proposed strategy and the Kaspar-Ray approach are in general different, it turns out that the mathematical expressions for the steady state gains, transfer functions etc. are identical. This implies that once the model is identified using either of the methods, they can be used in exactly the same way for the synthesis of feedback and feedforward control structures.

For the modelling procedure based on incorporation of the linear dynamic relationship (linear systems) in the PLS inner model, the decomposition of the X block is as given by equation (1.15). The dynamic analog of equation (1.16) is given by (' refers to the transpose operator)

$$Y = G_1(t_1)q_1' + G_2(t_2)q_2' + \cdots + G_n(t_n)q_n' + F_{n+1} = Y_1^{exp} + Y_2^{exp} + \cdots + Y_n^{exp} + F_{n+1} \quad (3.2)$$

Here, the G_i's denote the linear dynamic models (e.g. ARX) identified at each stage and $G_i(t_i)q_i'$ quantifies the measure of Y space explained by the i^{th} PLS dimension (Y_i^{exp}). We now define the operator G as the diagonal matrix comprising the dynamic elements identified at each of the n PLS dimensions i.e.,

$$G = \begin{bmatrix} G_1 & 0 & 0 & \cdots & 0 \\ 0 & G_2 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & G_n \end{bmatrix} \quad (3.3)$$

A graphical sketch of the dynamic PLS modelling procedure is provided in Figure 3.1. Note that the scaling information has been explicitly included using matrices S_x^{-1} and S_y with the subscript 'Sca' denoting the scaled plant data. $\hat{Y}_{Raw}$ indicates the predicted values

of the controlled outputs.

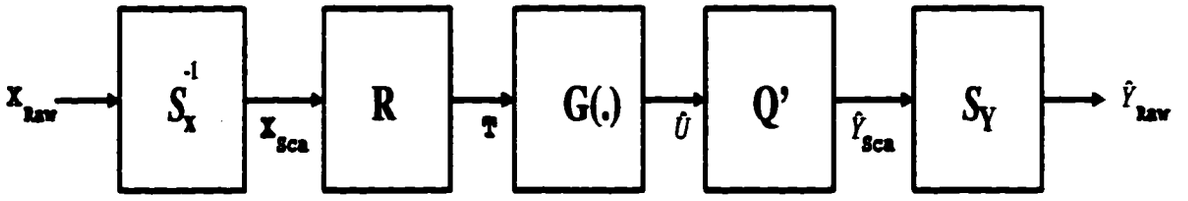

Figure 3.1: Information flow diagram for the proposed modelling strategy

For the model identified using the dynamic PLS algorithm, the transfer function relating input j to output i can be expressed as

$$\frac{\Delta y_i(z)}{\Delta x_j(z)} = \frac{sy_i}{sx_j}\left(\sum_{k=1}^{n} Q_{ik}G_k(z)R_{jk}\right) \tag{3.4}$$

with R_{jk} and Q_{ik} denoting the usual elements of matrices R and Q respectively (interpretation of the R matrix was presented in Chapter 1). It is seen that the transfer functions relating each plant input to each output is a linear combination of the dynamic elements identified at each PLS dimension. Depending on the relative magnitudes of R_{jk} and Q_{ik}, a particular dynamic component $G_k(z)$ *may or may not* contribute to the overall dynamics of that channel. Equation (3.4) is useful if it is desired to design a conventional control systems for the process.

As already mentioned, the Hammerstein structure will be employed to model nonlinear systems. Here, the score vectors obtained at each PLS dimension are related using a SISO Hammerstein model ("inner models" in Figure 1.1). As shown in Figure 3.2, the score vector t_i is transformed via a nonlinear static relationship (a polynomial of reasonable order) to t_i^*. A linear dynamic model (e.g. ARX model) is then determined between t_i^* and u_i. Though any method can be used for the identification of the Hammerstein models, the SISO version of the algorithm presented in Lakshminarayanan *et al.* (1995) is employed.

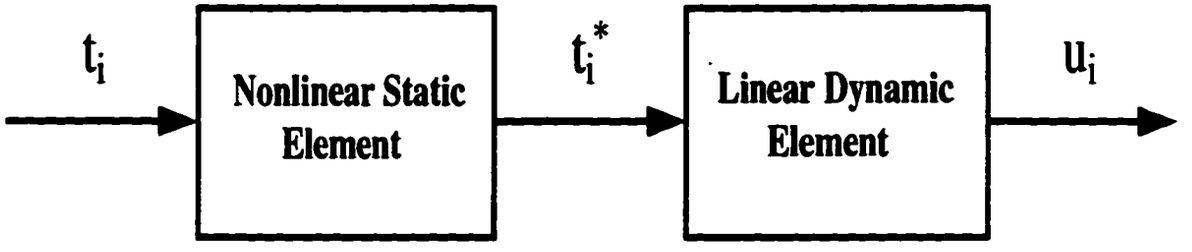

Figure 3.2: The Hammerstein Model

Denoting the identified Hammerstein models by H_i (i=1, $\cdots$, n), we obtain the equivalent of equation (3.2) as

$$Y = H_1(t_1)q_1' + H_2(t_2)q_2' + \cdots + H_n(t_n)q_n' + F_{n+1} = Y_1^{exp} + Y_2^{exp} + \cdots + Y_n^{exp} + F_{n+1} \quad (3.5)$$

Such an approach can be extended to include other nonlinear system parameterizations such as the Wiener model (this is done in the next chapter) or the Nonlinear time series models (e.g. nonlinear autoregressive with exogeneous inputs (NARX) and nonlinear autoregressive moving average with exogeneous inputs (NARMAX)).

3.5 Illustrative Examples of the Modelling Strategy

The proposed modelling strategy has been successfully applied to several multivariable systems. Three case studies involving a distillation column, a heated rod system and an acid-base neutralization system are presented here. Application of this procedure to the identification and control of the laboratory CSTH (described in Chapter 2) will be covered in the following chapter.

3.5.1 Example 1 : Distillation Column

Wood and Berry (1973) reported the following transfer functions for methanol-water separation in a distillation column. The composition of the top and bottom products expressed in weight percent of methanol are the controlled variables. The reflux and the reboiler steam flow rates are the manipulated inputs expressed in lb/min. Time is in minutes.

$$\begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix} = \begin{pmatrix} \frac{12.8e^{-s}}{16.7s+1} & \frac{-18.9e^{-3s}}{21s+1} \\ \frac{6.6e^{-7s}}{10.9s+1} & \frac{-19.4e^{-3s}}{14.4s+1} \end{pmatrix} \begin{bmatrix} x_1(s) \\ x_2(s) \end{bmatrix} \quad (3.6)$$

The transfer function form of the disturbance channel (feed flow rate and feed composition are the disturbances) is given by

$$\begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix} = \begin{pmatrix} \frac{3.8e^{-8.1s}}{14.9s+1} & \frac{0.22e^{-7.7s}}{22.8s+1} \\ \frac{4.9e^{-3.4s}}{13.2s+1} & \frac{0.14e^{-9.2s}}{12.1s+1} \end{pmatrix} \begin{bmatrix} d_1(s) \\ d_2(s) \end{bmatrix} \quad (3.7)$$

To model the relationship between the manipulated inputs and the controlled outputs. plant data was *collected* by exciting the plant with a series of step changes to the reflux and reboiler steam flow rates. The signal to noise ratio (SNR) was set at 10 by adding measurement noise. Following autoscaling of the inputs and outputs, the PLS based modelling was attempted. The dynamic elements were restricted to be second order (in both numerator and denominator) with delay for both the PLS dimensions. The resulting PLS model is

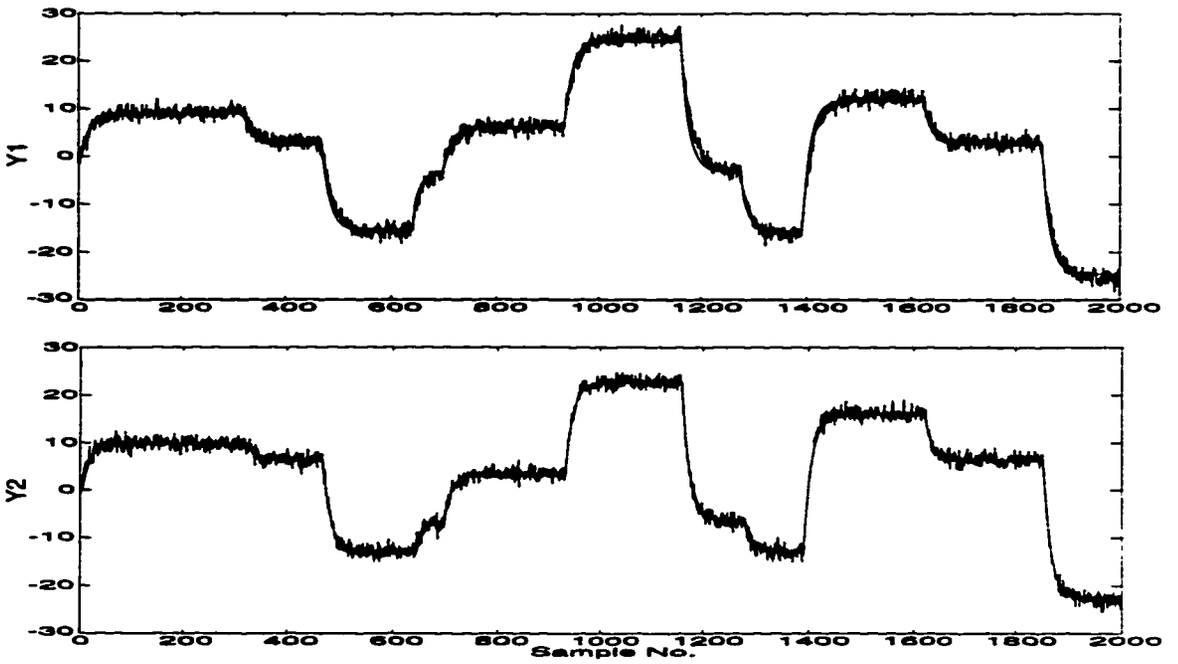

Figure 3.3: Identification of the Wood-Berry Column : Model (dashed line) and Actual Plant (solid line) responses

$$S_x = \begin{bmatrix} 0.4770 & 0 \\ 0 & 0.6374 \end{bmatrix} ; S_y = \begin{bmatrix} 12.7577 & 0 \\ 0 & 12.2821 \end{bmatrix}$$

$$P = \begin{bmatrix} 0.3228 & 0.9455 \\ -0.9465 & 0.3256 \end{bmatrix} ; R = \begin{bmatrix} 0.3256 & 0.9465 \\ -0.9455 & 0.3228 \end{bmatrix} ; Q = \begin{bmatrix} 0.6972 & 0.7597 \\ 0.7169 & -0.6503 \end{bmatrix}$$

$$G_1 = \frac{0.1417z^{-5}}{1 - 0.4305z^{-1} - 0.4706z^{-2}}$$

$$G_2 = \frac{0.0529z^{-5} + 0.0291z^{-6}}{1 - 0.2336z^{-1} - 0.2321z^{-2}}$$

Figure 3.3 shows the fit obtained to the plant data with the identified model. Using a different set of plant input-output data, a cross validation test of the identified model is performed. The results shown in Figure 3.4 indicates that the identified model provides a good representation of the plant behavior.

Using equation (3.4), the steady state gain matrix of the identified model is

$$K = \begin{bmatrix} 12.4327 & -18.1543 \\ 6.5938 & -19.3396 \end{bmatrix}$$

This compares quite favorably with the steady state gains given in equation (3.6).

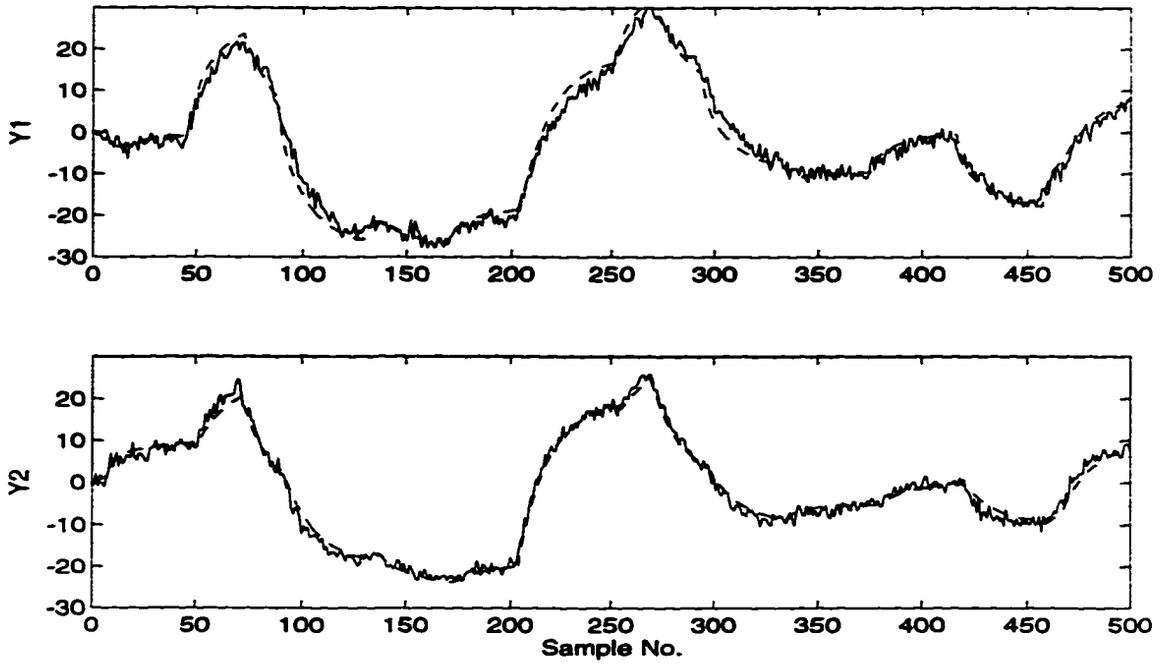

Figure 3.4: Cross validation for Wood-Berry Column : Model (dashed line) and Actual Plant (solid line) responses

3.5.2 Example 2 : Heated Rod System

Here, the heated rod system described by Kaspar and Ray (1993) is considered. The system consists of a rod with perfectly insulated ends being heated by three uniform heat sources along the length of the rod. Temperature measurements at the two ends of the rod and the internal boundaries of the heating zones (see Figure 3.5) are the outputs and the manipulated variables are the heat inputs applied from the heater elements. The system has three inputs and four outputs.

The physical phenomena of heat conduction in the rod can be represented by the following parabolic partial differential equation

$$\frac{\partial \Theta}{\partial t} = 0.25\frac{\partial^2 \Theta}{\partial Z^2} - \Theta + X(Z,t)$$

with the boundary conditions as :

$$\frac{\partial \Theta}{\partial Z} = 0 \text{ at } Z = 0$$
$$\frac{\partial \Theta}{\partial Z} = 0 \text{ at } Z = 1$$

Here Θ, Z and t denote the temperature, normalized spatial variable and time respectively. $X(Z,t)$ represents the spatially varying forcing function (the three zones of uniform heating)

For purposes of generating the input-output data, the transfer function matrix given in Kaspar and Ray (1993) is used rather than the partial differential equation system presented

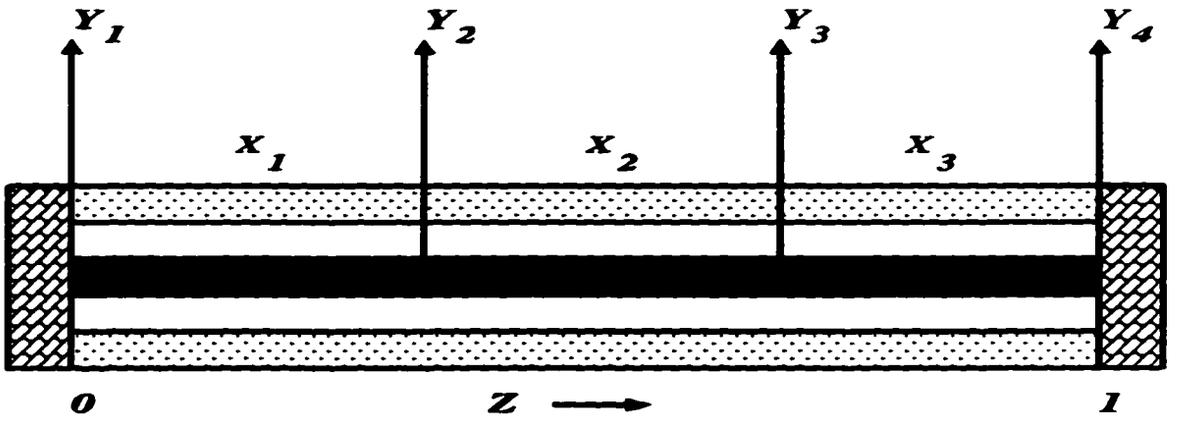

Figure 3.5: The Heated Rod System (Kaspar and Ray, 1993)

above. The transfer function matrix (in Laplace domain) for the above system is :

$$\begin{pmatrix} \frac{0.509}{0.688s+1} & \frac{0.289}{(0.989s+1)(0.145s+1)} & \frac{0.198}{(0.984s+1)(0.244s+1)(0.230s+1)} \\ \frac{0.400}{0.859s+1} & \frac{0.355}{0.932s+1} & \frac{0.243}{(0.991s+1)(0.274s+1)} \\ \frac{0.243}{(0.991s+1)(0.274s+1)} & \frac{0.355}{0.932s+1} & \frac{0.400}{0.859s+1} \\ \frac{0.198}{(0.984s+1)(0.244s+1)(0.230s+1)} & \frac{0.289}{(0.989s+1)(0.145s+1)} & \frac{0.509}{0.688s+1} \end{pmatrix}$$

Process data were obtained by perturbing the input signals. Measurement noise was added to the outputs to obtain a SNR of 10. The input-output data was autoscaled and then analyzed using the dynamic PLS algorithm. Three PLS dimensions were used to construct the model. For the dynamic elements, a discrete second order model with delay was identified. Figure 3.6 shows the model fit for all four outputs. Cross validation of the identified model using a different type of input sequence is shown in Figure 3.7. The model fit and cross validation results show that the identified model represents the actual behavior of the heated rod quite accurately. Details of the identified PLS model is provided below.

$$S_x = \begin{bmatrix} 1.9971 & 0 & 0 \\ 0 & 0.9667 & 0 \\ 0 & 0 & 0.5000 \end{bmatrix}$$

$$S_y = \begin{bmatrix} 0.9929 & 0 & 0 & 0 \\ 0 & 0.8116 & 0 & 0 \\ 0 & 0 & 0.5733 & 0 \\ 0 & 0 & 0 & 0.4984 \end{bmatrix}$$

$$P = \begin{bmatrix} 0.8490 & 0.5203 & 0.1530 \\ 0.4820 & -0.6380 & -0.6284 \\ 0.2165 & -0.5676 & 0.7627 \end{bmatrix}$$

$$R = \begin{bmatrix} 0.8444 & 0.5043 & 0.1356 \\ 0.4842 & -0.6152 & -0.5953 \\ 0.2296 & -0.6080 & 0.7934 \end{bmatrix}$$

$$Q = \begin{bmatrix} 0.5004 & 0.6790 & 0.3902 \\ 0.5151 & 0.3408 & 0.0340 \\ 0.5050 & -0.3380 & 0.0431 \\ 0.4789 & -0.5555 & 0.9191 \end{bmatrix}$$

$$G_1 = \frac{0.1055z^{-2} + 0.0567z^{-3}}{1 - 0.4792z^{-1} - 0.4433z^{-2}}$$

$$G_2 = \frac{0.0510z^{-1} + 0.0443z^{-2}}{1 - 0.3776z^{-1} - 0.4122z^{-2}}$$

$$G_3 = \frac{0.0479z^{-5} + 0.0220z^{-6}}{1 - 0.2954z^{-1} - 0.2677z^{-2}}$$

The above model yields the following steady state gain matrix which is in excellent agreement with that of the original system.

$$K = \begin{bmatrix} 0.5209 & 0.2811 & 0.2041 \\ 0.4018 & 0.3557 & 0.2563 \\ 0.2342 & 0.3569 & 0.3912 \\ 0.1844 & 0.2849 & 0.4983 \end{bmatrix}$$

3.5.3 Example 3. Acid-base Neutralization Process

The acid-base neutralization process that was considered in Chapter 2 is now revisited. The model that was identified (Table 2.9) provides a good input-output description. However. the control of the process using the above model would be unwieldy owing to the complexity of the static transformations. The control of the process is relatively straighforward with a dynamic PLS model as will be seen shortly. To recall, the level and pH of the liquid in the well stirred neutralization tank are the two outputs that are manipulated by the acid and base flow rates. The nominal value for the level and pH are 14 cm and 7.06 respectively. the acid and base flows being 16.6 ml/s and 15.6 ml/s respectively. Data was collected by perturbing the system inputs by ±10% of their nominal values using specially designed random signals (described in chapter 2). The sampling period is 15 seconds. A signal to noise ratio of 10 was used.

The input-output data were first autoscaled and then analyzed using the dynamic PLS

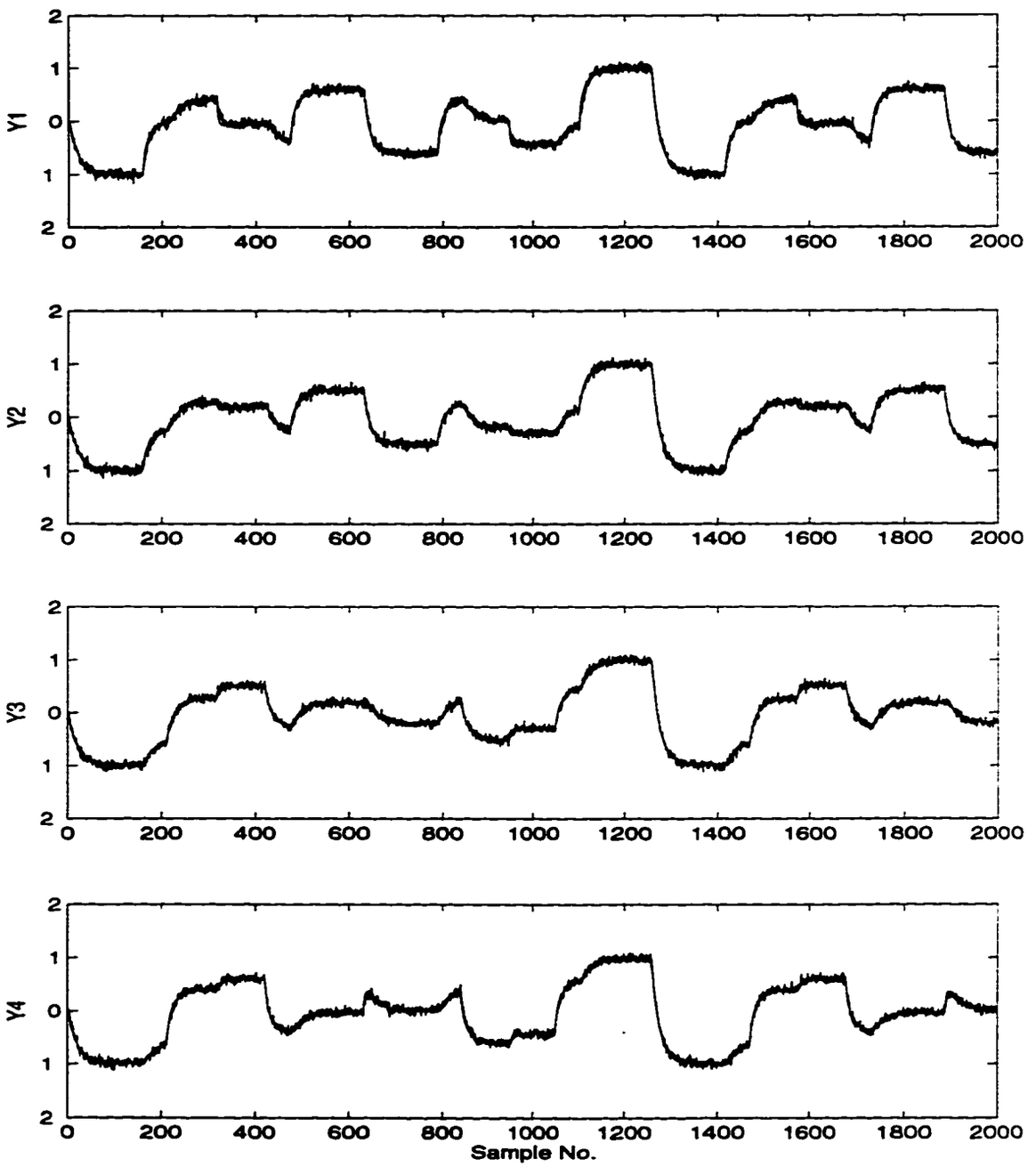

Figure 3.6: Identification results for the heated rod system : Model (dashed line) and Actual Plant (solid line) responses

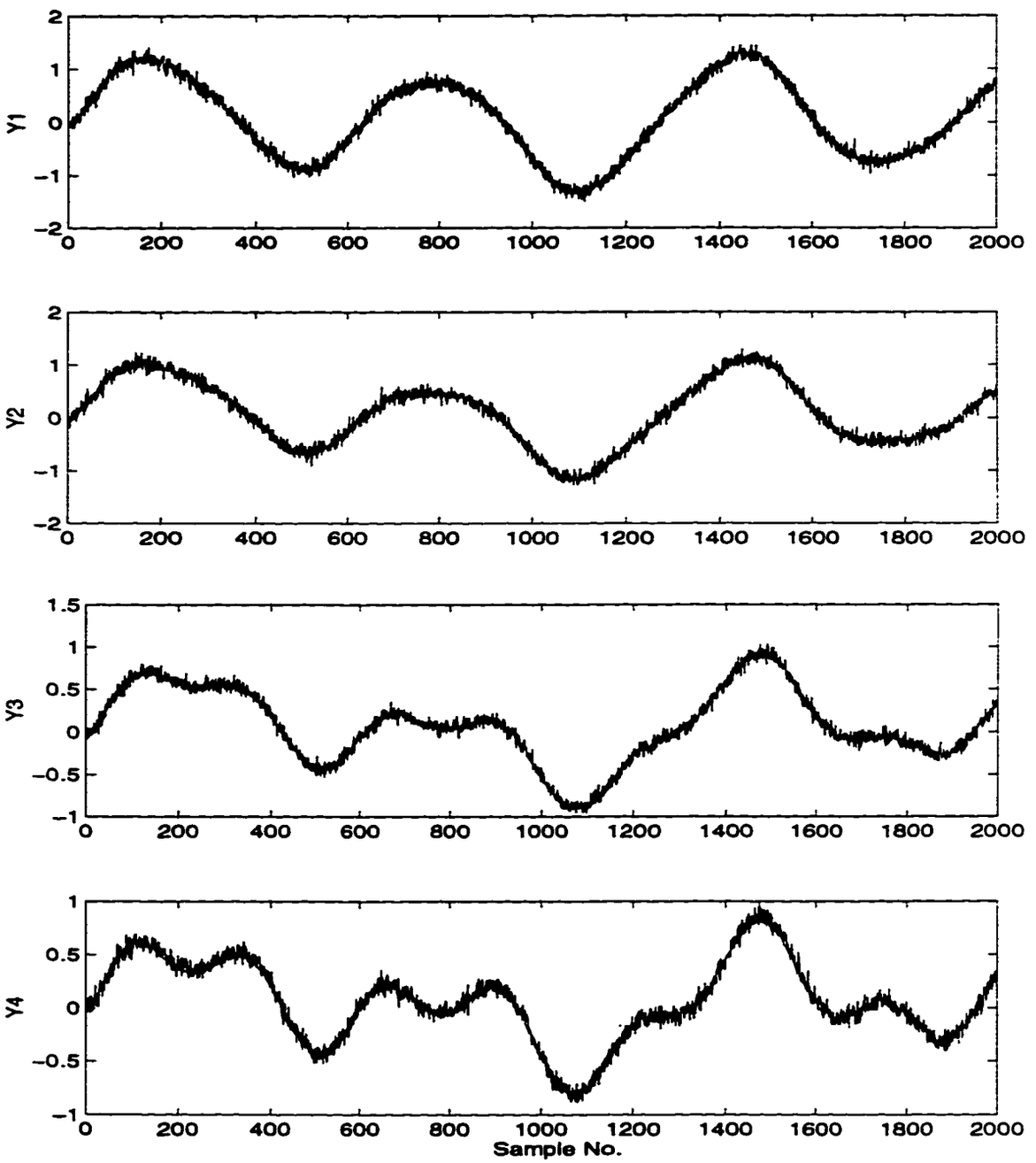

Figure 3.7: Cross validation for the heated rod system : Model (dashed line) and Actual Plant (solid line) responses

algorithm. As a first step, a dynamic model incorporating only linear elements was identified (details are not provided here). As expected, this model was not adequate in modelling the pH of the system. To model the nonlinearities in the system, a Hammerstein model was employed (using the SISO version of the algorithm presented in Lakshminarayanan *et al.* (1995)) to relate the input and output scores at each dimension. The identified dynamic PLS model is given below.

$$S_x = \begin{bmatrix} 1.1535 & 0 \\ 0 & 1.0122 \end{bmatrix} \tag{3.8}$$

$$S_y = \begin{bmatrix} 1.2875 & 0 \\ 0 & 1.3079 \end{bmatrix} \tag{3.9}$$

$$P = \begin{bmatrix} 0.7220 & 0.6796 \\ -0.6919 & 0.7335 \end{bmatrix} \tag{3.10}$$

$$R = \begin{bmatrix} 0.7336 & 0.6920 \\ -0.6797 & 0.7221 \end{bmatrix} \tag{3.11}$$

$$Q = \begin{bmatrix} 0.0347 & 1.0000 \\ -0.9994 & 0.0081 \end{bmatrix} \tag{3.12}$$

A closer look at the elements of Q indicate that the first PLS dimension essentially models the pH (output variable 2) of the system while the second PLS dimension models the level (output variable 1). This implies that the nonlinearities are confined to the first PLS dimension - so a Hammerstein model will be needed here. The second dimension can be modelled using only a linear dynamic element.

For the first PLS dimension, a Hammerstein model with *at least* a fourth order static polynomial was found necessary. However, for control purposes (this will be explained later), it is important that the order of the polynomial be odd and hence a fifth order polynomial was chosen to capture the static nonlinearities in the system.

The static nonlinearity (omitting the time variable) is given by :

$$t_1^* = 0.02t_1^5 + 0.1227t_1^4 - 0.0978t_1^3 - 0.5909t_1^2 + t_1 \tag{3.13}$$

The linear dynamic elements corresponding to the first and second dimensions are :

$$G_1 = \frac{0.1617z^{-1} - 0.0180z^{-2}}{1 - 0.8849z^{-1} + 0.0388z^{-2}} \tag{3.14}$$

$$G_2 = \frac{0.0458z^{-1} + 0.0522z^{-2}}{1 - 0.8744z^{-1} - 0.0601z^{-2}} \tag{3.15}$$

The results obtained using a Hammerstein model to relate the input and output scores at each dimension are presented in Figure 3.8. The model fit obtained using only linear inner

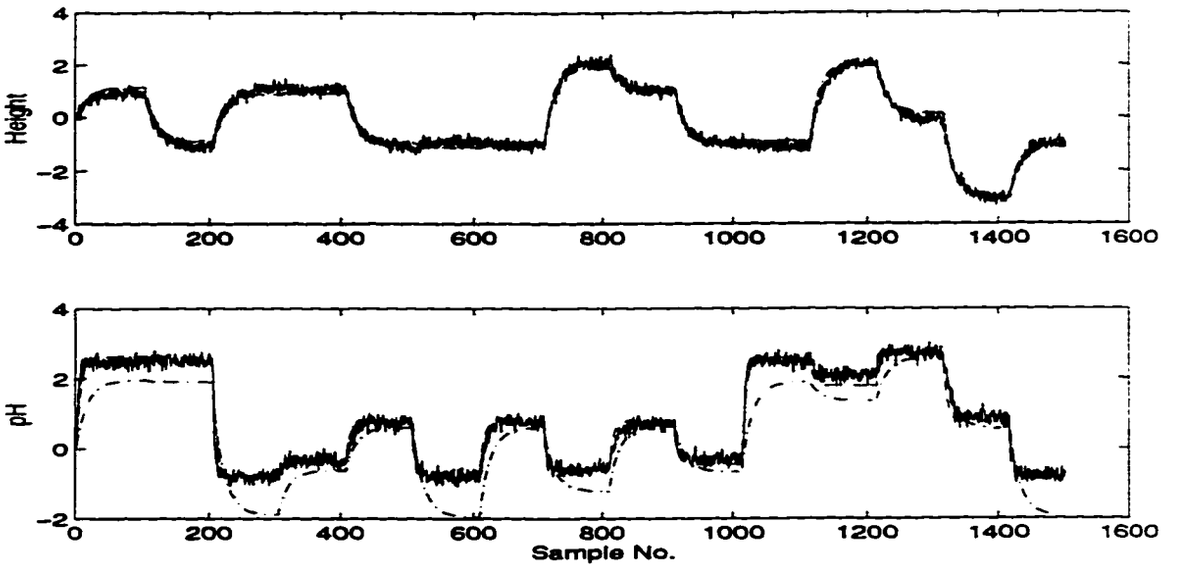

Figure 3.8: Comparison of Model fit for the acid-base neutralization system: PLS-Hammerstein model (dashed line), Linear model (dashed-dot) and Actual plant (solid line)

models is also shown. It is observed that the linear model is not able to capture the gain nonlinearities present in the pH measurements. The fit obtained using the Hammerstein inner model is excellent as is also evident from the cross validation run (Figure 3.9).

3.6 Process Control in the PLS Framework

Using linear dynamic PLS models, Kaspar and Ray (1992,1993) demonstrated a control strategy in which the PLS latent variables (T and U) are directly utilized in the synthesis of the control system. In this approach, the PLS matrices such as S_x, S_y, P and Q are employed as pre- and post-compensators on the plant. The Q matrix forms a basis for a space into which the scaled output variables are projected and the P matrix forms a basis onto which the scaled manipulated variables are projected. The controllers are designed independently based on the "inner" dynamic models identified at each dimension. Thus, the controller "sees" the error signals and the command signals in terms of the basis defined by the columns of the respective loading matrices (Q and P). Such a control strategy has a number of advantages. The process is somewhat decoupled owing to the orthogonality of the input scores and the rotation of the input scores to be highly correlated with the output scores. Controller design is simple - any theory available for SISO systems can be used. Because the dynamic part of the PLS model has a diagonal structure, the choice of the input-output pairing is automatic and is optimal in some sense. Infeasible setpoints (in terms of original variables) are not passed on to the controller because only the feasible part of the setpoint vector is retained after it is projected down to the latent variable subspace. This eliminates the problem of multi-loop controllers "fighting" each other in a vain bid to

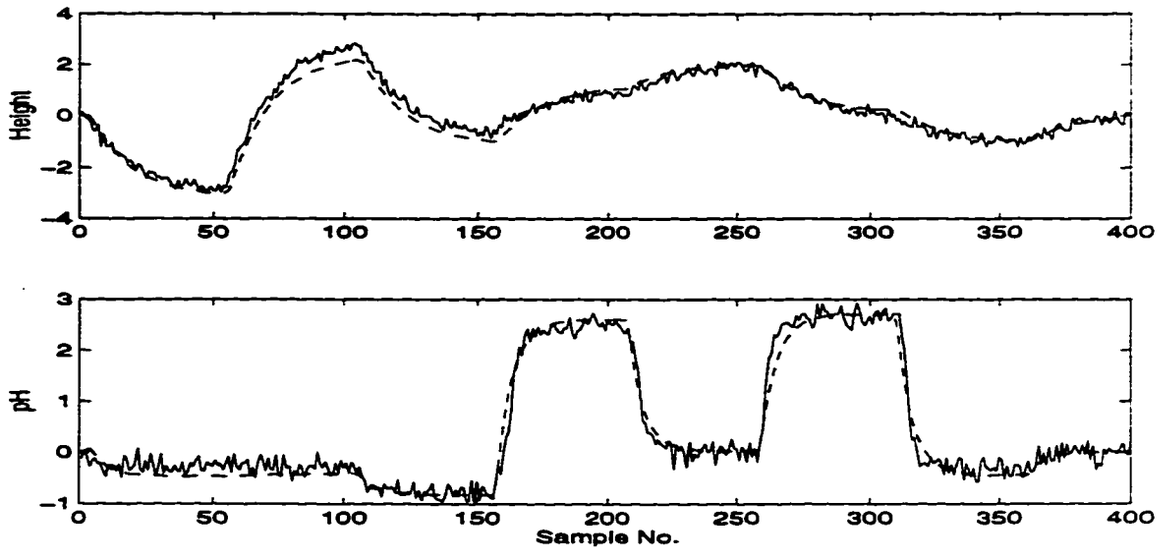

Figure 3.9: Cross validation with the identified PLS-Hammerstein model for the acid-base neutralization system : Model (dashed line) and Actual Plant (solid line)

reach an impossible setpoint. Due to the nature of the PLS model, nonsquare systems are readily handled.

The Kaspar-Ray scheme is shown in Figure 3.10. S_x and S_y are the diagonal scaling matrices determined prior to model identification. Q is the loading matrix for the Y block (output variables), Q^{-1} is the appropriate Moore-Penrose inverse of Q. P is the loading matrix for the input (X) block. E_Y is the error in terms of the original output variables. The *projected error* is E_U upon which the controllers act. The SISO controllers G_{C1} through G_{Cn} are designed based on the PLS inner models i.e., G_{Ci} is designed based on G_i (i = 1, ..., n) using any of the available alternatives (e.g. IMC, pole placement, frequency response techniques). T is the vector of scores computed by the controllers. The scores are then transformed into the real physical inputs which drive the process.

For the physical systems that are modelled by the Hammerstein structure, some modifications to the above scheme are necessary. As shown in Figure 3.11, blocks labelled RF_i (i = 1, ..., n) are included after each controller. The controllers are still designed based on the linear dynamic part of the Hammerstein model. Each RF_i is a root finding routine that is necessary to compensate for the static nonlinear part of the Hammerstein model.

Feedback control of linear systems is covered in detail by Kaspar and Ray (1992, 1993). Consequently, only the control of the nonlinear acid-base neutralization system will be illustrated using the ideas and results presented earlier.

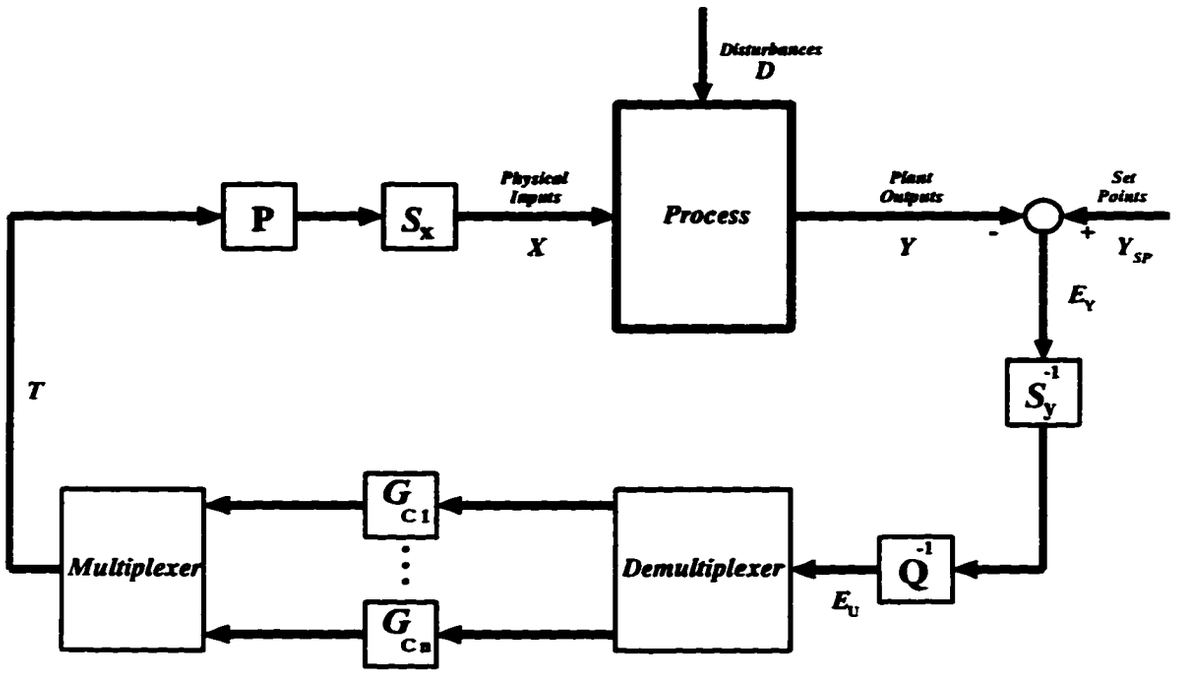

Figure 3.10: Feedback control using the PLS Framework : The Kaspar - Ray scheme for Linear Systems

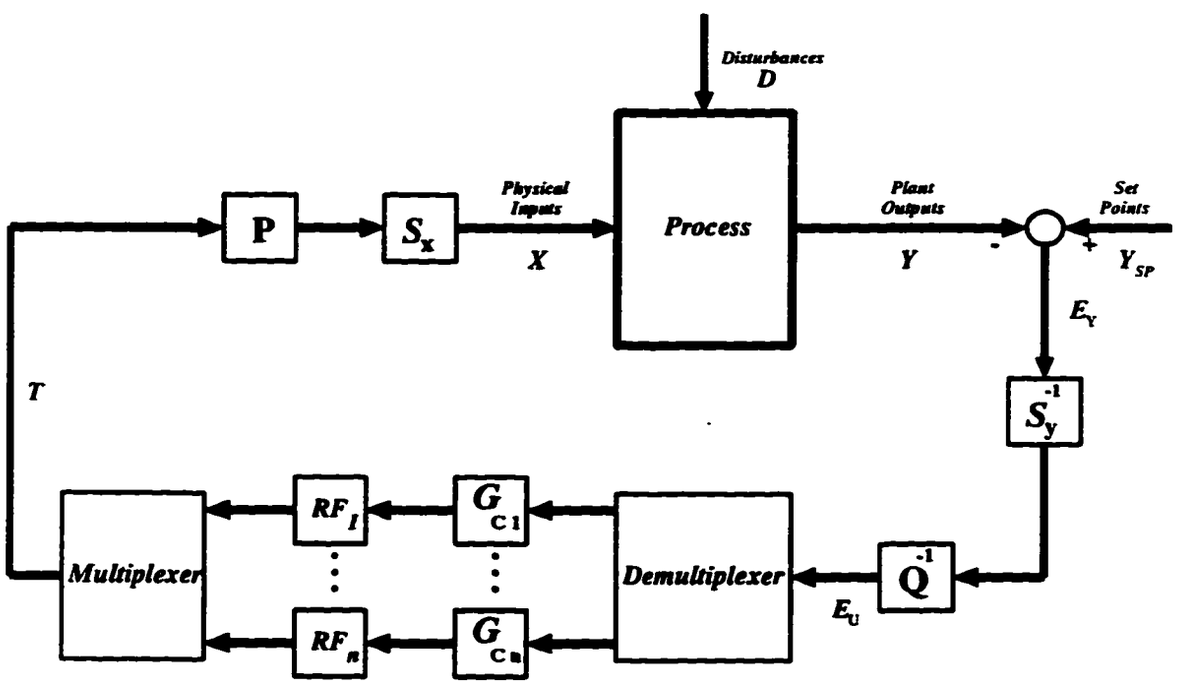

Figure 3.11: Feedback Control using the PLS Framework for systems modelled by the Hammerstein Structure

3.6.1 Control of the Acid-Base Neutralization Process

In a previous section, a model for the nonlinear pH system was identified. This model will now be used to control the two outputs (level and pH) by manipulating the acid and base flow rates. It is not reasonable to expect a single black-box model (linear or nonlinear) to characterize the steady-state and dynamic features of the process over the entire possible region of plant operation. In general, multiple models are to be identified for the different operating regions and the control strategy must effectively utilize these models. However, a single model will be employed here; its deficiencies in the control of the nonlinear process will be pointed out. Also, there exists a structural mismatch between the identified plant and the real process - this will manifest as a gap between the desired and achieved performance of the control system and may even make the closed loop system unstable.

The Vogel-Edgar algorithm (Vogel and Edgar, 1980) will be employed in this study to design the controllers G_{Ci}. This algorithm is superior to the minimal prototype controller (in terms of practical applicability) and the Dahlin algorithm (which could lead to the ringing phenomenon) and is ideal for a discrete second-order plus time-delay model. Besides, the Vogel-Edgar algorithm is more robust to modelling errors.

As a first step, it was examined if the linear model identified earlier would provide a satisfactory control for the process. Details of this linear model are not provided here because the closed loop control system shows sustained oscillations (Figure 3.12) for a set-point change of +1.5 in pH (which is within the pH values observed during the identification experiment) thereby establishing the inadequacy of the linear model.

We now utilize the identified nonlinear model as given in expressions (3.8) through (3.15). The control scheme is as shown in Figure 3.11 with n (the number of PLS dimensions) equal to 2. G_{C1} and G_{C2} are Vogel-Edgar controllers designed based on G_1 (equation (3.14)) and G_2 (equation (3.15)) respectively. The desired closed loop settling time is specified as five minutes for both the loops. To compensate for the static nonlinearity observed in the first PLS dimension, the output of controller G_{C1} is appropriately modified by using the root finding routine RF_1. The linear nature of the second PLS dimension implies that $RF_2 = 1$ and no modification needs to be performed on the output of G_{C2}.

Let the output of the controller G_{C1} be C_1. The output of RF_1 is determined by solving the roots of the polynomial (at each control interval),

$$0.02t_1^5 + 0.1227t_1^4 - 0.0978t_1^3 - 0.5909t_1^2 + t_1 = C_1 \tag{3.16}$$

Some remarks on the solution of the above equation is in order. To ensure that the polynomial has at least one real root, it is necessary that the order of the polynomial be odd. This explains why a fifth order polynomial was employed even though a fourth order polynomial provided good fit of the data. Moreover, several real roots may exist - the literature (Anbumani *et al.* (1981), Bhat *et al.* (1990)) suggests choosing the root with the smallest magnitude.

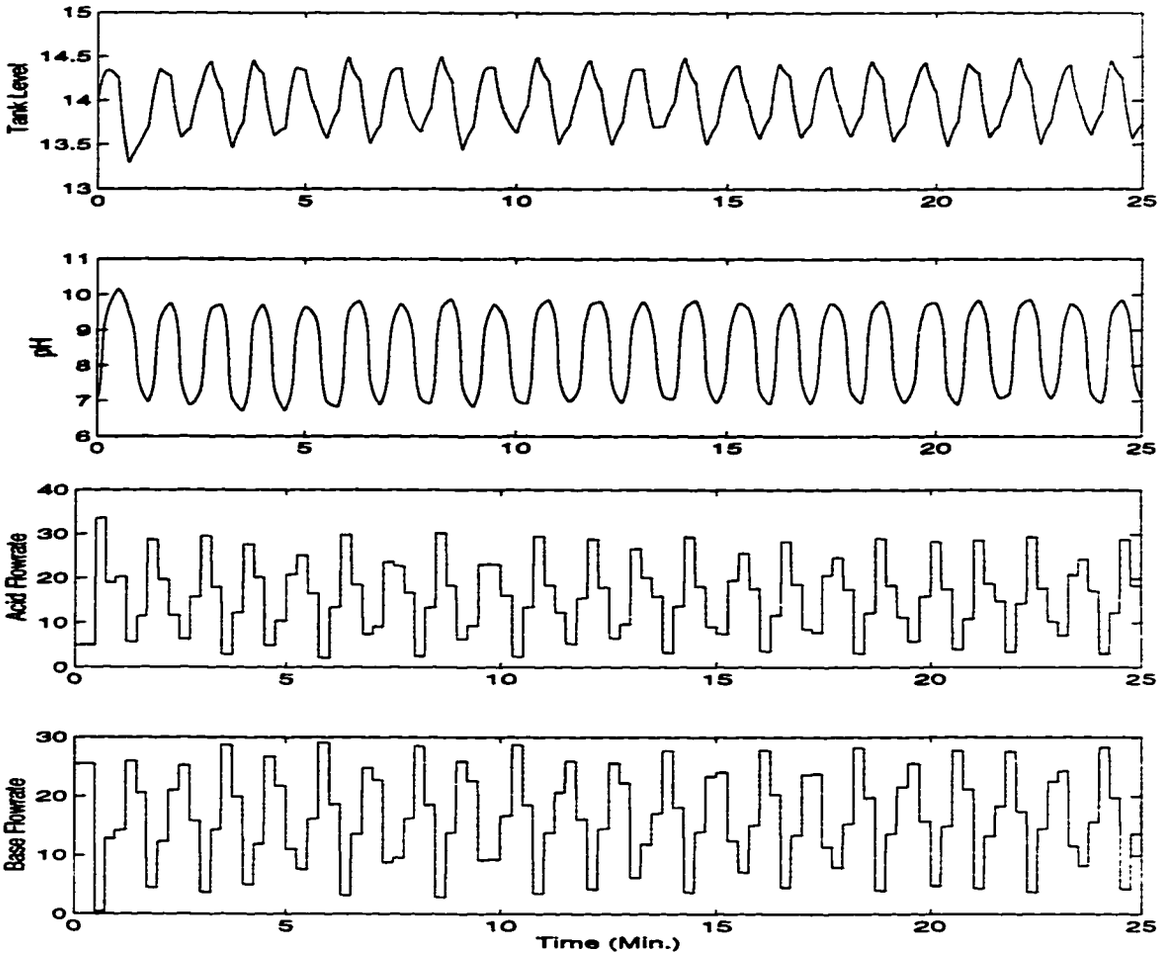

Figure 3.12: Response to a setpoint change in pH : Linear Model - Linear Controller

With the new control strategy in place, the setpoint in pH was changed from 7.06 to 8.5 with the level remaining at 14 cm. The closed loop response of the system is shown in Figure 3.13. The required pH value is reached within the desired time of five minutes. The tank level is only slightly disturbed. The control actions are acceptable and smooth.

Two more runs showing setpoint changes in both the level and pH are presented next. Figure 3.14 shows the response to changes in setpoint vector from [14 7.06] to (a) : [12 8.5] and (b) : [16 6.5]. The new setpoints are within the range of values of level and pH used in the identification experiment. These results indicate the utility of the Hammerstein inner model and the workability of the PLS based nonlinear control strategy. The control of the 2×2 neutralization system with the model identified in Chapter 2 (see Table 2.9) would not have been so straightforward.

The closed loop system is next analyzed for two extreme setpoint changes in level and pH. The new setpoint vectors are : [12 10] and [16 5]. In particular, the new pH values are outside the range of values considered in the identification experiment. From the simulation results (Figure 3.15). it is obvious that the control objectives are not met. This can be

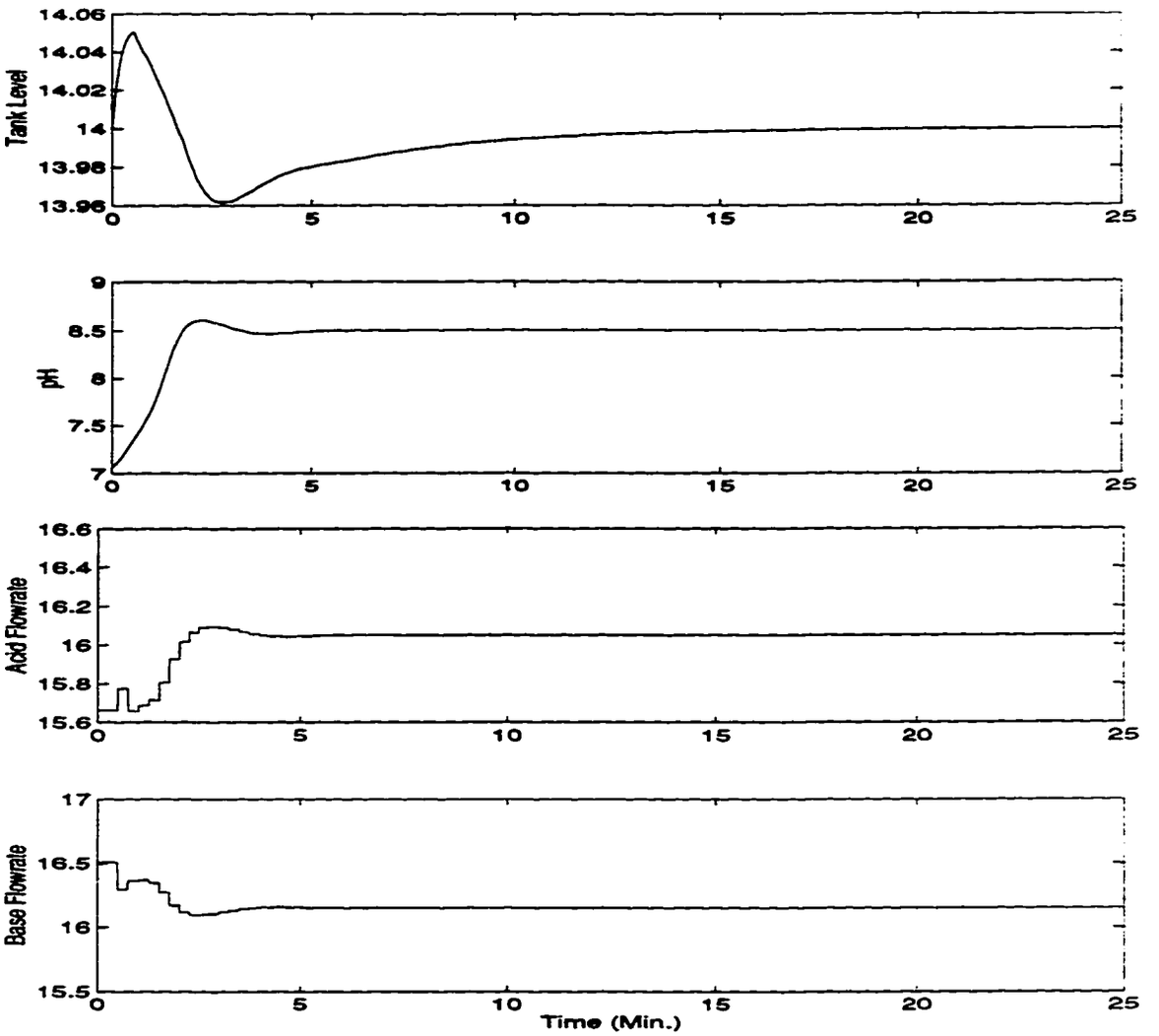

Figure 3.13: Response to a setpoint change in pH : Nonlinear Model - Nonlinear Controller

attributed to the fact that a single Hammerstein model is inadequate to describe the process behaviour over the entire range of operation and highlights the need for a multiple model or an adaptive framework.

The performance of the control system for two unmeasured buffer flow rate disturbances are shown in Figure 3.16. In case (a), the buffer flow rate was reduced from its nominal value of 0.6 ml/s to 0.2 ml/s. In case (b), the buffer flow rate was increased from 0.6 ml/s to 1.5 ml/s. Considering that the changes in buffer flow rate lead to large variations in the process gain, the performance of the control scheme is acceptable though the disturbance can be rejected faster by improved tuning. However, if the buffering capacity of the system is quite low, the control system exhibits unacceptable oscillatory behavior. An adaptive controller is required to provide better control performance over a wide range of buffering conditions.

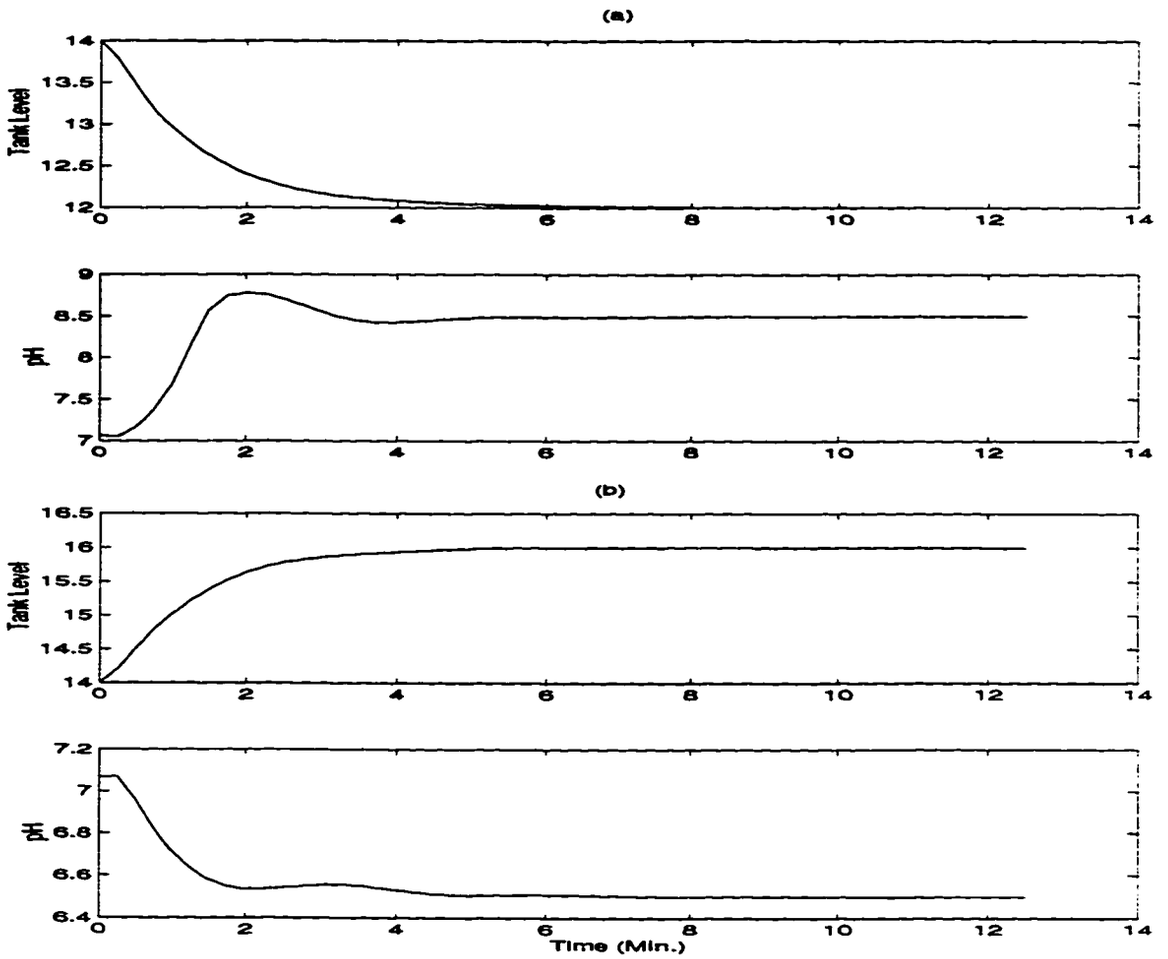

Figure 3.14: Response to two moderate setpoint changes in level and pH : Nonlinear Model - Nonlinear Controller

3.7 Multivariable Feedforward Control in the PLS Framework

One of the primary reasons for the control of industrial processes is to eliminate the effects of load disturbances. For disturbances that are *measured*, it is possible to design feedforward controllers that are capable of adjusting the manipulated variables before the controlled variables deviate from their setpoints. Usually feedforward control is never used by itself; it is effective when used in conjunction with feedback control that does not provide satisfactory control performance. Addition of stable feedforward control loops to existing feedback loops on a process does not affect the stability of the closed loop control system. Furthermore, and more importantly, the performance of the feedforward controller *does not* degrade significantly with modelling errors (Marlin, 1995).

Feedforward controller design for a SISO process depends on the models for the process and disturbance channels. The feedforward controller is the negative of the ratio of the dis-

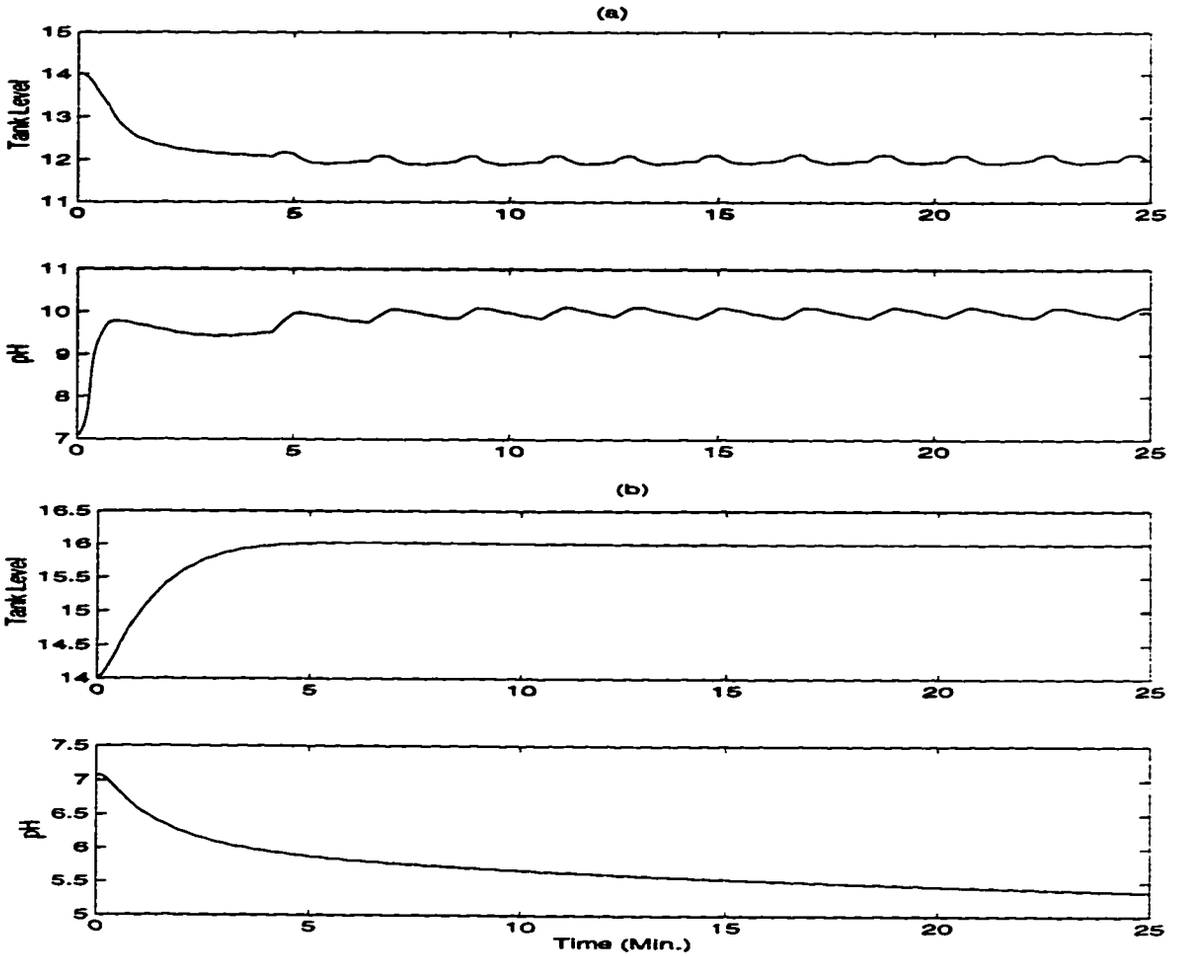

Figure 3.15: Response to two extreme setpoint changes in level and pH : Nonlinear Model - Nonlinear Controller

turbance transfer function to the process transfer function. The SISO feedforward controller is usually realized as a lead/lag element. Sometimes, even the lead/lag and time delay elements are ignored and a steady state feedforward controller is employed. The literature is relatively scarce as far as multivariable feedforward controllers are concerned. Shen and Yu (1992) discuss the concept of indirect feedforward control. In their design, fast secondary measurements are used to infer the load changes and secondary controllers are designed to cancel the effect of these load disturbances on the process outputs. A certain *interaction measure array* (Γ) is defined and is used to design the secondary controllers for quick rejection of specific disturbances. This method is applicable only when some secondary process measurements are available. Stanley *et al.* (1985) proposed the *relative disturbance gain* (RDG) to compare the disturbance rejection capabilities of the multi-loop SISO controllers versus the inverse based multivariable controllers such as the decoupler. Shen and Yu (1992) discuss the relationship between Γ and RDG.

Under the assumption that only primary outputs are available for control, a new strategy

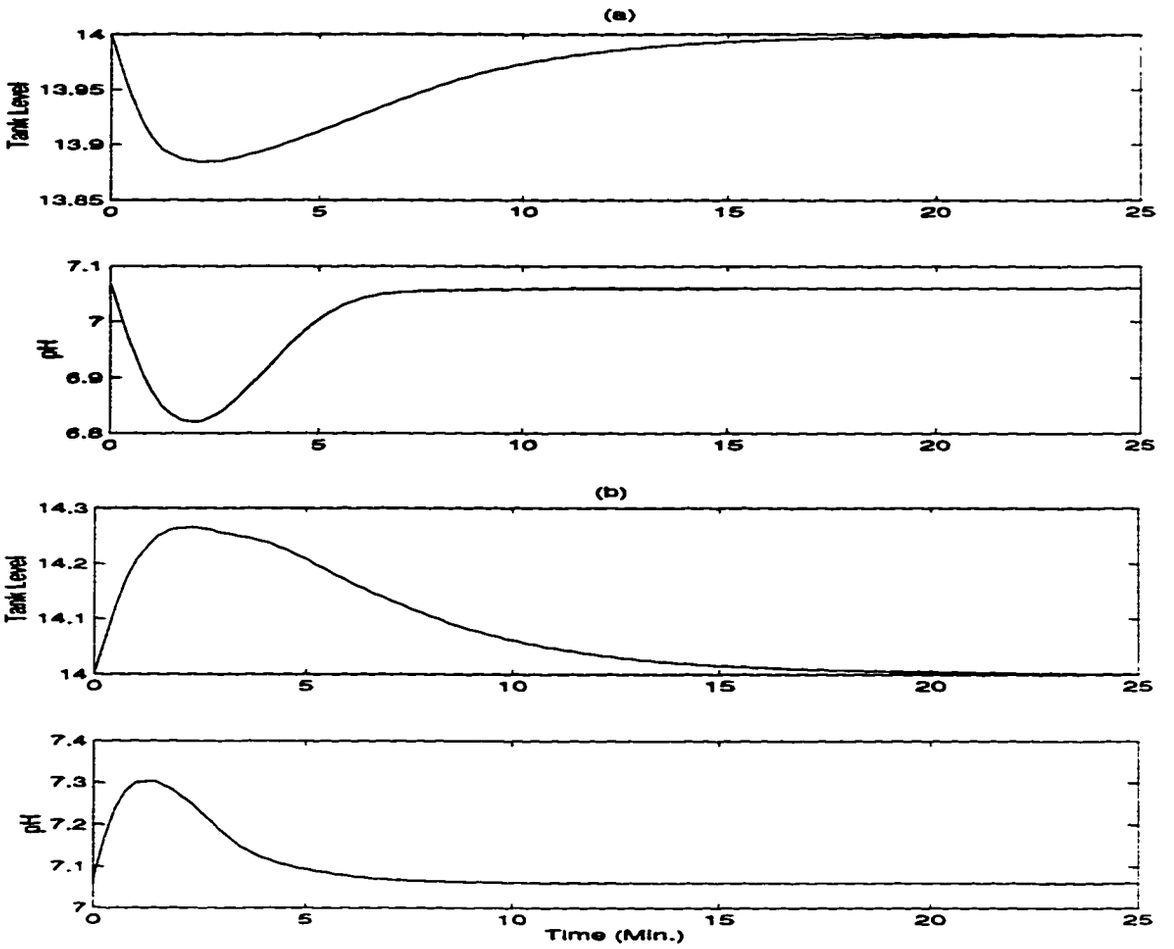

Figure 3.16: Regulatory response to two step changes in the buffer flow rate : (a) 0.6 ml/s $\rightarrow$ 0.2 ml/s (b) 0.6 ml/s $\rightarrow$ 1.5 ml/s using the nonlinear model and the nonlinear controller

for the design of a multivariable feedforward controller is proposed. Each element of this multivariable controller is realizable as a ratio of two simple transfer functions - this permits retaining the simplicity and elegance of the SISO feedforward design approach. As with SISO feedforward design, the multivariable feedforward controller can be implemented using either lead/lag elements with time delay or just pure gain elements.

First of all, it is assumed that a dynamic PLS model similar to the one between the manipulated inputs (X) and controlled outputs (Y) is available to describe the relationship between measured disturbances (D) and the controlled outputs (Y). The disturbance-output PLS model (with 'm' PLS dimensions) is characterized by : (i) the diagonal scaling matrices for the output and disturbance spaces - W_y and S_d (ii) the matrices containing the weights attached to the output and disturbance variables in each dimension - Q^d and R^d and (iii) the diagonal matrix G^d containing the dynamic relationship between D and Y - each diagonal element of G^d is denoted by G_j^d (j = 1. .... m).

The relationship between the manipulated inputs and the controlled outputs can be

summarized by (refer Figure 3.1) :

$$Y_x = S_y Q G \left\{ X S_x^{-1} R \right\}' \tag{3.17}$$

In a similar manner, the relationship between the measured disturbances and the controlled outputs is given by :

$$Y_d = W_y Q^d G^d \left\{ D S_d^{-1} R^d \right\}' \tag{3.18}$$

To offset the effect of the measured disturbances on the controlled outputs. the required change in the manipulated inputs must be determined. This is done by setting $-Y_d = Y_x$ as follows :

$$-W_y Q^d G^d \left\{ D S_d^{-1} R^d \right\}' = S_y Q G \left\{ X S_x^{-1} R \right\}' \tag{3.19}$$

Recognizing that the scores in the manipulated input space (T) and the disturbance variable space (T^d) are defined by $T = X S_x^{-1} R$ and $T^d = D S_d^{-1} R^d$ respectively. equation (3.19) can be rewritten as

$$-W_y Q^d G^d T^{d'} = S_y Q G T' \tag{3.20}$$

Since the PLS based control is based on the scores rather than the original variables. the input scores (T) must be expressed in terms of the disturbance scores (T^d). Defining $\Lambda = S_y Q$ and $\Omega^d = W_y Q^d$, the above equation can be expressed as

$$-\Omega^d \begin{bmatrix} G_1^d & 0 & 0 & \cdots & 0 \\ 0 & G_2^d & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & G_m^d \end{bmatrix} T^{d'} = \Lambda \begin{bmatrix} G_1 & 0 & 0 & \cdots & 0 \\ 0 & G_2 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & G_n \end{bmatrix} T' \tag{3.21}$$

A rearrangement of the above equation gives

$$T' = -\left\{ \Lambda \begin{bmatrix} G_1 & 0 & 0 & \cdots & 0 \\ 0 & G_2 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & G_n \end{bmatrix} \right\}^{\dagger} \Omega^d \begin{bmatrix} G_1^d & 0 & 0 & \cdots & 0 \\ 0 & G_2^d & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & G_m^d \end{bmatrix} T^{d'} \tag{3.22}$$

with a further simplification yielding the design equation for the multivariable feedforward controller (in the latent space) as :

$$T' = \underbrace{-\begin{bmatrix} G_1 & 0 & 0 & \cdots & 0 \\ 0 & G_2 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & G_n \end{bmatrix}^{-1} \Lambda^{\dagger} \Omega^{d} \begin{bmatrix} G_1^d & 0 & 0 & \cdots & 0 \\ 0 & G_2^d & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & G_m^d \end{bmatrix}}_{\textit{Feedforward Controller, FFC}} T^{d'} \tag{3.23}$$

Element [i,j] (i = 1, .. , n ; j = 1, .. , m) of the matrix FFC can be written as (using Cramer's rule for square and nonsquare systems; see Appendix B) :

- Case (a) : Q is nonsquare (ny < n)

$$FFC_{ij} = -\frac{G_j^d}{G_i} \frac{[\det(Q_{ij}^{*} Q') - \det(Q_i^{*} Q_i^{*'})]}{\det(QQ')} \tag{3.24}$$

- Case (b) : Q is square (*i.e.* ny = n)

$$FFC_{ij} = -\frac{G_j^d}{G_i} \frac{\det(Q_{ij}^{*})}{\det(Q)} \tag{3.25}$$

- Case (c) : Q is nonsquare (ny > n)

$$FFC_{ij} = -\frac{G_j^d}{G_i} \frac{\det(Q' Q_{ij}^{*})}{\det(Q' Q)} \tag{3.26}$$

In the above expressions, the matrix Q_{ij}^{*} is obtained by replacing the i^{th} column in matrix Q by the weighted j^{th} column of Q^d. Matrix Q_i^{*} obtained by simply deleting the i^{th} column in matrix Q.

$$Q_{ij}^{*} = [q_1 \mid q_2 \mid \cdots \mid q_{i-1} \mid W_y S_y^{-1} q_j^d \mid q_{i+1} \mid \cdots \mid q_n] \tag{3.27}$$

$$Q_i^{*} = [q_1 \mid q_2 \mid \cdots \mid q_{i-1} \mid q_{i+1} \mid \cdots \mid q_n] \tag{3.28}$$

It is seen that each element of the multivariable feedforward controller can be expressed as a ratio of two transfer functions multiplied by a constant. This makes the design of the multivariable feedforward control simple and elegant. If only a steady state feedforward compensation is sought, the dynamic components in equation (3.23) can be replaced with their respective steady state gains. The combined feedback plus feedforward control strategy for linear systems is shown in Figure 3.17.

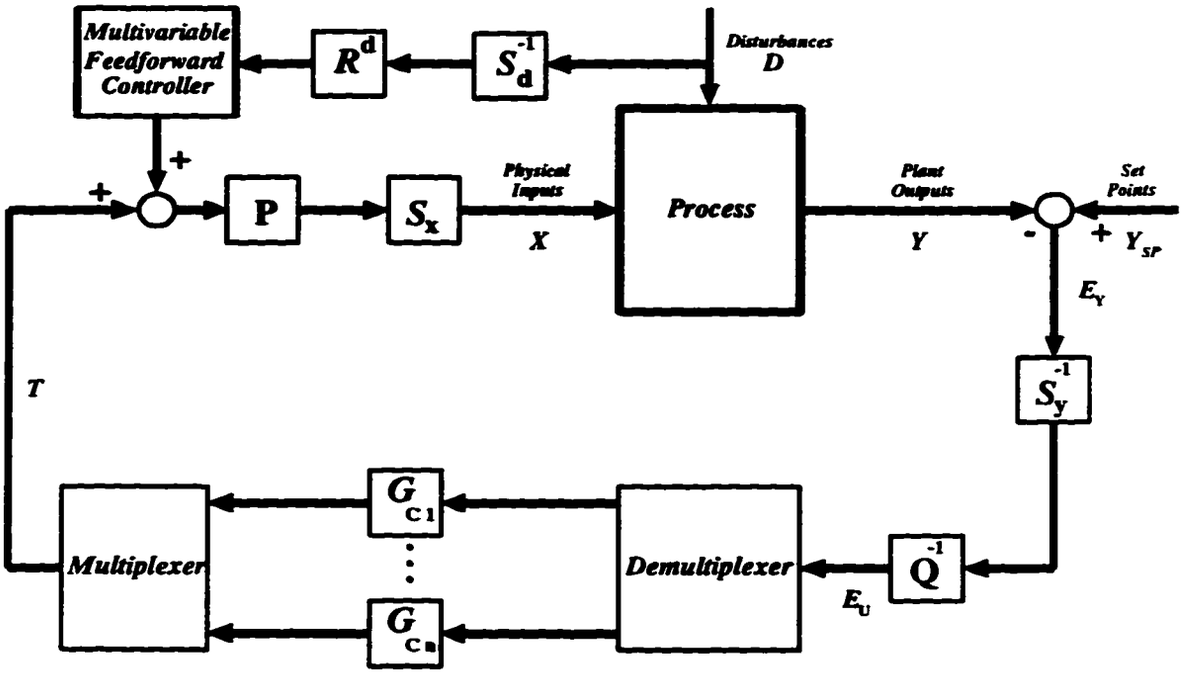

Figure 3.17: The combined feedback - feedforward control strategy for linear systems : The PLS Framework

3.7.1 Feedforward Control of the Wood-Berry Column

The following PLS model was obtained for the Wood-Berry column by collecting the disturbance-output data keeping the manipulated inputs stationary.

$$S_d = \begin{bmatrix} 0.2232 & 0 \\ 0 & 1.9753 \end{bmatrix}; W_y = \begin{bmatrix} 1.0012 & 0 \\ 0 & 1.1466 \end{bmatrix}$$

$$P^d = \begin{bmatrix} 0.8233 & 0.4818 \\ 0.5782 & -0.8763 \end{bmatrix}; R^d = \begin{bmatrix} 0.8763 & 0.5782 \\ 0.4818 & -0.8233 \end{bmatrix}; Q^d = \begin{bmatrix} 0.6949 & 0.2051 \\ 0.7191 & 0.9787 \end{bmatrix}$$

$$G_1^d = \frac{0.0485z^{-6} + 0.0593z^{-7}}{1 - 0.4679z^{-1} - 0.4516z^{-2}} \quad and \quad G_2^d = \frac{0.0591z^{-6}}{1 - 0.3793z^{-1} - 0.3700z^{-2}}$$

Using the above PLS model and the one obtained earlier (between the manipulated inputs and the controlled outputs), the multivariable feedforward control law is implemented on the Wood-Berry column. The scores for the manipulated inputs computed by the feedforward control law are added to those computed by the feedback controller to obtain a *combined* feedback-feedforward control action. Regulatory control for two step disturbances in the feed flow rate (-0.35 units at t=0) and the feed composition (-3 units at t=125 minutes) are shown in Figure 3.18.

From the ISE values reported in Table 3.1, it is evident that the variations in product quality can be considerably reduced by incorporating feedforward control for these measured

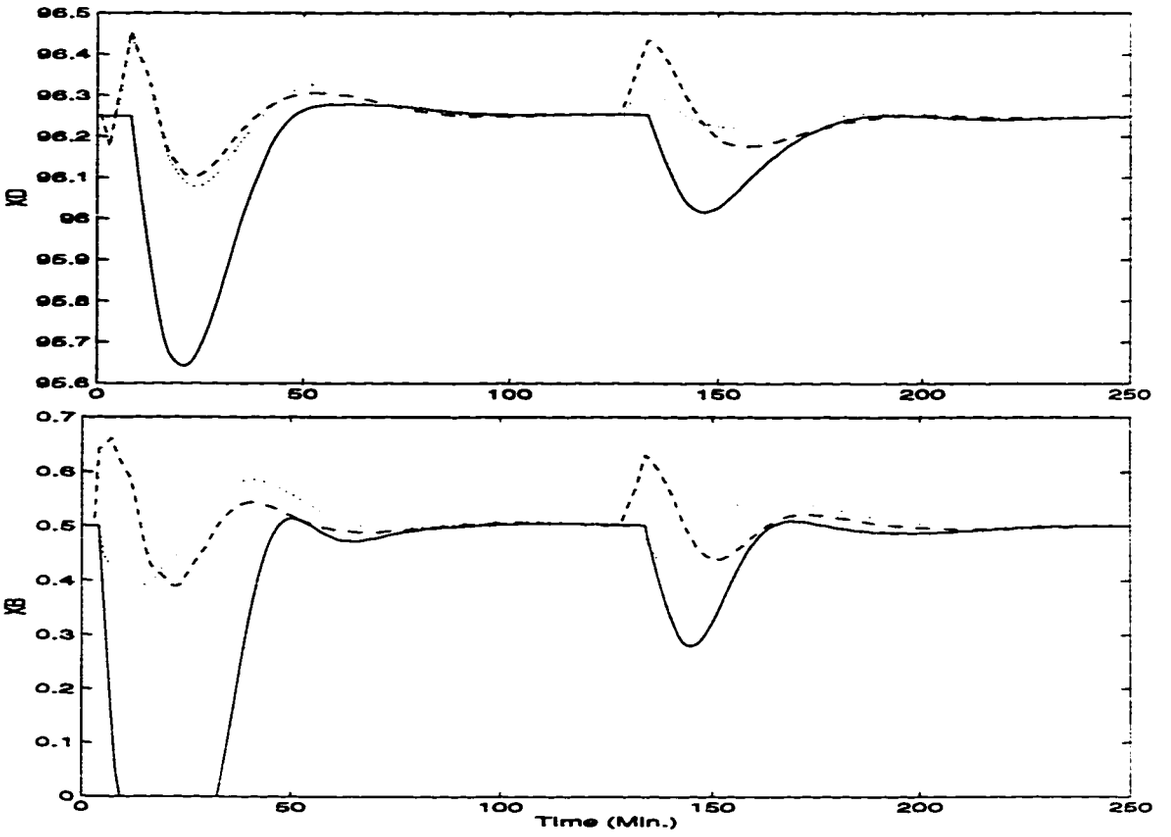

Figure 3.18: Regulatory Control of the Wood-Berry Column to a step change of -0.35 units in feed flow rate (at t=0) and a step change of -3 units in feed composition (at t=125 minutes) . Feedback control only (solid line), Feedback *plus* steady state feedforward control (dashed line) and Feedback *plus* dynamic feedforward control (dotted line)

disturbances. The ISE values also indicate that, in the presence of model-plant mismatch (as is the case here), a dynamic feedforward controller may not always provide a significantly better performance compared to the more easily implemented steady state feedforward controller. Considerable improvements in control were noticed by implementing only a single feedforward controller FFC_{11}. This is because the first PLS dimension captures the majority of the variations in the process and disturbance channels.

Table 3.1: Summary of ISE values : Wood-Berry Column

Controlled Variable	ISE Values		
	FB only	FB + Steady State FF	FB + Dynamic FF
X_D	7.0454	0.8304	0.6791
X_B	7.9183	0.4203	0.3618

3.7.2 Feedforward Control of the Acid-Base Neutralization Tank

In this section, the development of the feedforward control strategy for the acid-base neutralization system is described. The buffer flow rate is assumed to be the major measurable disturbance. Instead of employing a modified version of the linear feedforward control law (*cf.* equation (3.23)), the structure of the model identified for this system (i.e., equations (3.8) through (3.15)) will be utilized. For this system, it was shown that the first PLS dimension models the pH and the second dimension models the level. This implies that once the models relating the buffer flow rate to the pH and level are obtained, two feedforward controllers can be synthesized - one for each dimension.

To obtain the models relating the buffer flow rate to the level and pH, the buffer flow rate was perturbed about its nominal value of 0.6 ml/s by ±0.4 ml/s. During this "experiment", the acid and base flow rates were regulated at their nominal value of 16.6 ml/s and 15.6 ml/s respectively. This open loop data is used to construct the models.

The buffer flow rate *versus* pH relationship was modelled by the following Hammerstein model. Denoting the buffer flow rate as 'd', the static nonlinearity was identified as

$$d^{*} = 0.0389d^3 - 0.6423d^2 + d \tag{3.29}$$

The linear part of the Hammerstein model is

$$G_1^d = \frac{0.1871z^{-1} - 0.0672z^{-2}}{1 - 0.8817z^{-1} + 0.0248z^{-2}} \tag{3.30}$$

Note that the above linear model relates the transformed buffer flow rate (d^{*}) to pH. The model fit and the cross validation run using this Hammerstein model are shown in Figure 3.19 - the model appears to capture the relationship to a good measure.

The linear model relating the buffer flow rate to the level is :

$$G_2^d = \frac{0.1761z^{-1} - 0.1199z^{-2}}{1 - 1.0907z^{-1} + 0.1341z^{-2}} \tag{3.31}$$

The validity of the identified model is exemplified by the model fit and cross validation results shown in Figure 3.20.

Having obtained convincing models, the feedforward controllers can now be developed.

- Feedforward controller for buffer flow rate - level subsystem

  Using the information flow diagram (Figure 3.1), the PLS model (equations (3.8) through (3.15)), equation (3.31) and the fact that the second PLS dimension essentially models the level, the following equation is obtained :

$$Y_1 = Q_{12}\, sy_1\, G_2\, t_2 + G_2^d\, d \tag{3.32}$$

  with Q_{12} and sy_1 denoting the usual elements in the Q and S_y matrices respectively.

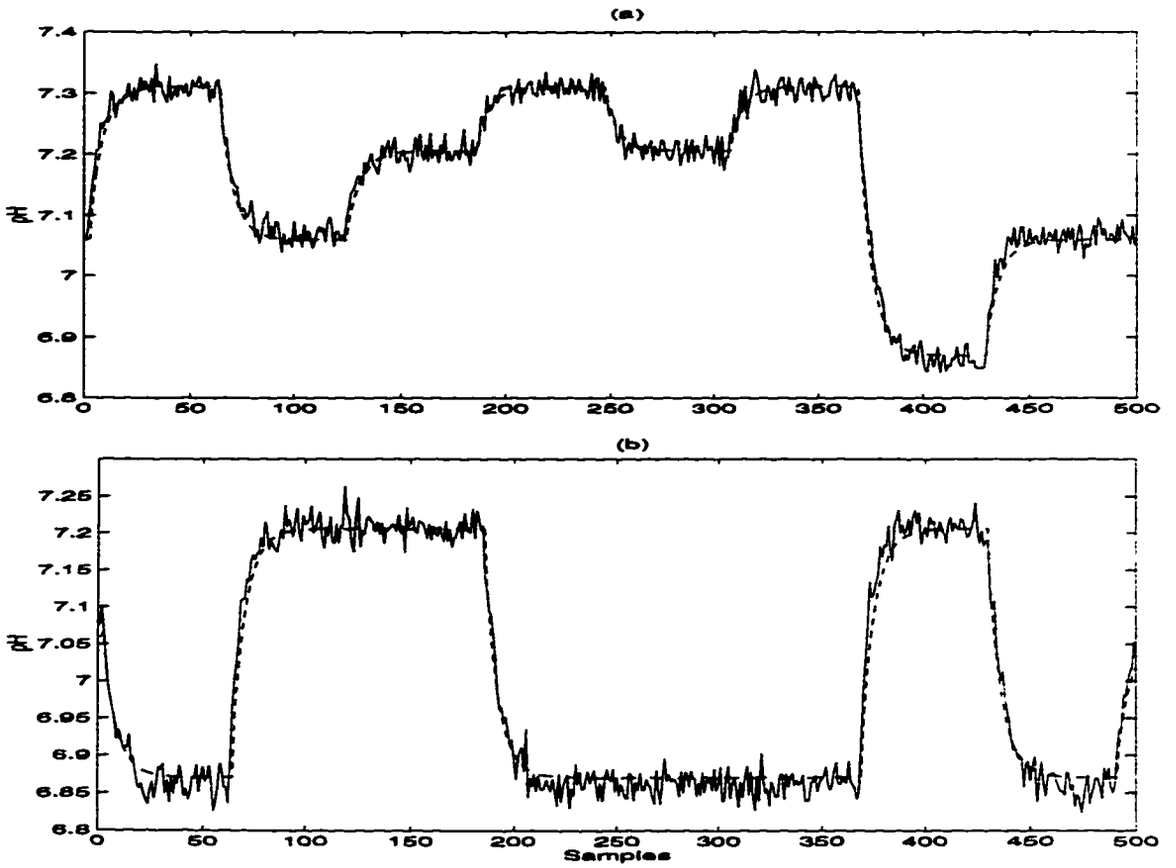

Figure 3.19: (a) Model fit and (b) cross validation for the buffer flow rate vs. pH relationship - Actual pH (solid line) and Model predictions (dashed line)

Setting the deviation variable Y_1 equal to zero in the above expression, the feedforward controller is derived as

$$t_2 = -\left(\frac{1}{Q_{12}\, sy_1}\right) \frac{G_2^d}{G_2} d \tag{3.33}$$

The poles of this feedforward controller were found to lie outside the unit circle - this will result in an unstable closed loop. The implementation of this controller was therefore restricted to a steady state design. The feedforward control action computed via equation (3.33) is superimposed on the output of the feedback controller G_{C2}.

- Feedforward controller for buffer flow rate - pH subsystem

  The development of this feedforward controller is similar to that presented above. Now, the first PLS dimension of the model identified in equations (3.8) through (3.15) must be examined. Employing equations (3.29) and (3.30), we obtain

$$Y_2 = Q_{21}\, sy_2\, G_1\, t_1^* + G_1^d\, d^* \tag{3.34}$$

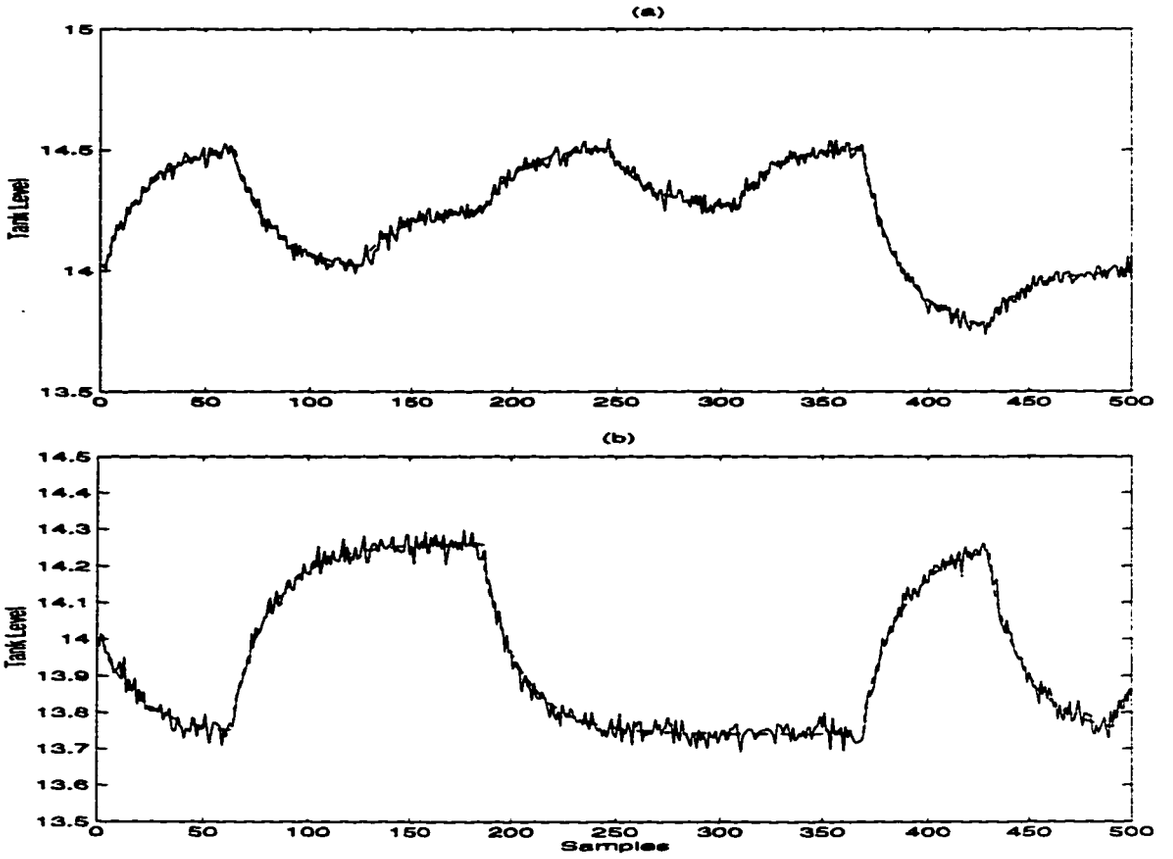

Figure 3.20: (a) Model fit and (b) cross validation for the buffer flow rate vs. level relationship - Actual level (solid line) and Model predictions (dashed line)

giving the following feedforward controller.

$$t_1^* = -\left(\frac{1}{Q_{21}\, sy_2}\right) \frac{G_1^d}{G_1} d^* \tag{3.35}$$

The measured value of the buffer flow rate (d) is first transformed (using equation (3.29)) to d^*. The feedforward control action (t_1^*) is computed using equation (3.35) and is added to the output of controller G_{C1}.

Comparison of the combined feedback-feedforward control strategy with the feedback control strategy for the same type of disturbances considered earlier is presented in Figure 3.21. The ISE values for the two control strategies are presented in Table 3.2. The change in the ISE values for level is dramatic - the ISE values for the combined feedback-feedforward control strategy is only about 1% of that obtained with feedback only control. Due to the model-plant mismatch, improvements obtained in the ISE values for pH was restricted to about 80-90%.

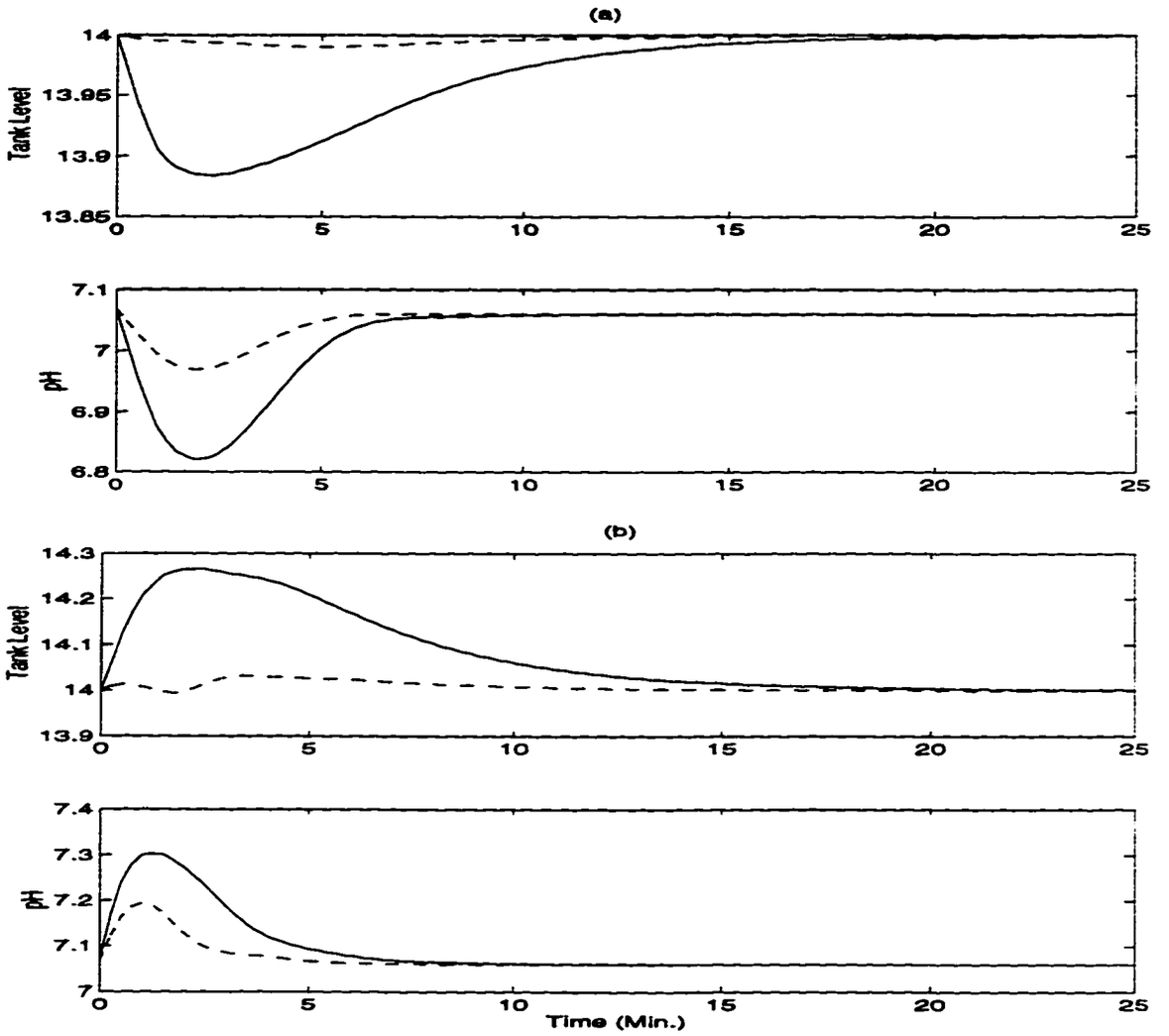

Figure 3.21: Regulatory response to two step changes in the buffer flow rate : (a) 0.6 ml/s → 0.2 ml/s (b) 0.6 ml/s → 1.5 ml/s using feedback control only (solid line) and a combined feedback-feedforward control strategy (dashed line)

Table 3.2: Summary of ISE values : Acid-Base Neutralization System

	ISE Values (Level)		ISE Values (pH)	
	FB only	FB + FF	FB only	FB + FF
Step change of -0.4 units in buffer flow rate	0.2645	0.0020	0.6027	0.0749
Step change of +0.9 units in buffer flow rate	1.4110	0.0154	0.5141	0.0995

3.8 Conclusions

Using well established analysis and design tools available for SISO systems, a PLS based framework was presented for the modelling and control of multivariable systems. A multivariable feedforward control strategy with a simple structure is also proposed and incorporated in the latent subspace. The results indicate that the approach may be applicable to a broad class of systems including those with nonlinear characteristics.

The control of the nonlinear acid-base neutralization system was relatively simple since a Hammerstein structure was found to adequately model the PLS inner relationships. For practical applications, it is important to include constraint handling capabilities to the control system by implementing advanced model predictive control (MPC) schemes using the dynamic PLS models identified with the proposed approach. This is the subject matter of our next chapter.

Chapter 4

Model Based Predictive Control using Dynamic PLS Models

4.1 Overview

Conventional digital control of multivariable linear and nonlinear systems was dealt with in the previous chapter. However, for practical applications it is necessary to incorporate constraint handling capability in the control scheme. Here, the dynamic PLS models identified using the strategy outlined in the previous chapter is employed in the implementation of advanced model based predictive controllers. Issues in the implementation of constrained control schemes are discussed. Identification and constrained control of a laboratory stirred tank heater (discussed in chapter 2) using the PLS framework are demonstrated along with other simulation examples involving linear and nonlinear systems.

[1]Sections of this chapter have been submitted for possible publication/presentation as :

1. S. Lakshminarayanan, Rohit S. Patwardhan, Sirish L. Shah and K. Nandakumar, "A Dynamic PLS Framework for Advanced Process Control". Submitted to the IFAC ADCHEM '97 Meeting. August 1996.
2. S. Lakshminarayanan, Sirish L. Shah and K. Nandakumar, "A Case Study of Nonlinear Modelling and Control using PLS". Submitted to the Chemical Engineering issue of the Journal of Institute of Engineers, Singapore. November 1996.

4.2 Contributions of this chapter

- Use of the advanced model based predictive control algorithms in the dynamic PLS framework is illustrated. Improvement in control performance and the ability to handle process constraints are direct consequences of this approach. The control calculations are simpler owing to the diagonal structure of the inner process model.
- Constrained control of the pH neutralization system is demonstrated using a Wiener type model (nonlinear static element following a linear dynamic element) in the inner model of the dynamic PLS framework.
- The first real-time application (to the best of the authors' knowledge) of the PLS compensation strategy is reported in this chapter. Input-output data from a laboratory stirred tank heater is used to obtain a dynamic PLS model which is used to illustrate a DMC application operating in the PLS latent space.

4.3 Introduction

Model Predictive Control (MPC) schemes are gaining increasingly wide acceptance in the chemical process industries. Various forms of model predictive control schemes have been reported in the last decade owing to great interest evinced both by industrial practitioners and academic researchers. The techniques include : IDentification and COMmand algorithm (IDCOM) (Richalet *et al.*, 1978), Dynamic Matrix Control (DMC) (Cutler and Ramaker, 1980), Model Algorithmic Control (MAC) (Rouhani and Mehra, 1982), Internal Model Control (IMC) (Garcia and Morari, 1982), Extended Horizon Adaptive Control (EHAC) (Ydstie, 1985), Multivariable Optimal Constrained Control Algorithm (MOCCA) (Sripada and Fisher, 1985) and Generalized Predictive Control (GPC) (Clarke *et al.*, 1987). Several successful industrial applications have been reported for most of the above algorithms.

Besides their ability to handle large multivariable (square or nonsquare) systems, the key feature that makes these algorithms attractive for industrial applications is their ability to handle constraints on the process variables. To achieve higher profits, the supervisory control layer often forces the process to operate at the intersection of constraints. Rather than adding ad-hoc fixups to unconstrained control laws, the MPC schemes incorporate the constraints explicitly in the control design strategy by solving a constrained optimization problem at each sampling instant in a receding horizon fashion. A variety of process descriptions (mathematical models) ranging from the step/impulse response coefficients, discrete transfer function models to state space models are employed by these MPC schemes - each having its merits and drawbacks.

In this chapter, the PLS based multivariable modelling strategy proposed in the previous chapter is combined with one of the popular and powerful MPC schemes - the DMC algorithm. Since DMC represents a mature control algorithm with several published and

proprietary applications, only a brief description will be provided. This will be followed by a section dealing with the mapping of the original process constraints into the latent space constraints - this section may be regarded as the core material of this chapter. The mapping is relatively straightforward but the material presented here is intended to alert the reader to some of the possible problem formulations. Process applications involving a mix of simulation examples (Wood-Berry column and the acid-base neutralization tank) and a laboratory stirred tank heater will be discussed in the final section.

4.4 An overview of Dynamic Matrix Control

The DMC algorithm has been extensively described in the literature (García *et al.* (1990)). Utilizing step response data, the DMC algorithm is designed on the basis of a multistep objective function subject to input amplitude, rate and output constraints. Usually, at each sampling instant, several control moves are computed but only the first control move is implemented thus imparting it a receding horizon character. The basic ideas of the DMC algorithm can be summarized as follows :

1. Predict future response of the process using the identified model
2. Compute appropriate control actions that results in the minimization of an objective (cost) function subject to process constraints.

   The objective function is usually a function of : (1) the deviation of plant outputs from their targets over the prediction horizon (N_2) and (2) the weighted control action over the control horizon (N_x). The constraints relate to the maximum and minimum values of the manipulated variable and the rate of change of the manipulated variable moves. It is not uncommon to have constraints on the output variables so as to maintain the product quality within a desired range.
3. Implement only the first computed control move. Repeat, steps 1 through 3 at each sampling instant.

Assume that there are 'nx' manipulated inputs and 'ny' controlled outputs. Let $\underline{r}$ denote the setpoint vector over the prediction horizon and $\underline{\hat{y}}$ denote the predicted outputs (using the model and the current process measurements) over the same horizon. With $\Delta\underline{x}$ representing the N_x incremental future moves, we can write the objective function in compact vector notation as :

$$J = (\underline{r} - \underline{\hat{y}})^T \Gamma (\underline{r} - \underline{\hat{y}}) + \Delta\underline{x}^T \Lambda \Delta\underline{x} \tag{4.1}$$

with Γ and Λ representing the output and input weighting matrices (which are positive definite and usually of a diagonal structure) and

$$\underline{r} = [r_1(k+1)\ r_2(k+2)\ \cdots\ r_{ny}(k+1)\ \cdots\ \cdots\ r_1(k+N_2)\ r_2(k+N_2)\ \cdots\ r_{ny}(k+N_2)]^T$$
$$\underline{\hat{y}} = [\hat{y}_1(k+1)\ \hat{y}_2(k+2)\ \cdots\ \hat{y}_{ny}(k+1)\ \cdots\ \cdots\ \hat{y}_1(k+N_2)\ \hat{y}_2(k+N_2)\ \cdots\ \hat{y}_{ny}(k+N_2)]^T$$
$$\Delta\underline{x} = [\Delta x_1(k+1)\ \cdots\ \Delta x_{nx}(k+1)\ \cdots\ \cdots\ \Delta x_1(k+N_x-1)\ \cdots\ \Delta x_{nx}(k+N_x-1)]^T \text{ i.e.,}$$
$$\Delta\underline{x} = \left[\Delta\underline{x}(k+1)^T\ \Delta\underline{x}(k+2)^T\ \cdots\ \Delta\underline{x}(k+N_x-1)^T\right]$$

The objective function (equation (4.1)) is minimized subject to the following constraints

$$\underline{x}_{min} \leq \underline{x} \leq \underline{x}_{max} \tag{4.2}$$

$$\Delta\underline{x}_{min} \leq \Delta\underline{x} \leq \Delta\underline{x}_{max} \tag{4.3}$$

$$\underline{y}_{min} \leq \underline{\hat{y}} \leq \underline{y}_{max} \tag{4.4}$$

The prediction equations in DMC are based on a linear finite step response model relating the manipulated variables to the process outputs. For a SISO system, the step response model is given as :

$$\hat{y}(k+1) = \sum_{i=1}^{N} S_i\,\Delta x(k+1-i) + S_N\,x(k-N) \tag{4.5}$$

A generalization of the above step response model may be used to construct the prediction equations for the MIMO DMC algorithm. It is assumed that the measured disturbances (or their estimates) do not change over the prediction horizon (i.e., $\Delta\underline{d}(k+l) = \underline{0}; l = 1, 2, \cdots, N_2$) and that the manipulated variables change only over a horizon N_x (i.e., $\Delta\underline{x}(k+l) = \underline{0}; l = N_x, \cdots, N_2$).

Predictions of the process outputs over the entire prediction horizon can be expressed as :

$$\underline{\hat{y}} = S\Delta\underline{x} + \underline{\hat{y}}^* + \underline{\hat{d}} \tag{4.6}$$

where S is the dynamic matrix given by

$$S = \begin{bmatrix} S_1 & 0 & \cdots & 0 \\ S_2 & S_1 & \cdots & 0 \\ \vdots & \vdots & \cdots & S_1 \\ \vdots & \vdots & \ddots & \vdots \\ S_{N_2} & S_{N_2-1} & \cdots & S_{N_2-N_x+1} \end{bmatrix} \tag{4.7}$$

Each S_i is a $ny \times nx$ matrix comprising the step response coefficients of the process model. The term $\underline{\hat{y}}^*$ is the contribution of the past input moves (up to time k-1) and the initial conditions (whose effect will die out after N sample intervals) to the future values of

the output and can be represented as

$$\hat{\underline{y}}^* = \begin{bmatrix} S_2 & S_3 & S_4 & \cdots & \cdots & S_N \\ S_3 & S_4 & \cdots & \cdots & S_N & 0 \\ \vdots & \cdots & \vdots & \cdots & \cdots & \vdots \\ S_{N_2+1} & \cdots & S_N & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \Delta\underline{x}(k-1) \\ \Delta\underline{x}(k-2) \\ \vdots \\ \Delta\underline{x}(k-N-1) \end{bmatrix} + S_{ss} \begin{bmatrix} I \\ I \\ \vdots \\ I \end{bmatrix} \begin{bmatrix} \underline{x}(k-N) \\ \underline{x}(k-N+1) \\ \vdots \\ \underline{x}(k-N+N_2-1) \end{bmatrix} \tag{4.8}$$

where S_{ss} is a matrix formed out of the final steady state values of the step response coefficients.

The disturbance signal $\hat{\underline{d}}$ can be estimated as

$$\hat{\underline{d}}(k+l) = \hat{\underline{d}}(k) = \underline{y}_m(k) - \hat{\underline{y}}(k); \quad l = 1, 2, \cdots, N_2 - 1 \tag{4.9}$$

In words, the future disturbance effects are modelled as the difference between the plant measurement and the model output predicted for the current time k.

For the unconstrained case, there exists an analytical solution $\Delta\underline{x}_{opt}$ that minimizes the quadratic objective function given by equation (4.1). When constraints do exist (as in most experimental and industrial setups), use of numerical optimization codes such as Quadratic Program SOLver (QPSOL) or convex optimization becomes mandatory. These algorithms solve the optimization problem at each sampling instant using the latest available process measurements. Under this scheme, the control law portrays a nonlinear nature since different sets of constraints may be active at any sampling instant.

4.5 Constrained Model Predictive Control in the PLS Latent Space

In the case of linear systems, equation (3.4) provides a model in terms of the original variables. Existing MPC algorithms can then be used for process control. For nonlinear systems, such a transformation may be tedious - even if it were done, controller design and calculations will be very unwieldy. Consequently, there is a strong incentive to modify the original MPC algorithms and perform control calculations in terms of the PLS latent variables with the PLS matrices (scaling and loadings matrices) serving as compensator blocks in the closed loop system.

For the unconstrained case, the dynamic inner models can be used in two ways : (1) Each inner model can be used to develop SISO DMC controllers (2) A MIMO DMC controller which utilizes all the n inner models together. If constraints are imposed on the manipulated variables, the constraints in the latent space are coupled (as described below). If strategy (1) is employed, then the controllers must act in a co-ordinated fashion. Otherwise, constraints on the original variables will be violated. With strategy (2), a one-time transformation of the constraints is adequate for the efficient implementation of the constrained DMC

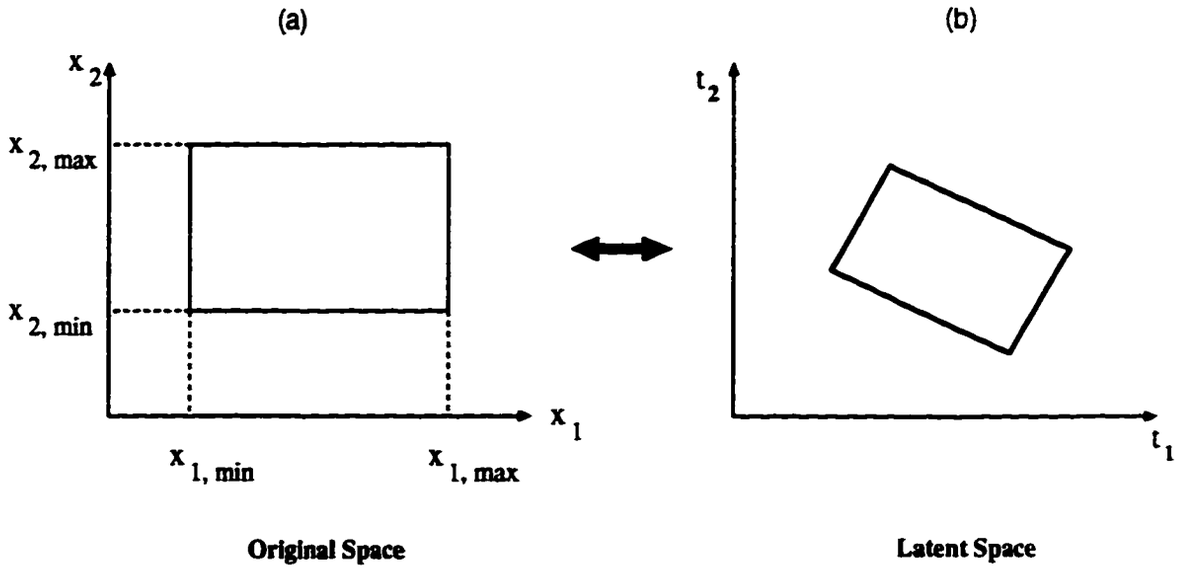

Figure 4.1: Constrained region in the original and latent spaces

algorithm.

In the original space, the constraints are represented by equations (4.2) through (4.4). For a case involving two manipulated variables, the amplitude constraints are shown in Figure 4.1(a) (mathematically expressed in equation (4.2)). In terms of the latent space variables and the PLS matrices, the above equation may be written as :

$$\underline{x}_{min} \leq tP^T S_x \leq \underline{x}_{max} \tag{4.10}$$

A graphical plot of the constraints in terms of the input space latent variables (T) is depicted in Figure 4.1(b). Use of the PLS inner models with the constraints as given in equation (4.10), will ensure the satisfaction of constraints in the original space. Such a mapping is *one to one* (when no reduction is dimensionality has been done as is the case here) - each point in the constrained original space has a unique image in the constrained latent space and vice versa. The outcome of transforming the original constraints into latent space constraints is that the constraints that were decoupled in the original space become coupled in the latent space. A similar analysis holds for the rate constraints as well. Output constraints are not considered in this work. Hence a multivariate approach (simultaneously determining the manipulated variable moves in the latent space) to control calculations is mandatory.

To examine the consequences of posing decoupled constraints in the latent space, it is necessary to determine the maximum and minimum values in the t-space, $\underline{t}_{max}$ and $\underline{t}_{min}$, such that the constraints in the original space are satisfied i.e., find $\underline{t}_{min} \leq \underline{t} \leq \underline{t}_{max}$ such that $\underline{x}_{min} \leq \underline{x} \leq \underline{x}_{max}$. When the constrained regions of the previous approach (shown by broken lines) and this approach (solid lines) are plotted together, we notice the suboptimality of

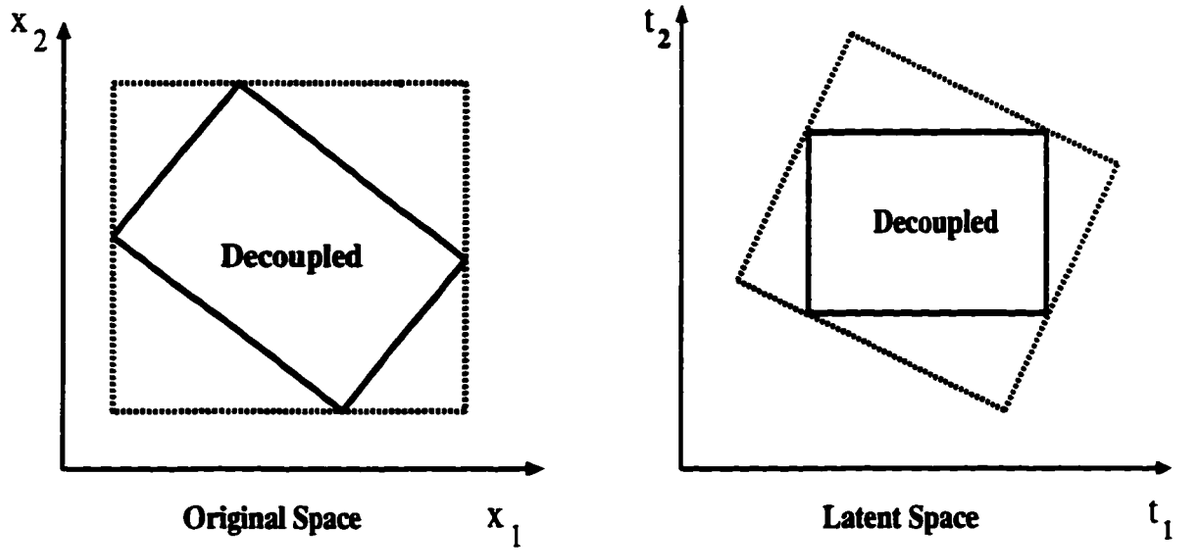

Figure 4.2: Effect of decoupling constraints in the latent Space

this approach (see Figure 4.2) - not all of the original constraint region is utilized (space bounded by broken lines) because we seek to match only the necessary conditions i.e.. the maximum and minimum values. The constraints are satisfied but the controller does not use some *permitted* regions in the input space implying that some setpoints cannot be reached and some disturbances will not be rejected completely.

The objective function in terms of the original variables is given by

$$Min\ J = \Delta x^T(S^T\Gamma S + \Lambda)\Delta x - 2(\underline{r} - \underline{\hat{y}}^* - \underline{\hat{d}})^T\Gamma S\Delta x \tag{4.11}$$

$$subject\ to\ A\Delta x + B \geq 0\ (general\ form\ of\ constraints)$$

In the latent space the above problem can be restated as

$$Min\ J' = \Delta t^T(G^T\Gamma' G + \Lambda')\Delta t - 2(r' - f' - d')^T\Gamma' G\Delta t \tag{4.12}$$

$$subject\ to\ A'\Delta t + B' \geq 0$$

where the primed quantities are the corresponding expressions in terms of the latent variables. The dynamic matrix G, the free response (future output predictions) and the disturbance estimates are all obtained from the inner PLS dynamic model. The original constraints are transformed to latent space constraints using equation (4.10) (a similar equation can be written for rate constraints i.e. Δt as well).

Note that the latent variables are scaled variables, hence the control and output weightings have to be chosen accordingly. For example, in the control of the Wood-Berry column.

it was observed that a control weighting of $\Lambda = 100I$ in the original space yields a similar performance as $\Lambda' = 0.05I$ in the latent space (ISE values were compared for several servo and regulatory runs). The time scales however, are invariant to the transformations and therefore the choice of N_1, N_2 *and* N_x can remain the same in both the original and the latent spaces.

4.6 Illustrative Examples

The constrained PLS-DMC algorithm is now tested on three process systems. First, some features such as constraint mapping, effect of tuning parameters etc. are illustrated using the simulation example of the Wood-Berry column. This example helps in establishing the fact that the proposed strategy is feasible and practical. Besides, it demonstrates that the constraint mapping suggested here is on a firm ground. The second example involves testing the PLS-DMC algorithm on a real physical system. From a set of input-output data, a dynamic PLS model is obtained for the laboratory stirred tank heater. The PLS model is then used to implement a model based predictive controller on the process. Finally, the acid-base neutralization system that was considered in the earlier chapters is modelled and controlled using a Wiener-PLS model. If the Hammerstein-PLS model identified in the previous chapter were to be used for constrained model based predictive control, it would result in the solution of an optimization problem involving a quadratic objective function and a set of nonlinear constraints. Instead, the Wiener-PLS model can be used effectively to provide constrained control of the neutralization system using the existing PLS-DMC algorithm. This is because the nonlinearity in the Wiener model is on the output side and does not involve the manipulated variables. Besides, it serves to illustrate that the PLS scores can be related in a variety of ways (algebraic/dynamic/linear/nonlinear etc.).

4.6.1 Constrained Control of the Wood-Berry Column

In Chapter 3, control of the Wood-Berry column was illustrated using digital controllers based on the dynamic PLS model. The control strategy was not able to handle constraints on the process variables (reflux and reboiler steam flow rates). Here, the DMC algorithm is utilized to perform constrained control of the column. The dynamic PLS model for this system was derived in Chapter 3.

To establish the fact that the constraints posed in the latent space do satisfy the original process constraints, considerable noise was introduced into the closed loop system resulting in the activation of both the amplitude and rate constraints. Figure 4.3 highlights the geometry of the input constraints in the original (X) and latent spaces (T). The crosses (x) indicate the constraints posed in the control design ($-0.06 \leq \Delta x \leq 0.06$, $-0.3 \leq x \leq 0.3$) and the circles (o) are the values from the simulation run. The top part of Figure 4.3 shows the constrained space in terms of the original and latent variables. It is seen that the inputs

meet the constraints in both the original and latent spaces. Furthermore, there is complete utilization of the permitted region. When decoupled constraints are posed in the latent space, the input rate and amplitude constraints are satisfied. However (as the bottom part of Figure 4.3 indicates), not all of the permitted region in the $x, \Delta x$ space is used resulting in suboptimal control .

As a next step, the effect of two DMC tuning parameters are examined to see if the usual trends are followed. In Figure 4.4, the regulatory response to step disturbances in feed flow rate (+1 unit at time = 0 seconds) and feed composition (+10 units at time = 125 seconds) is presented for two values for the input weighting matrix. The solid line is the response for $\Lambda = 0.05I$ and the dashed line is the response for $\Lambda = I$. As expected, the effect of the increased Λ is to make the closed loop response sluggish. For the same set of disturbances, an increase in the value of N_2 (the prediction horizon) from 5 (solid line in Figure 4.5) to 25 (dashed line) results in a relatively slower rejection of the disturbances. These simulations confirm that the DMC tuning parameters have the same effect irrespective of whether the control is performed on the original space or the latent variable space.

4.6.2 Real-Time Control of the Laboratory Stirred Tank Heater

The laboratory stirred tank heater, considered in Chapter 2, is used here to demonstrate the PLS-DMC control strategy. Using the same input-output data analyzed in chapter 2 (in connection with the CVA and N4SID methods), the following dynamic PLS model is obtained using the technique outlined in the previous chapter :

$$S_x = \begin{bmatrix} 6.3633 & 0 \\ 0 & 5.8337 \end{bmatrix} : S_y = \begin{bmatrix} 2.5156 & 0 \\ 0 & 3.9477 \end{bmatrix}$$

$$P = \begin{bmatrix} 0.6974 & -0.6979 \\ -0.7166 & -0.7162 \end{bmatrix} : R = \begin{bmatrix} 0.7164 & -0.7169 \\ -0.6982 & -0.6997 \end{bmatrix} ; Q = \begin{bmatrix} 0.8595 & 0.6620 \\ 0.2651 & -3.8435 \end{bmatrix}$$

$$G_1 = \frac{0.0131z^{-1}}{1 - 1.7830z^{-1} + 0.7962z^{-2}}$$

$$G_2 = \frac{0.0077z^{-1}}{1 - 1.7106z^{-1} + 0.7183z^{-2}}$$

The model fit arrived at using the above model is depicted via the scatter plot in Figure 4.6. As with the CVA and N4SID methods, the tank level is not modelled as well compared to the exit temperature.

A personal computer running real-time MATLAB/SIMULINK was used to control the stirred tank heater. The model obtained above is used in the DMC calculations while employing $N_1 = 1$, $N_x = 2$, $N_2 = 20$ and $\Lambda' = 10\,\mathrm{I}$ as the controller parameters. The results

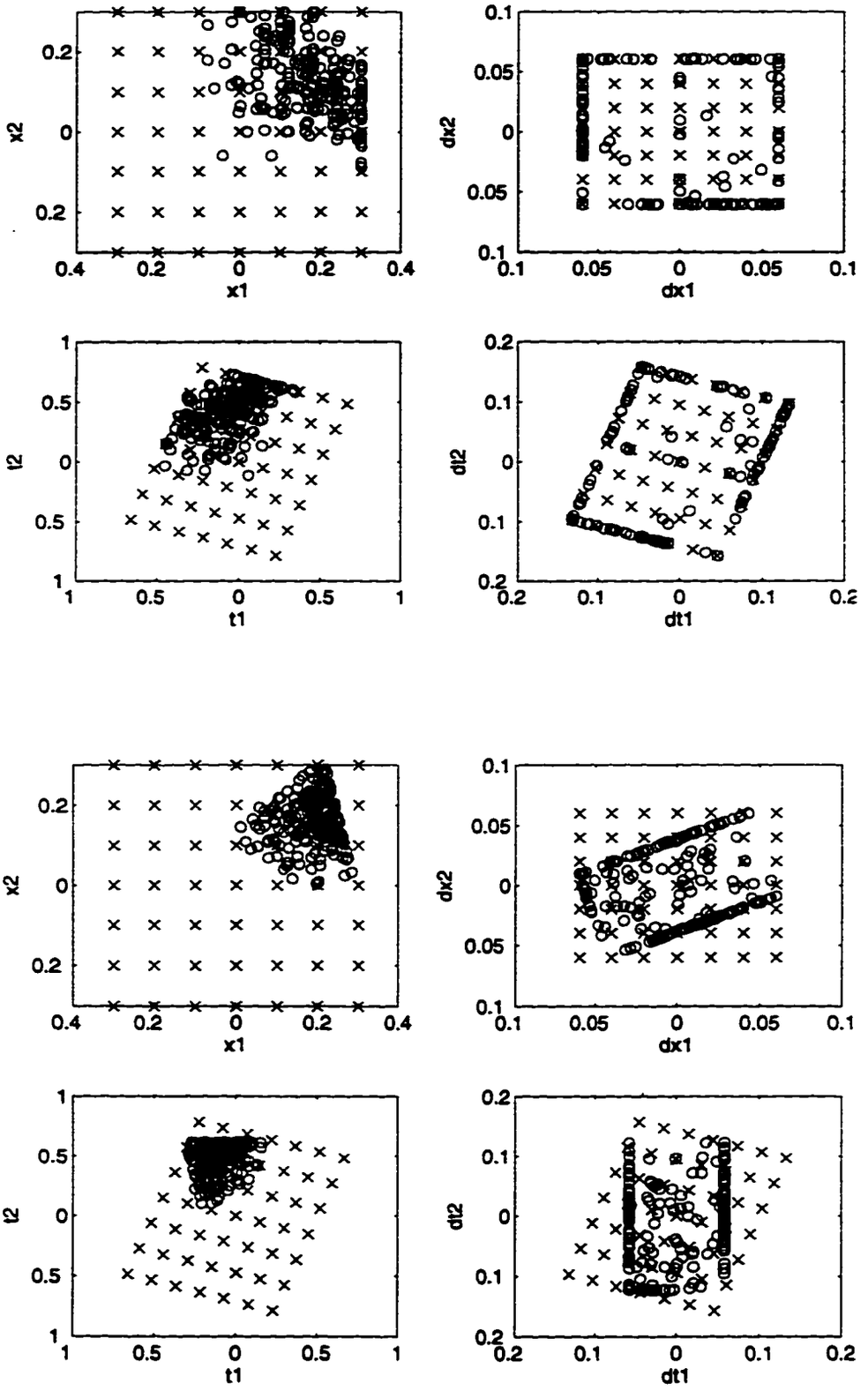

Figure 4.3: Projection of process variables on to the constrained space - (Top) Coupled Constraints in the latent space : (Bottom) Decoupled constraints in the latent space

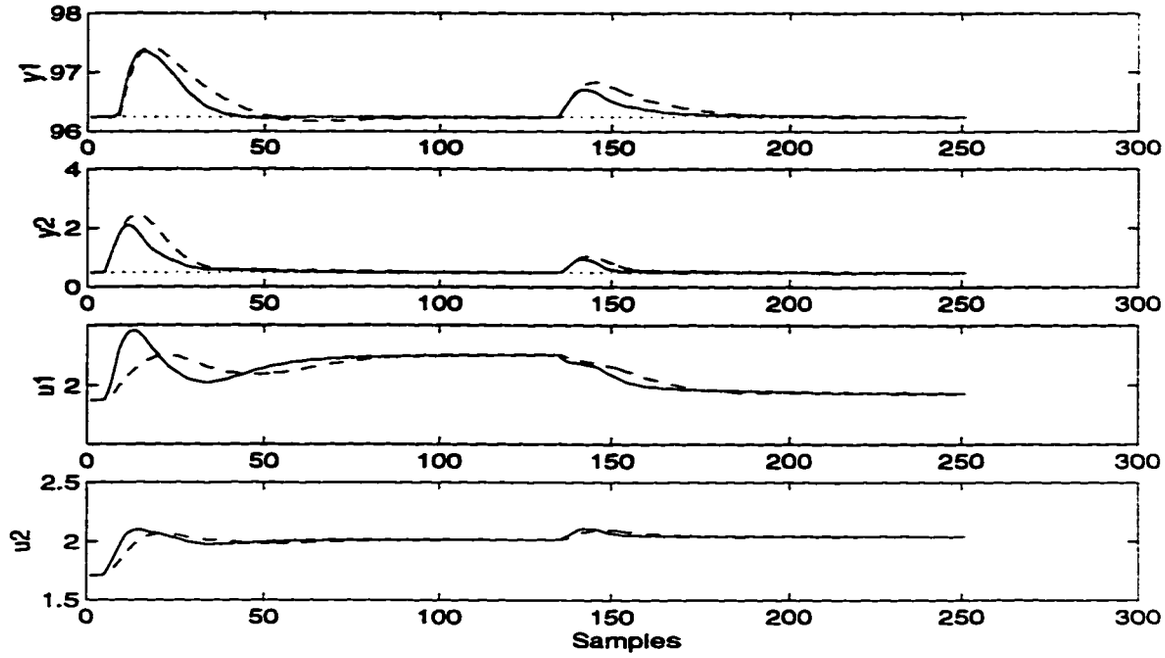

Figure 4.4: Effect of input weighting (Λ') on the closed loop performance of the PLS-DMC strategy

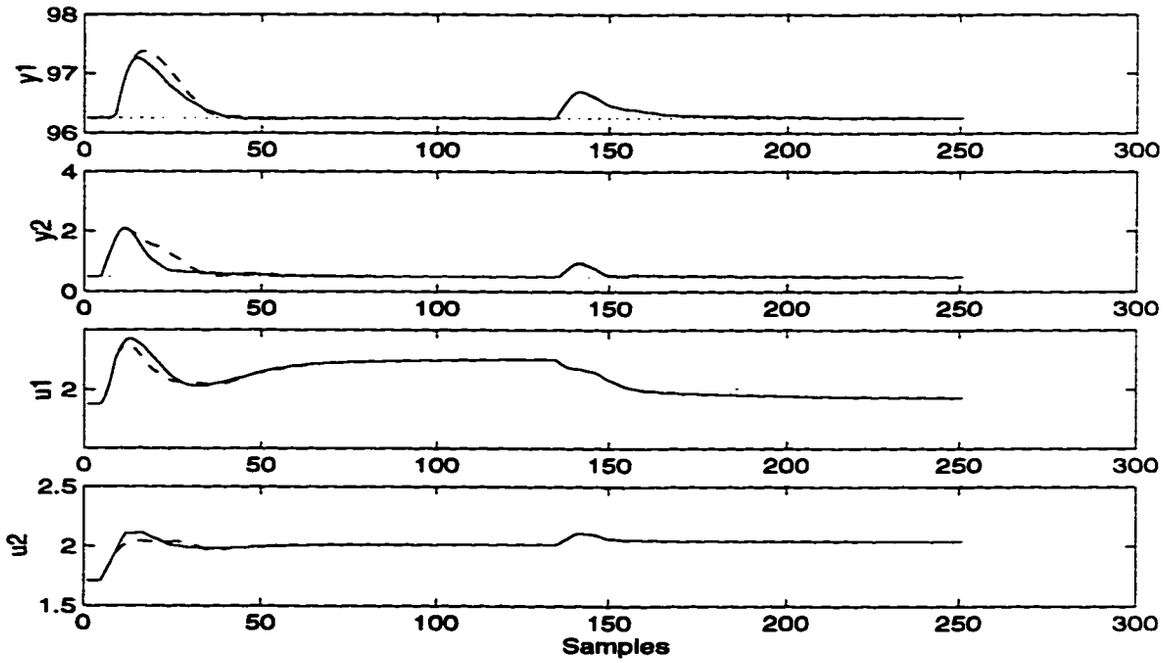

Figure 4.5: Effect of prediction horizon (N_2) on the closed loop performance of the PLS-DMC strategy

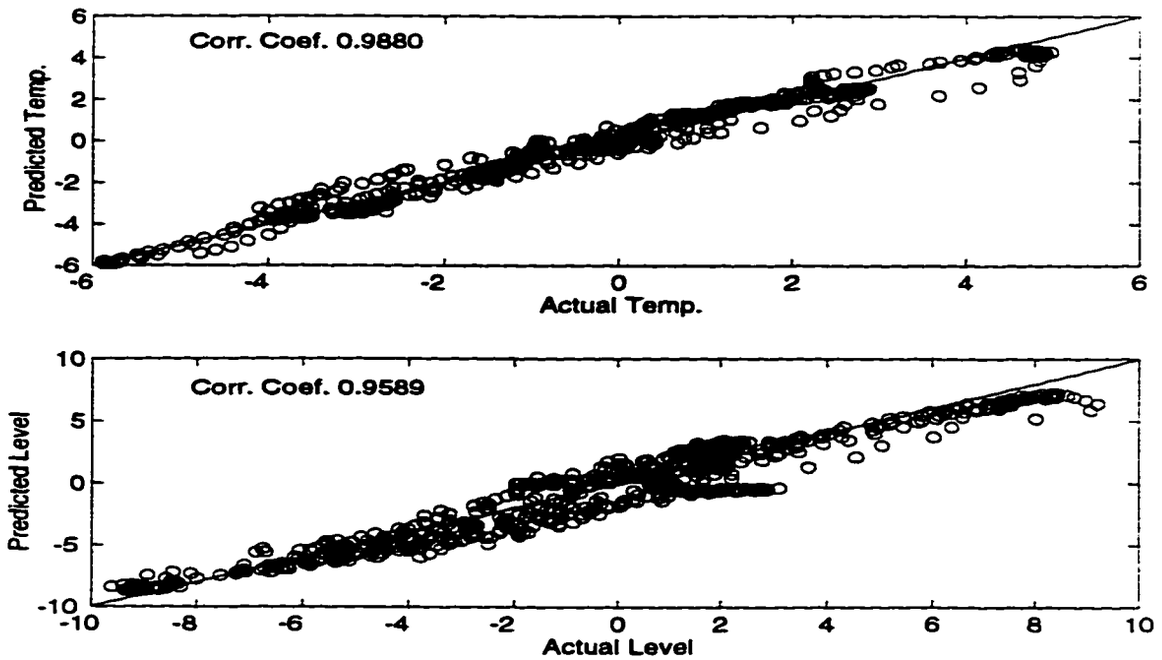

Figure 4.6: Model fit for the Laboratory Stirred Tank Heater using the dynamic PLS model

of the laboratory run showing servo and regulatory responses is depicted in Figure 4.7. Separate and simultaneous setpoint changes were made in the output variables. In order to demonstrate the performance of the controller in eliminating unmeasured disturbances. an unknown amount of cold water was dumped into the vessel around the 1575^{th} sample point. It is seen that the servo and regulatory responses were satisfactory. The constraints placed on the amplitude (0-100%) and rate (±10%) of the manipulated variables remained inviolate over the entire duration of the experiment.

4.6.3 Constrained Control of the Acid-Base Neutralization System

The acid-based neutralization system has been considered in the previous two chapters. Conventional digital control of the above system, employing the Hammerstein-PLS model. was illustrated in the previous chapter. The lack of ability to handle constraints on the manipulated variable moves is the main drawback of that approach. To be able to deal with constraints on the rate and amplitude of the acid and base flow rates, a model predictive controller that is based on an optimization approach is necessary. With the growing availability of powerful computers and nonlinear programming methods (for optimization). nonlinear process control problems such as this can be solved via on-line optimization techniques. The topic of nonlinear model predictive control (using fundamental process models) is reviewed by Biegler and Rawlings (1991). A review of differential geometry concepts for nonlinear process control is provided by Kravaris and Arkun (1991). Using nonlinear transformation of some process variables, the nonlinear problem is converted to a linear problem

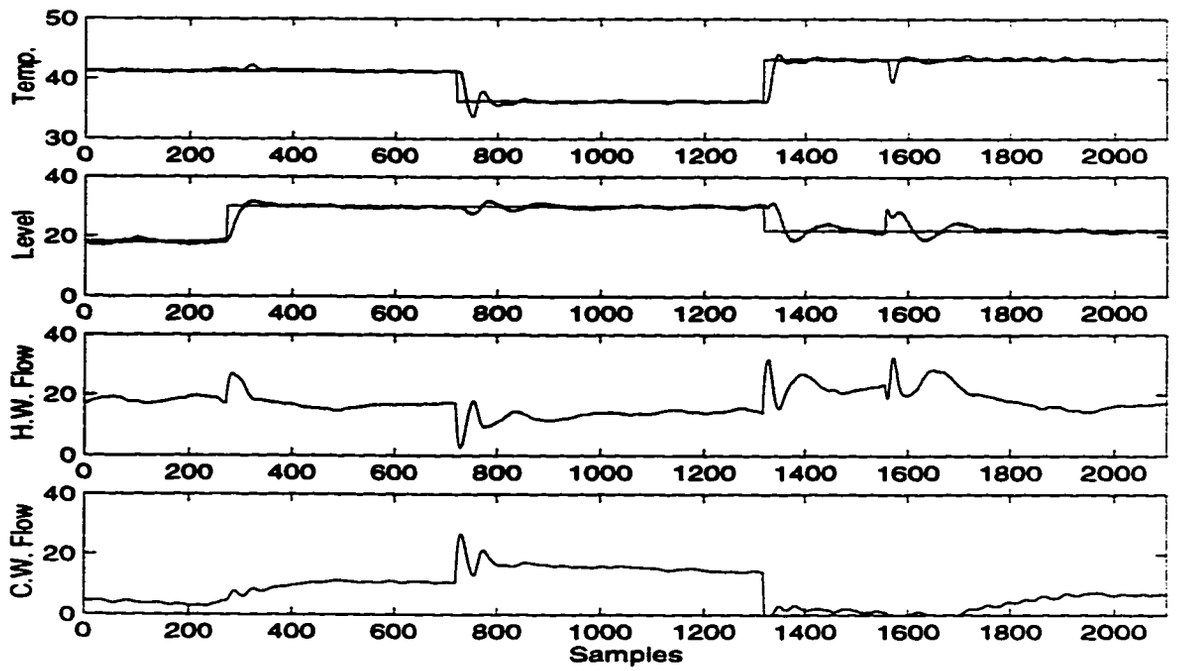

Figure 4.7: Experimental evaluation of the constrained PLS-MPC scheme

involving the transformed variable. Direct use of fundamental nonlinear process models for process control has been considered in the Generic Model Control (GMC) approach of Lee and Sullivan (1988). Nonlinear MPC approaches based on black-box models derived from plant input-output data have also been reported. Zhu and Seborg (1994) employ a Hammerstein model to control pH in an acid-base neutralization tank using an unconstrained MPC law. Recently Norquay *et al.* (1996) illustrated a constrained MPC algorithm (Only output constraints were considered in their work) using a Weiner model to control the pH in an experimental neutralization tank. It is these approaches (that are based on purely empirical models) that will be the focus of this thesis.

One fact that comes to light from a literature review is that *both the Hammerstein and Wiener structures* are suitable parameterizations for the modelling of the acid-base neutralization system. From a control point of view (particularly when constraints are placed on the rate and amplitude of the manipulated inputs), the Wiener model is more amenable for use within a linear MPC framework. In contrast to the Hammerstein model, the Wiener model consists of a linear dynamic element followed by a nonlinear static element (see Figure 4.8). The output of the linear dynamic element can therefore be considered equal to the inverse nonlinear transform of the process output (for example, in terms of the PLS score variables we can write $u_1 = \mathcal{N}(t_1^*)$ or $t_1^* = \mathcal{N}^{-1}(u_1)$. Here, t_1^* is the filtered input scores and u_1 represents the output scores of the first PLS dimension).

Since the nonlinearity involves the process outputs, with a suitable transformation of the process output, a linear MPC law can be implemented to perform constrained (input

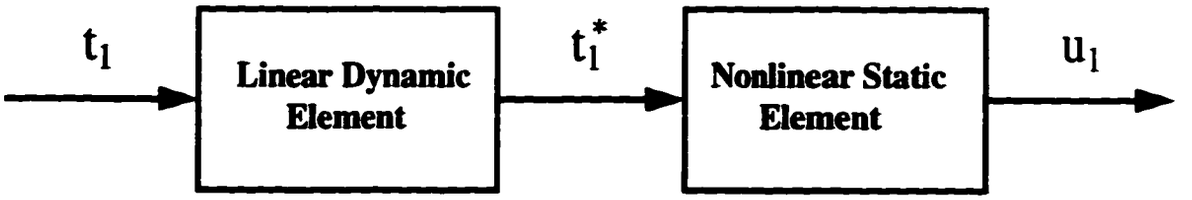

Figure 4.8: Structure of the inner relationship in the WIENER-PLS model

rate and amplitude) control of the system represented by the Wiener model. With the Hammerstein model, the nonlinear transformation is on the input side and therefore the linear input constraints are converted to nonlinear constraints. The presence of nonlinear input constraints complicates the control calculations; a nonlinear optimization technique must be used (for a detailed study, the interested reader is referred to Patwardhan (1996)). Here, the Wiener-PLS model will be used to simplify the control strategy as well as to prove that several model structures can substitute for the inner PLS relationship.

First, some plant data was collected by perturbing the neutralization system with probing signals of magnitude $\pm 10\%$ of the steady state values of the acid and base flow rates (refer Example 3 of Chapter 3 for some details). The input-output data were autoscaled and the dynamic PLS algorithm was used to obtain a model. The model is given by equations (4.13) through (4.20).

$$S_x = \begin{bmatrix} 1.0447 & 0 \\ 0 & 1.0656 \end{bmatrix} \tag{4.13}$$

$$S_y = \begin{bmatrix} 0.9558 & 0 \\ 0 & 0.9505 \end{bmatrix} \tag{4.14}$$

$$P = \begin{bmatrix} -0.6920 & 0.7402 \\ 0.7224 & 0.6723 \end{bmatrix} \tag{4.15}$$

$$R = \begin{bmatrix} -0.6723 & 0.7224 \\ 0.7402 & 0.6920 \end{bmatrix} \tag{4.16}$$

$$Q = \begin{bmatrix} 0.0653 & 0.9999 \\ 0.9979 & 0.0115 \end{bmatrix} \tag{4.17}$$

As in the case of the Hammerstein-PLS model described in the previous chapter, a close look at the elements of the Q matrix reveals that the first PLS dimension models the pH and the second dimension models the level. This means that the nonlinearity is confined to the first PLS dimension. The inner relationship for the first PLS dimension is captured by a Wiener structure and an ARX model is adequate for the second PLS dimension.

The linear part of the Wiener model is

$$G_1 = \frac{0.1325 z^{-1}}{1 - 0.8949 z^{-1}} \tag{4.18}$$

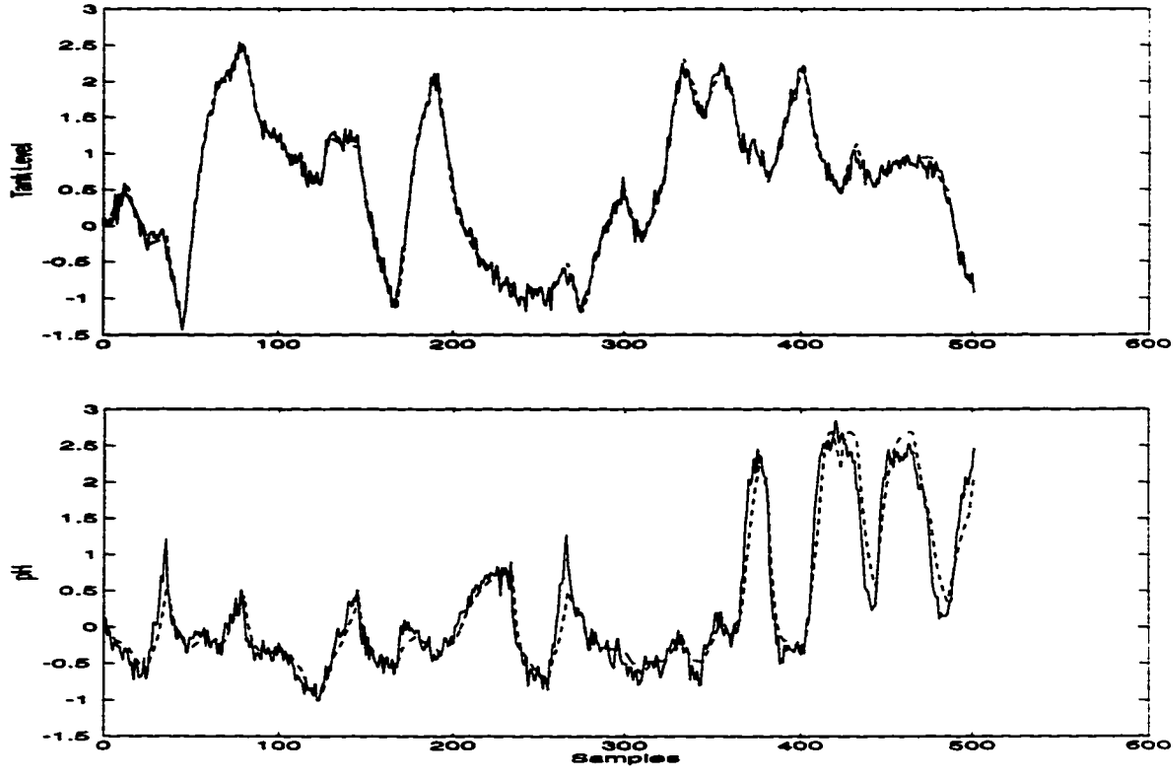

Figure 4.9: Model fit using a Wiener model in the PLS inner relationship : Model (dashed line) and Actual Plant (solid line)

The static nonlinearity (i.e., $u_1 = \mathcal{N}(t_1^*)$) is determined as

$$u_1 = -0.0351\, t_1^{*5} - 0.0908\, t_1^{*4} + 0.1622\, t_1^{*3} + 0.6056\, t_1^{*2} + 0.8333\, t_1^* \tag{4.19}$$

The inner model for the second dimension is :

$$G_2 = \frac{0.1384z^{-1} - 0.0308z^{-2}}{1 - 0.9908z^{-1} + 0.0455z^{-2}} \tag{4.20}$$

The model fit is shown in Figure 4.9 and indicates a reasonably good fit of the data. This fact is also borne by the cross validation performed using a different input sequence and presented in Figure 4.10.

Equation (4.19) needs to be solved in order that the process nonlinearity be removed from the information processed by the controller. This implies that the roots of the 5^{th} order polynomial must be determined at each sampling interval (similar to the control using the Hammerstein model in chapter 3). To simplify the computations during control, the inverse nonlinear transformation (i.e, $t_1^* = \mathcal{N}^{-1}(u_1)$) is identified. The polynomial transformation

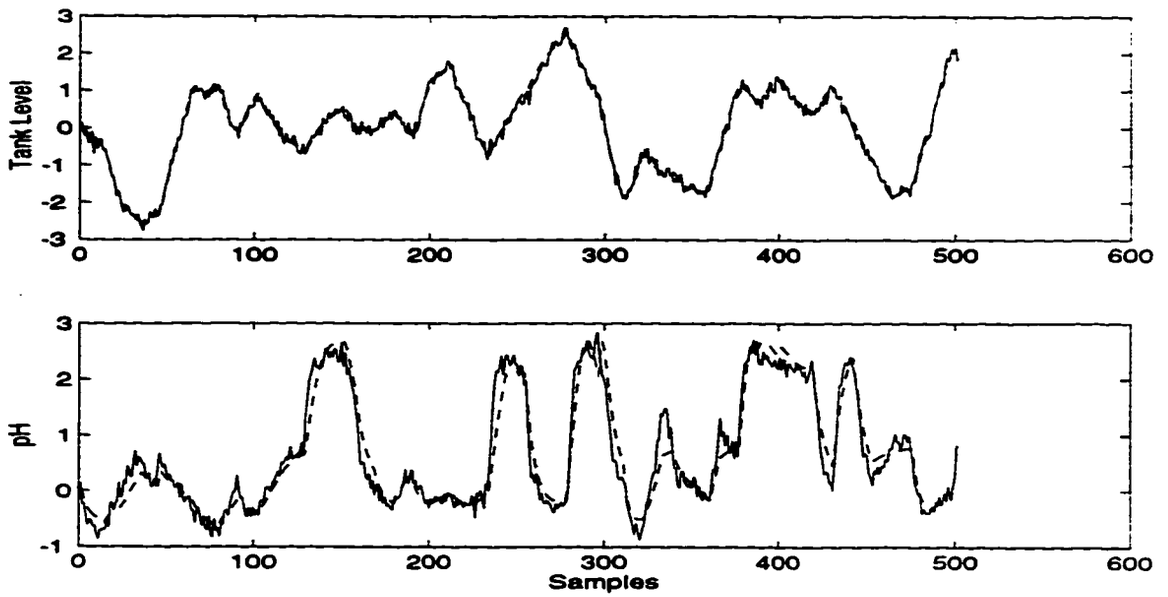

Figure 4.10: Cross Validation of the Wiener-PLS model : Model (dashed line) and Actual Plant (solid line)

is given by

$$t_1^* = -0.0554\,u_1^5 + 0.2603\,u_1^4 - 0.0422\,u_1^3 - 0.9743\,u_1^2 + 1.4508\,u_1 \qquad (4.21)$$

The polynomial given by equation (4.21) can be evaluated at each sampling instant to determine t_1^* needed by the linear constrained control algorithm. The resulting Wiener-MPC constrained control strategy is shown in Figure 4.11. The pre- and post- compensator blocks are similar to that discussed in the previous chapter. The controller block is now a linear PLS-MPC algorithm such as the PLS-DMC algorithm discussed earlier. By including the inverse nonlinear transform in the control loop, it is made sure that the controller *sees* and *controls* a linear multivariable process.

The Wiener-PLS model identified earlier was used to implement a constrained Wiener-DMC algorithm on the acid-base neutralization process with $\Lambda = 0.3I$, $N_2 = 10$, $N_1 = 1$ and $N_x = 2$ as the controller parameters. The acid and base flow rates are constrained to remain within ± 5 ml/s of their steady state values. Furthermore, the manipulated variables were subject to maximum move size limitations of magnitude 0.5 ml/s.

The response of the neutralization system to several setpoint changes in level and pH are shown in Figure 4.12. The control appears to be satisfactory. Figure 4.13 indicates that all the constraints are satisfied in both the latent and original variables. Again, the crosses (x) indicate the constraints posed in the control and the circles (o) are the values from the simulation run. The horizontal streak of circles ($t_2 \approx 0$) in the t_1 versus t_2 plot result from setpoint changes made in the pH while keeping the level constant (since the first dimension

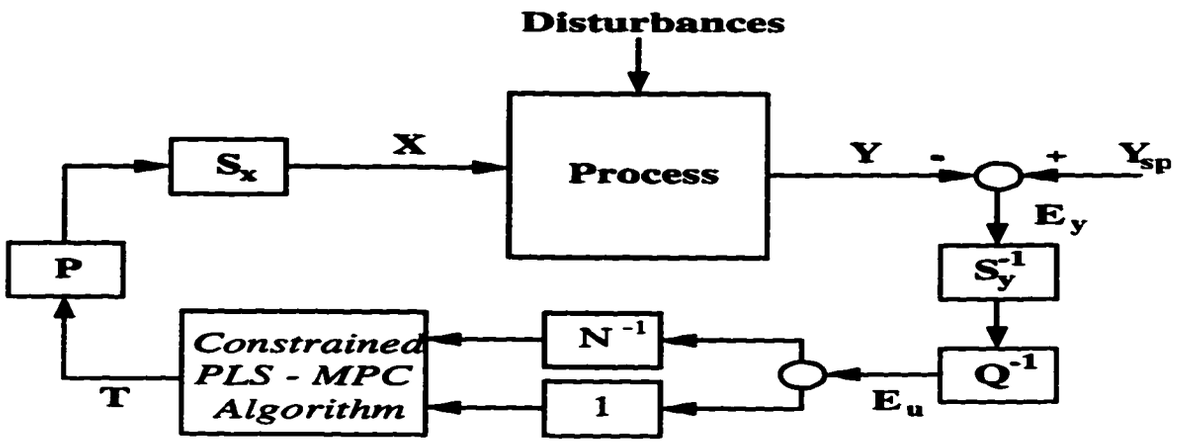

Figure 4.11: Schematic of the constrained Wiener-DMC control strategy for the acid-base neutralization system

models the pH. movement is noticed only along t_1).

It must be conceded that since a high order polynomial has been used in the Wiener model, setpoint changes of large magnitude may result in confusing the controller - the actual process gain may be quite different from the gain of the model and in certain cases may be of different sign ! To avoid such problems, it may be necessary to replace the high order polynomial representation of the nonlinearity by several linear elements or have multiple Wiener models covering the entire range of process operation.

4.7 Conclusions

This chapter described a framework for implementing advanced model predictive control systems that seamlessly integrates with the dynamic PLS modelling strategy described in the previous chapter. The control calculations are done based on the input and output scores after effecting a suitable modification of the original constraints. Issues related to the mapping of the constraints were discussed. It was shown that the independent constraints on the rate and amplitude of each manipulated input gets transformed into dependent constraints on the input scores. Nonlinear systems that are modelled with the dynamic PLS algorithm, can also be controlled using this framework - however, the computational load will be considerably high compared to that of linear systems. Feasibility of the PLS-MPC approach and its usefulness in the control of linear and nonlinear systems has been demonstrated. Whether the proposed strategy will simplify the existing tuning procedures remains an open question. So is the issue of robustness.

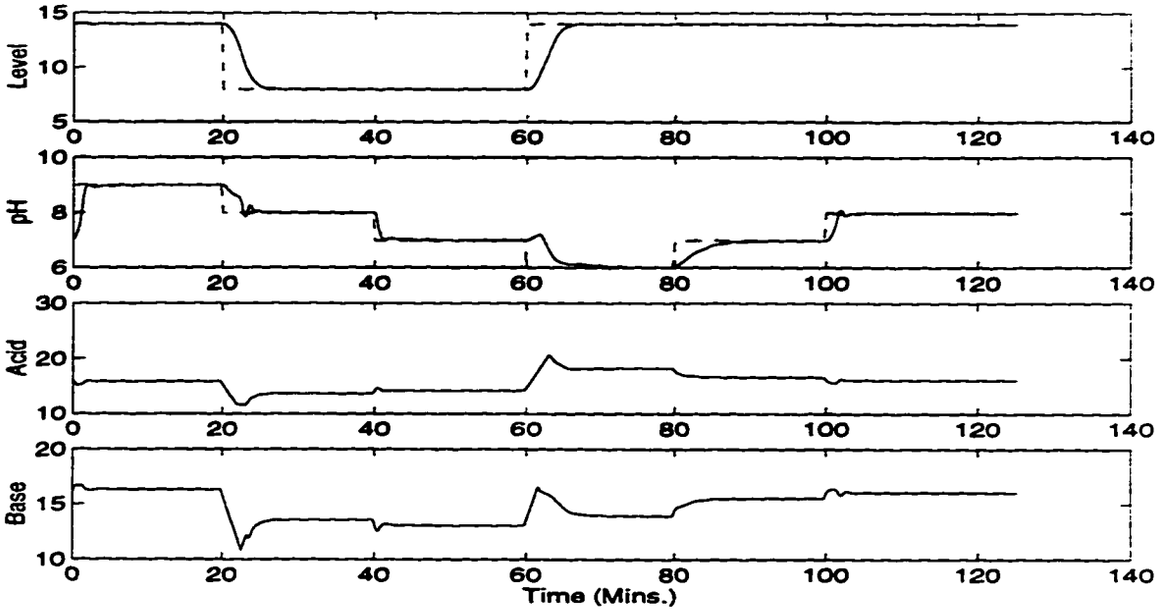

Figure 4.12: Constrained PLS-DMC control of the Acid-Base Neutralization System using the WIENER-PLS model

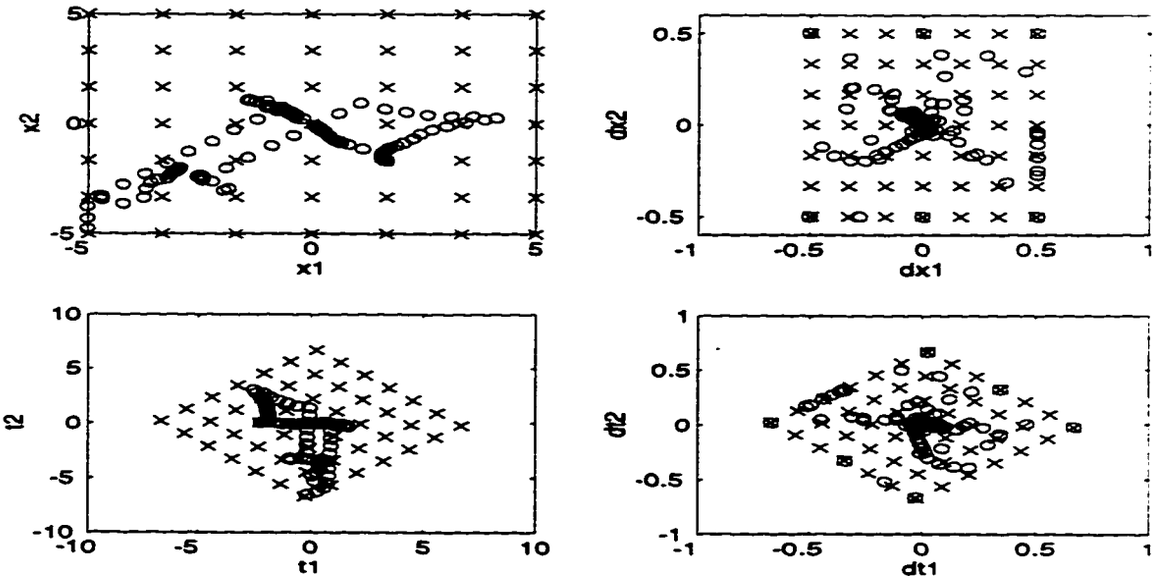

Figure 4.13: Projection of rate and amplitude of the manipulated variables on to the constrained space - (Top) Original space; (Bottom) Latent space

Chapter 5

Monitoring and Fault Detection of Batch Processes

5.1 Overview

A data-based approach to the monitoring and fault detection of batch and semibatch processes is considered. Developed by Nomikos and MacGregor (1994a. 1994b. 1995) using PCA and PLS in a SPC type framework. this method obviates the need for any fundamental process models or a rule-based troubleshooting system. Using data obtained from several batches of good runs. a *template of normal plant operation* is obtained. This usually takes the form of a PCA or PLS model that captures the relationships between the variables as they evolve in time. Monitoring charts (with confidence limits) may be obtained from these models and can be used to compare the performance of batches currently under production.

In this chapter, we describe the basics of the PLS based monitoring approach for batch processes. In order to utilize the concepts for our applications. we make provisions for handling the multiple rates of sampling (that are particularly common in batch units) as well as suggest strategies to build the *normal template* from a database that contains good runs of varying length. The monitoring algorithm is evaluated on simulations involving a fed-batch antibiotic producing fermentation unit and a semibatch polymerization reactor.

[1]Sections of this chapter have been presented as

1. S. Lakshminarayanan, R.D. Gudi, Sirish L. Shah and K. Nandakumar,"Online Monitoring of a Fed-batch Fermentor using Multirate-Multiblock-Multiway Projection to Latent Structures", AIChE Annual Meeting, Miami Beach. 1995.
2. S. Lakshminarayanan, R.D. Gudi, Sirish L. Shah and K. Nandakumar,"Monitoring Batch Processes using Multivariate Statistical Tools : Extensions and Practical Issues", Presented at the 13^{th} IFAC World Congress, San Francisco, 1996.

5.2 Contributions of this chapter

- The PLS based batch process monitoring technique proposed by Nomikos and MacGregor is extended to include : (1) multirate sampling and (2) normal batches with varying run lengths. This makes the original algorithm more suited for practical applications.

5.3 Introduction

A great majority of high quality specialty chemicals are manufactured in the batch mode. The enormous flexibilities offered by batch processes such as the production of multiple products, less rigorous equipment design and sizing procedures have aided in their growing popularity and success. The uncertain and shifting market forces will ensure that more and more chemicals will be produced this way.

Batch processes operate for a finite duration of time. Their time evolution is characterized by nonlinearity and lack of a steady state. For the routine production of a product (or a group of products) it is necessary to track a prescribed recipe during every batch. Reproducibility in batch operations is seldom possible owing to the many sources of variability (e.g. process disturbances, *start-stop-change nature* of the process) one has to cope with. Furthermore, there is always a risk of the high value product becoming contaminated during the production run. The high market value associated with the product and the operation of the batch unit calls for a good monitoring and fault detection strategy. An early detection of faults can help in taking corrective action, when possible, to alleviate the fault or to shut down the batch to prevent wastage of expensive feed material and process utilities.

Monitoring and fault detection of batch processes can be accomplished using traditional Kalman filter based methodologies (e.g. King (1986), MacGregor *et al.* (1986)) *if* an adequate mathematical model incorporating conservation principles and empirical relationships is available. Often times, such mathematical descriptions are not available and it is difficult to characterize the interrelationships between variables over the entire duration of the batch. Use of Artificial Intelligence (AI) tools such as expert systems that are constructed using the experience of plant personnel frequently involves extensive consultation with plant operators to be able to build a good knowledge base on the process. These are also restrictive because they can take into account only as many fault occurrences and process variable interactions, that the plant personnel can envision. Pattern classification techniques have also been proposed for process monitoring (Venkatsubramanian and Chan (1989)). Here, a database of regular and faulty modes of plant operation is constructed using historical plant data. Future plant operation can then be classified as *good* or *bad* using a pattern classifier. Obtaining data sets that capture the many possible faults has proved to be a bottleneck with this approach.

Through a series of papers, Nomikos and MacGregor (1994a. 1994b. 1995) introduced the use of statistical models based on multivariate statistical techniques such as Principal Components Analysis (PCA) and Partial Least Squares (PLS) for process monitoring and fault detection. Dong and McAvoy (1994) used a nonlinear PCA model for batch process tracking. In these approaches, *normal* plant data, that are abundant and readily available. are used to construct a "template" of the normal operation of the process. This template is the *data-based statistical model* that holds good as long as the process is in a state of *statistical control*. Using a lower dimensional window on the process, these methods can detect any faults that cause deviation from the prescribed operation recipe. Faults can be isolated by interrogating the underlying statistical model or by using contribution plots (Miller *et al.*, 1993). This is similar to the PLS based monitoring and fault detection strategy described in Chapter 1 with some modifications made to reconcile with the temporal evolution of process variables in batch and semibatch processes.

This chapter is organized as follows. A description of the PLS based methodology proposed by Nomikos and MacGregor for the monitoring of batch processes is first provided. Extensions of the technique in order to handle the multiple rates of measurement and data records of unequal lengths are presented next. In the final section. we consider simulation examples involving a bioreactor[2] and a polymerization reactor[3].

5.4 Statistical Analysis of Batch Data

5.4.1 Database Structure

The database structure from which the monitoring scheme is to be developed is explained using the example of a fermentation bioreactor. This presents no loss of generality as most batch systems will conform to the measurement scenario depicted here. Figure 5.1 shows the measurement system commonly associated with the bioreactor system. The primary process variables or culture states (measured infrequently) are the biomass. substrate and antibiotic concentrations forming the primary variables block (PVB). The final antibiotic concentration needs to be predicted on-line and makes up the quality block (QB). Secondary measurements are available more rapidly from the fermentation through the use of non-invasive sensors. Typical among these are the CO_2 evolution rate (CER) and the oxygen uptake rate whose values can be made available through the analysis of exit gas concentrations using an on-line mass spectrometer. The dissolved oxygen concentration is a critical process variable for an aerobic fermentation and its levels are measured using a dissolved oxygen probe. Measurements of dissolved CO_2, broth levels as well as estimates (from a Kalman filter, for example) of quantities such as the gas-liquid mass transfer coeffi-

[2]Some portions of this work were done in collaboration with Dr. Ravindra Gudi as part of an ongoing research project

[3]I like to thank Prof. John MacGregor and Dr. Paul Nomikos for providing this data set

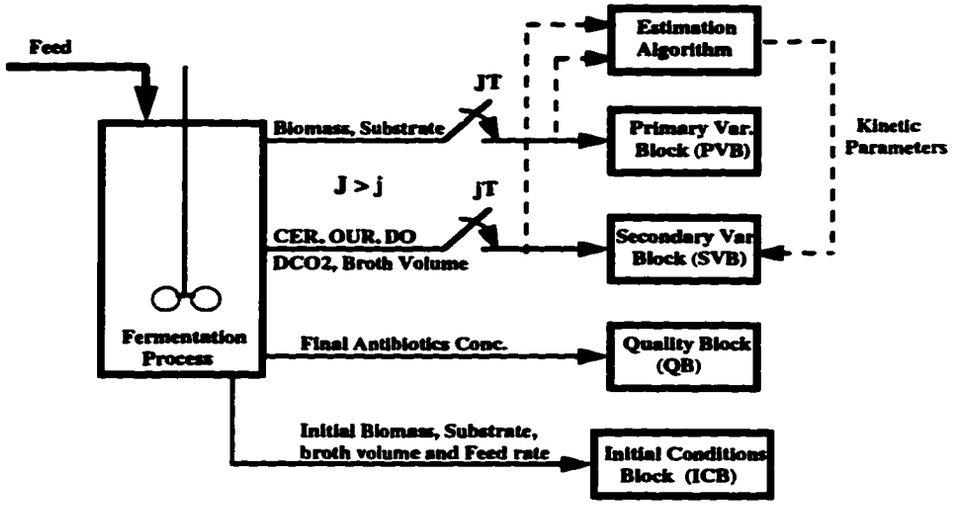

Figure 5.1: Measurement System for the Fed-batch Fermentation Process

cients, specific growth rate are also assumed to be available frequently. These measurements form the secondary variables block (SVB). The initial conditions block (ICB) is formed by incorporating the feed rate (assumed constant) along with the initial conditions on the biomass, substrate concentrations and the broth volume that are used in the fermentation.

Data from such batch/semibatch systems are assumed to be logged on to a database. In the model building step, data on batch runs that produced acceptable final product are first extracted in order to characterize the *healthy* operation of the plant. As shown in Figure 5.2, these normal runs are grouped into blocks that naturally emerge from the characteristics of the multirate measurement system. It is assumed that data is available on NB batches. The number of variables in the initial conditions block, the primary variables block, the secondary variables block and the quality block are denoted by NIC, NP, NS and NQ respectively. NPS and NSS indicate the number of samples available for the primary and secondary variables respectively. The ICB and the QB are two dimensional as expected. PVB and the SVB are three dimensional entities as they carry information on time evolution of several variables for many "normal" batches. Note that the SVB has a greater *depth* compared to PVB because relatively fewer measurements of the primary states are measured during the course of the batch. It is also very common to have a smaller number of primary variables compared to secondary variables. In addition, data from normal batches used in building up these blocks can have differing time lengths. Thus, the PVB and the SVB may not have a constant depth for all the normal batches. Consequently, the real database may appear as shown in Figure 5.3. Note that in Figures 5.2 and 5.3, the arrows from the initial conditions block, primary variables block and the secondary variables block point towards

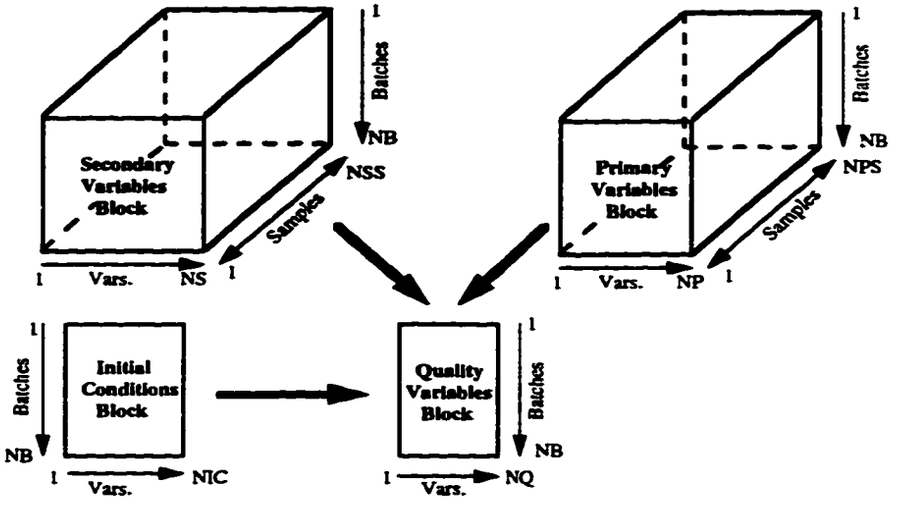

Figure 5.2: The database structure for multirate batch process monitoring. Note that all batches in the database have the same run lengths.

the quality block. This is because besides the task of process monitoring, the PLS model is also intended to provide online predictions of the final product quality using the available process measurements.

In the approach proposed by Nomikos and MacGregor, the process variables are lumped into a single three dimensional data block. Here, two *three-dimensional* data blocks have been explicitly defined taking the liberty to assume that all the primary measurements become available at the same time. This demarcation of variables into primary and secondary blocks is preferred from the point of view of process monitoring - abnormal behavior in any of the blocks can be picked up quickly and easily. Such an idea has been used by MacGregor *et al.* (1994b) where a continuous tubular reactor is sectioned into two parts in order to isolate process faults and upsets effectively. In their study, the process variables were segregated into multiple blocks to handle the *physical* aspects of the problem. In this study, the multiblock feature is employed to deal with the multirate sampling scenario.

5.4.2 The Wangen-Kowalski Algorithm

In Chapter 1, the PLS model was developed using only two blocks of variables - the X and Y blocks. It is obvious that the original PLS algorithm needs to be modified to cope with several blocks of variables (in this case, we have 3 blocks of variables that predict the quality block - i.e., three X blocks and one Y block). The three dimensional blocks do not pose a major problem. It has been shown that similar results can be obtained either using tensorial computations (details can be found in Sanchez and Kowlaski, 1990): here the three

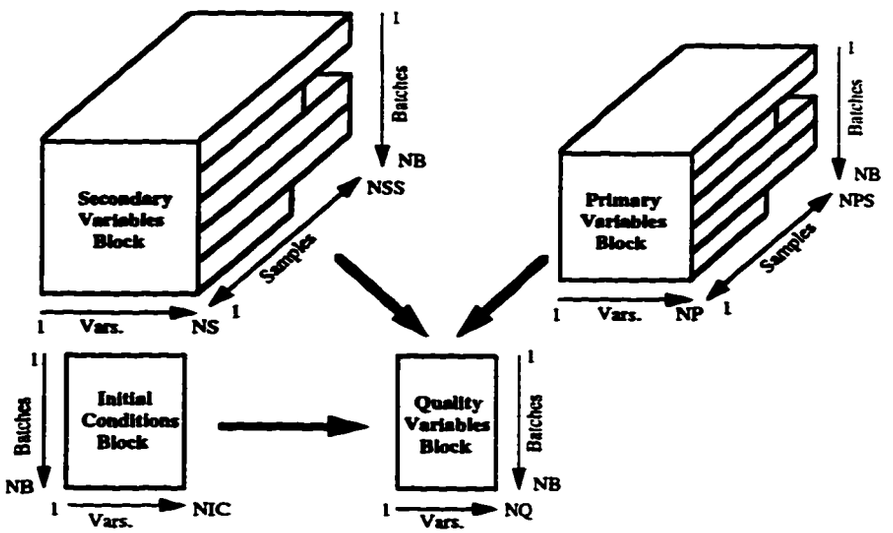

Figure 5.3: The database structure containing batches with different run lengths.

dimensional nature of the blocks are preserved) or by unfolding the three-dimensional arrays into two-dimensional matrices followed by a simple multiblock PLS analysis as proposed by Wangen and Kowalski (1988). The unfolding operation is done by slicing up the three-dimensional array at each sampling instant and placing them side by side. For example, the primary variable block of dimension $NB \times NP \times NPS$ can be unfolded into a two-dimensional matrix of size $NB \times (NP * NPS)$ (the * sign indicates the usual multiplication operation).

A detailed description of the algorithm is outside the scope of this thesis. It suffices to say that the Wangen-Kowalski algorithm is conceptually similar to the two block PLS algorithm described in chapter 1 and that it can handle even complicated relationships between blocks of variables (this algorithm can be used in more complicated situations; e.g. event driven operations such as batch distillation). This approach deals with two-dimensional matrices, therefore interpretation of the results is straightforward.

5.4.3 Data Pretreatment

Before unfolding the three-dimensional arrays, it is necessary to *fill them up* - i.e., replace the unknown entries for the shorter runs lengths with reasonable values. If this were not done, two choices remain. They can be summarized as follows :

- The PLS model can be built by only considering samples until the shortest run length in the database. This is too restrictive since the plant operator will be devoid of a monitoring tool if the run length of a new batch exceeds the run length of the shortest run length found in the database.

- Several PLS models can be obtained by considering all run lengths between the shortest and the longest. This may become quite unwieldy as it may be very difficult to maintain several models and switch between them as time progresses.

A more reasonable approach is to fill up the shorter runs with appropriate values and make all batches of the same run length (corresponding to the longest run in the database). This means we still have only one PLS model and the capability of monitoring new batches until a time equal to the longest run length in the reference database. For each time instant, obtain the mean and the standard deviation of the variables using only the batches that were in operation. Replace each "known" entry with its scaled deviation. Now, fill the unknown data with :

1. Zeros. This is equivalent to considering that if the particular batch continued any longer, it would not deviate from the mean trajectory obtained from longer batches.

2. The scaled deviations noted at the end of each batch run. Here, it is assumed that if the particular batch continued any longer, it would deviate by exactly as much from the scaled mean trajectory as it did at the end of the run. This appears to be a more realistic situation and will be employed in this study.

The above operation serves two purposes : (1) it autoscales the data using the available information and (2) the *jagged* database structure depicted in Figure 5.3 is transformed into a *complete* database (as in Figure 5.2). With the unfolding of these *complete* three-dimensional arrays, the Wangen-Kowalski algorithm can be readily applied. Also, the subtraction of the average trajectories from the process variables serves to remove the dominant nonlinear and nonstationary components from the data - so linear model building tools can be used with a fair measure of confidence.

5.4.4 Development of the PLS Model and Monitoring Charts

The construction and use of the PLS model with the confidence limits are done using the procedure reported in Nomikos and MacGregor (1994a). Figure 5.4 shows the concepts involved in the model building step. Data characterizing the normal operation of the batch unit are first organized and pretreated. This will ensure that the matrices are in a form suitable for processing by the PLS algorithm. The data blocks are unfolded (into two dimensional arrays) and are appropriately scaled (mean centered or autoscaled). If the batch runs are of unequal lengths, the incomplete part of the data set is filled up using the procedure outlined above - this automatically results in autoscaled matrices.

The multiblock PLS algorithm (Wangen and Kowalski, 1988) is used to extract the dominant dimensions in the process data. The procedure is remarkably similar to the ordinary PLS algorithm and generates the scores and loadings matrices for each block of data - the initial conditions, the primary and secondary variables and the quality block. In

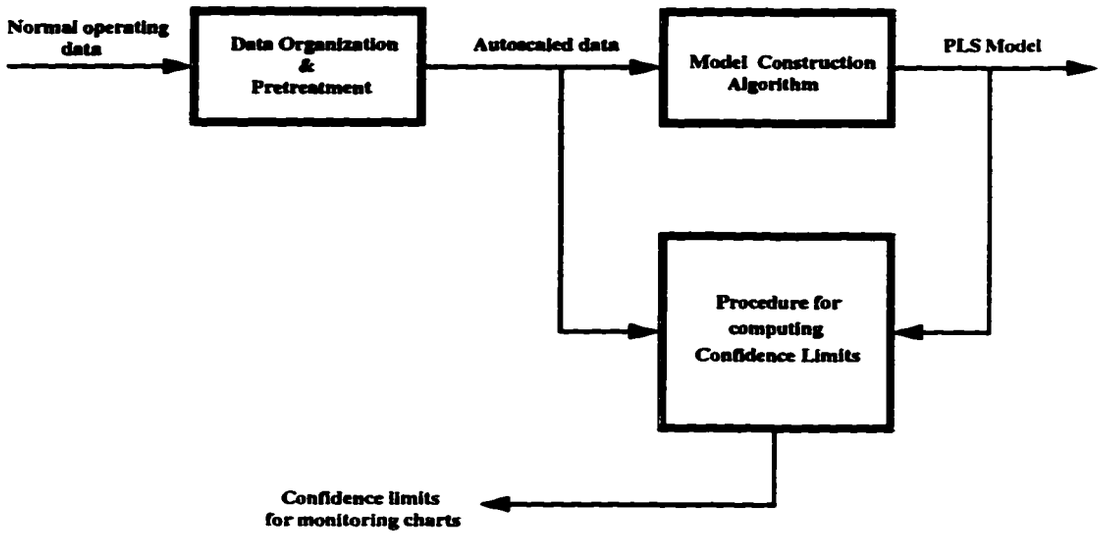

Figure 5.4: Framework for PLS Model Construction

addition to these matrices, the scores and loadings matrices are also obtained for a composite block that comprises of the initial conditions block, the primary variables block and the secondary variables block. These composite matrices can be assumed to be suitably weighted combinations of the individual scores and loading matrices. As in ordinary PLS, the number of PLS dimensions to be used in the model is decided based on the percentage of variance explained or by the use of statistically sound procedures such as cross validation. The logical structure and layout of the Wangen-Kowalski algorithm for our database structure is shown in Figure 5.5. Here, Z_1, Z_2, Z_3 and Z_4 represent the initial conditions block, the primary variables block, the secondary variables block and the quality block respectively. The methodology is described only for a single dimension. Latent variables (score vectors) generated from these blocks are labelled as t_1, t_2, t_3 and u_4 respectively. The score vectors, t_1, t_2 and t_3 are combined to give a composite score vector t_c. The *inner relationship* model is then obtained by the regression of t_c on u_4 (represented by the doubleheaded arrow in the Figure). For the next PLS dimension, the matrices Z_1, Z_2, Z_3 and Z_4 are deflated (as explained in chapter 1) and the above calculations are repeated.

The most important part of the model construction step is the derivation of the confidence limits for use in online monitoring. The scores trajectories obtained for each of the data blocks cannot be used to construct these limits. Online monitoring of batch processes are quite different from the monitoring of continuous processes and pose some interesting problems. There is no difficulty for the initial conditions block as all the information is

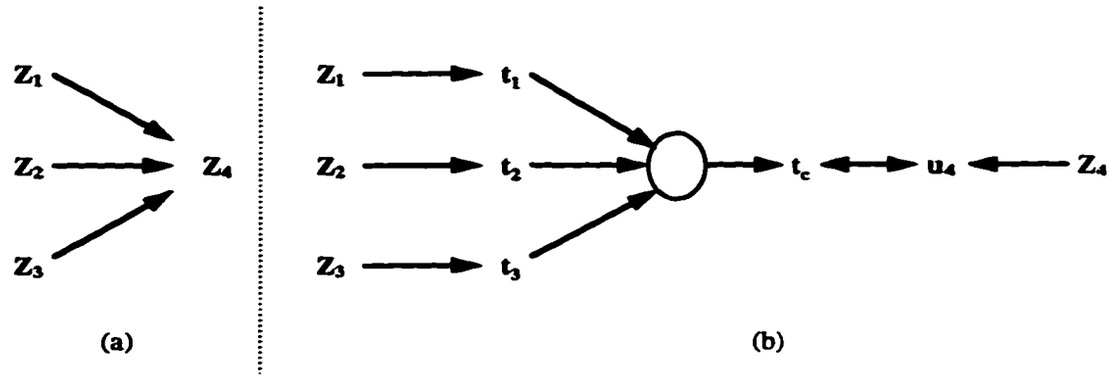

Figure 5.5: Schematic representation of the Wangen-Kowalski algorithm for a single PLS dimension : (a) Basic Relationship (b) Logical layout of the PLS algorithm

available. However, the primary and secondary variable information is not complete - at the start of the batch it is empty and gradually gets filled as the batch progresses from start to finish. This means that at any time instant, except at the end of the batch, a pragmatic guess of the future primary and secondary variables must be made. Only if these matrices are full can the PLS model be used to predict the final quality variables. Since the confidence limits for the score trajectories will be generated by passing each of the normal batches through the PLS model as though it were a currently operating batch, it is mandatory to decide the approach that will be employed to fill up the unknown data during online monitoring. Nomikos and MacGregor (1995) suggest three approaches to fill up the unknown data in the process variables vector. A brief summary of their guidelines is presented below :

- Approach 1 : This approach assumes that the future observations are in full agreement with the mean trajectories as calculated from the reference database. This means that we fill the autoscaled values (that is used in the M3PLS algorithm) with zeros. The result is a good graphical representation of the batch run but at the cost of the t-scores (see Nomikos and MacGregor (1995) for more details) being reluctant to flag an abnormal plant operation particularly at the start of a batch run.

- Approach 2 : An alternate approach is to assume that the future deviations from the mean trajectories are equal to the deviations noticed at the current time. This is done at each sampling time. With this approach, the t-scores are sensitive and pick up an abnormality more quickly.

- Approach 3 : The third approach capitalizes on the ability of PCA/PLS to handle

missing data. In the PCA/PLS literature, missing data can be filled up by restricting them to be consistent with the already observed values up to the current sample. This method appears superior to the other two approaches if at least 10% of the batch history is available for it gives large and unexplainable t-scores at the beginning of the batch. Also, the control limits calculated with this approach have constant trajectories in contrast to the earlier strategies. It is known that PLS can handle missing data, if the missing entries are few and randomly located in the data set. Consequently, this approach may not be appropriate for use in a scenario where there are too many missing entries that are not even randomly located.

Nomikos and MacGregor (1995) recommend careful employment of these approaches for process monitoring depending on the nature of the process. They report that the second approach generally works well in most cases. In this work, a slight variant of this approach is utilized. At each time instant, the future values are set equal to the autoscaled deviation values at the current sampling instant. This is easily justified because the autoscaled deviation values are used by the M3PLS model for monitoring and predictions. With this approach the nature and contour of the monitoring charts are very much akin to those obtained from the second approach of Nomikos and MacGregor (1995). Such an approach provides the external reference distribution (e.g. scores for each block and at each time interval) and facilitates calculation of the control limits. In doing so, it is assumed that the external distribution sufficiently captures the inherent variations observed in the database of acceptable process operations and will be applicable to assess new batch runs. The above procedure is indicated by the block labelled *Procedure for computing Confidence Limits* in Figure 5.4.

A prediction interval for a single future observation is an interval that will, with a specified degree of confidence, contain the next sample from the process. Assuming that the reference as well as the future data are random samples from the same parent population (having identical production procedures and similar process conditions), the prediction intervals may be computed (Hahn and Meeker, 1991). A two sided $100(1-\alpha)\%$ prediction interval to contain the mean of a future, independently and randomly selected observation, using the reference data containing an independent random sample of size n from the same process described by a normal distribution, is (*cf.* equation (4.2) of Hahn and Meeker, 1991) given as :

$$[UCL, LCL] = \bar{x} \pm t_{(1-\frac{\alpha}{2},n-1)}(1+\frac{1}{n})^{\frac{1}{2}}S \tag{5.1}$$

where UCL and LCL refer to the upper and lower control limits respectively, $\bar{x}$ denotes the estimated mean and S is the standard deviation computed from the reference distribution. The factor $t_{(1-\frac{\alpha}{2},n-1)}$ represents the critical values of the Student's t-distribution for a specified degree of freedom and confidence. Equation (5.1) can be used to obtain the confidence intervals for the scores concerning the initial conditions, primary variables and

the secondary variables. Charting quadratic forms such as the squared prediction errors (SPE's) require the computation of one-sided confidence limits. Equation (5.1) is suitably modified and used for this purpose (as in the case of chapter 1, the SPE values are computed using the difference between the actual and approximated sample values. See the section on *Tools for Online Process Monitoring*, Chapter 1). Usually, the 95% and the 99% confidence intervals are developed.

The confidence limits for the scores trajectories is readily available to monitor the scores for the entire duration of a new batch run. However, the confidence limits on the predicted quality variables are obtained online. The limits are calculated based on involved statistical concepts such as estimable functions and generalized inverses. Such issues are presented in Searle (1982). Some approximate expressions are provided by Phatak (1993) and Nomikos and MacGregor (1994a) - the former has a stronger theoretical basis but is computationally intensive and is restricted to data sets that have only one quality variable. The less accurate but more convenient approach of Nomikos and MacGregor is employed here. The confidence intervals for a predicted quality variable (denoted by $\hat{y}$) at each sampling instant is given by :

$$[UCL, LCL] = \hat{y} \pm t_{(1-\frac{\alpha}{2},k)} \, (MSE)^{\frac{1}{2}} (1 + \hat{t}_c^{'} (T_c^{'} T_c)^{-1} \hat{t}_c)^{\frac{1}{2}} \tag{5.2}$$

In equation (5.2), UCL and LCL denote the upper and lower confidence limits. $\hat{y}$ refers to the current predicted value of the final quality variable, $t_{(1-\frac{\alpha}{2},k)}$ denotes the critical value of the student's t-distribution for a confidence level α and k degrees of freedom. If there are n dimensions in the PLS model, the degrees of freedom k equals NB-n-1. $\hat{t}_c$ denotes the composite t-scores computed for the current batch and T_c stands for the composite T matrix obtained from the database of normal runs (during the model building step). The MSE is obtained from the procedure that is used to estimate the confidence limits and denotes the mean squared error. It is computed at each sampling instant via equation (5.3).

$$MSE = \frac{(y - \hat{y})^{'} (y - \hat{y})}{k} \tag{5.3}$$

Equation (5.2) is applied individually to each of the quality variables.

5.4.5 Online Monitoring

At this point, it is assumed that the following details are available from the data treatment and the model building step :

- Scaling information for each block at each sampling instant. This information is obtained from the data pretreatment step
- The PLS model : loading matrices for each block (including the composite block) and the inner model relating the composite scores to the output scores

- Confidence limit trajectories for score variables pertaining to initial condition, primary, secondary and the composite variables
- Confidence limit trajectories for the squared prediction errors (SPE's) in initial condition, primary and secondary variables
- The MSE values (computed using equation (5.3) for each sampling instant
- The composite matrix, T_c

At each sampling instant (major and minor), the data obtained are first scaled in exactly the same way the normal data were scaled in the model building step. The major sampling instant denotes the case when measurements of both the primary and secondary variables are available. The minor sampling instant denotes the sampling periods when only the secondary measurements are available. Then, the *unknown* data is filled up using any of the approaches recommended earlier. Care must be taken to see that the same approach that was used in the model building step (procedure to compute the confidence limits) is followed. The scores and the SPE's corresponding to each block are computed using the available data and the loading matrices. All of this information can be projected to their respective monitoring charts. With the composite scores, the inner model and the loadings matrices, the final quality variables ($\hat{y}$) can be predicted. Now, equation (5.2) can be used (individually for each quality variable) to determine the prediction intervals for the final quality variables. If the scores or SPE's for the process variables (initial conditions, primary or secondary) indicate an out-of-control status, it is advisable not to rely on the predictions of the final quality variables given by the PLS model. An abnormal situation may indicate that the PLS model is no longer representative of the current process behavior and too much emphasis should not be placed on the predictions obtained.

A schematic representation of the steps involved in online monitoring of a new batch is shown in Figure 5.6. In the next section, two case studies are presented in order to illustrate the concepts presented above.

5.5 Case Studies

Two case studies are considered here. The first case study involves a fed-batch antibiotic producing fermentation reactor and the second represents results based on data made available from a simulated polymerization reactor[4].

[4]The data was provided by Prof. MacGregor's group, McMaster University, Hamilton, Ontario, Canada. The help and advice offered by Prof. MacGregor, Dr. Paul Nomikos and Dr. T. Kourti is gratefully acknowledged

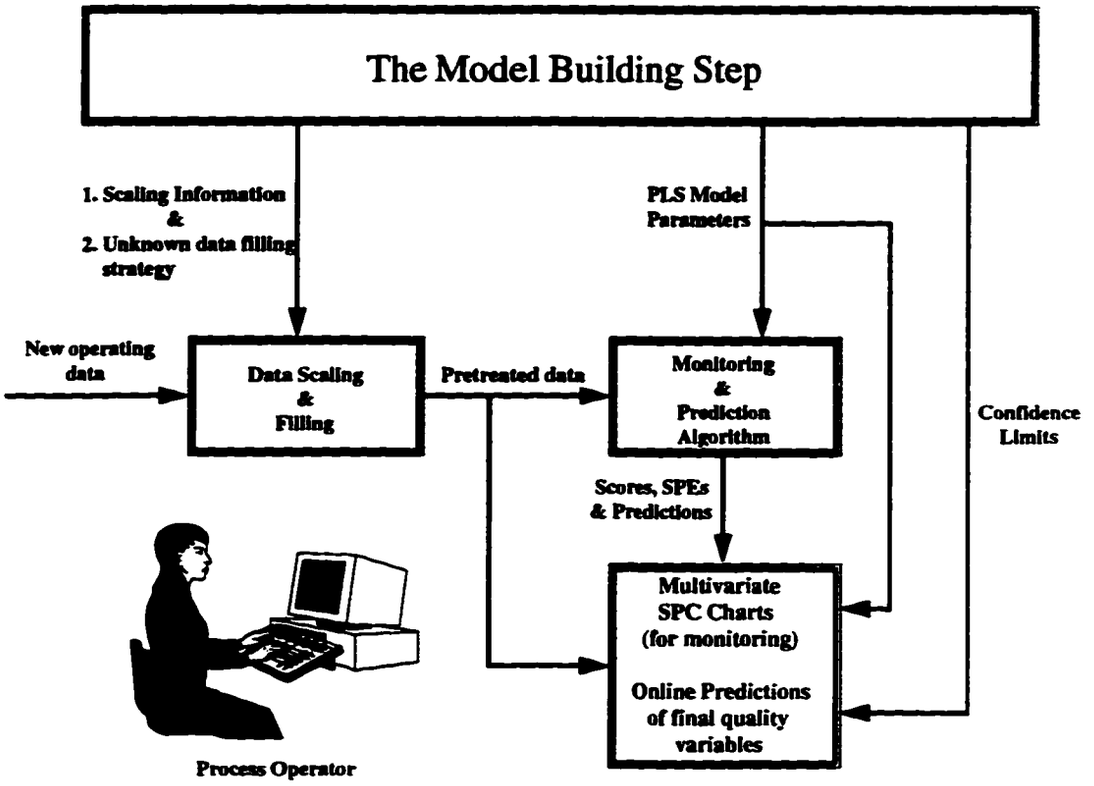

Figure 5.6: Schematic representation of PLS based online monitoring for batch processes : The Nomikos-MacGregor approach

5.5.1 Case Study 1 : The Fed-batch Bioreactor

The measurements available from the bioreactor and their segregation into various blocks have been described in considerable detail earlier. The system was modelled in sufficient detail by using equations that describe the substrate inhibition effects on the biomass growth and antibiotic production through the use of different empirical models (Bajpai and Reuss 1980). Due to the presence of significantly different time scales (the gas phase dynamics are significantly faster than the broth phase dynamics), a stiff differential equation solver was used to solve the model equations (the interested reader is referred to Gudi (1995) for more details on the model and the simulation procedure). A total of 47 simulations incorporating the common batch to batch variations were performed by using different initial conditions to yield a database of normal batch runs. The run lengths of the batches varied anywhere between 100 and 120 hours. The profiles of the secondary process variables that constitute the SVB were sampled every 30 minutes while the primary process variables in the PVB were sampled every 150 minutes. White Gaussian noise with zero mean was added to the measurements to simulate noisy measurements with a relatively smaller signal to noise ratio.

The database generated from the above simulations was used to obtain the PLS model

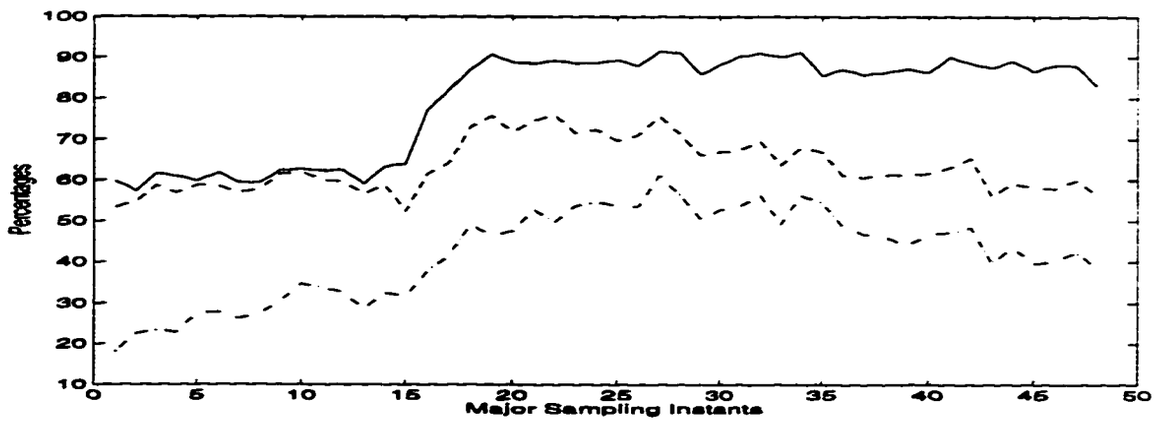

Figure 5.7: Cumulative percentage sum of squares utilized for the PLS model as a function of the number of dimensions and time : Primary Block

and the monitoring charts for the PLS based algorithm. The data was pretreated first: as explained earlier, this consisted of data filling, scaling and unfolding. The Wangen-Kowalski algorithm was used to obtain the multiblock PLS model which had a relationship structure given by Figure 5.5(a). Three PLS dimensions were found to explain about 97.5% of the variability in the quality block. The dimension of the PLS model was therefore fixed at three. The percentage sum of squares utilized from the initial conditions block, the primary variables block and the secondary variables block were 76.4%, 63.2% and 50.3% respectively. The cumulative percentage (sum of squares) utilized from the primary and secondary blocks as a function of the number of PLS dimensions and time is given in Figures 5.7 and 5.8 respectively. The dash-dot line represents the cumulative percentages used up by the first dimension, the dashed line represents the cumulative percentages utilized by the first *and* second dimensions and the solid line represents the total primary and secondary block information used in the prediction of the quality variable. From these Figures, it is seen that as time progresses more of the primary block information (about 90%) and less of secondary block information (about 50%) is used up to predict the final quality. Also, the third PLS dimension for the primary block employs considerably more information from the end of the batch: in the earlier part (see Figure 5.7) no improvement is visible after the second dimension. For the secondary block, the third dimension uses up information from the middle stages of the fermentation reaction to model the output variable.

The inner relationship plots for the first two dimensions is given in Figure 5.9. Note that the composite block scores are used to represent the input scores. All the batches are found to lie close to the 45^o line indicating that the linear PLS model is adequate (since no nonlinear trends are observed). The correlation for the linear fit is also shown in the Figures. The high correlation coefficient for the second dimension points to the fact that this dimension is very useful in predicting the output (quality) variable.

The monitoring algorithm was first evaluated by presenting online data from a *new*

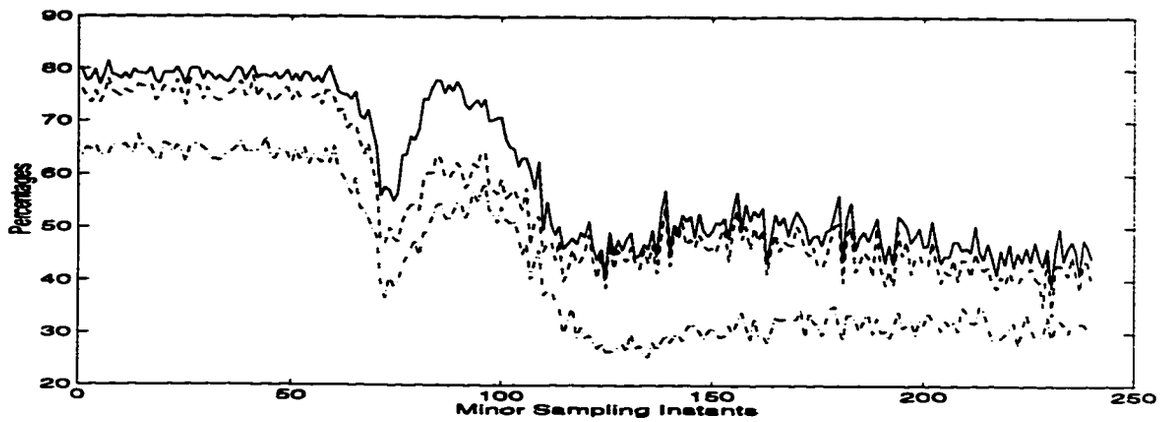

Figure 5.8: Cumulative percentage sum of squares utilized for the PLS model as a function of the number of dimensions and time : Secondary Block

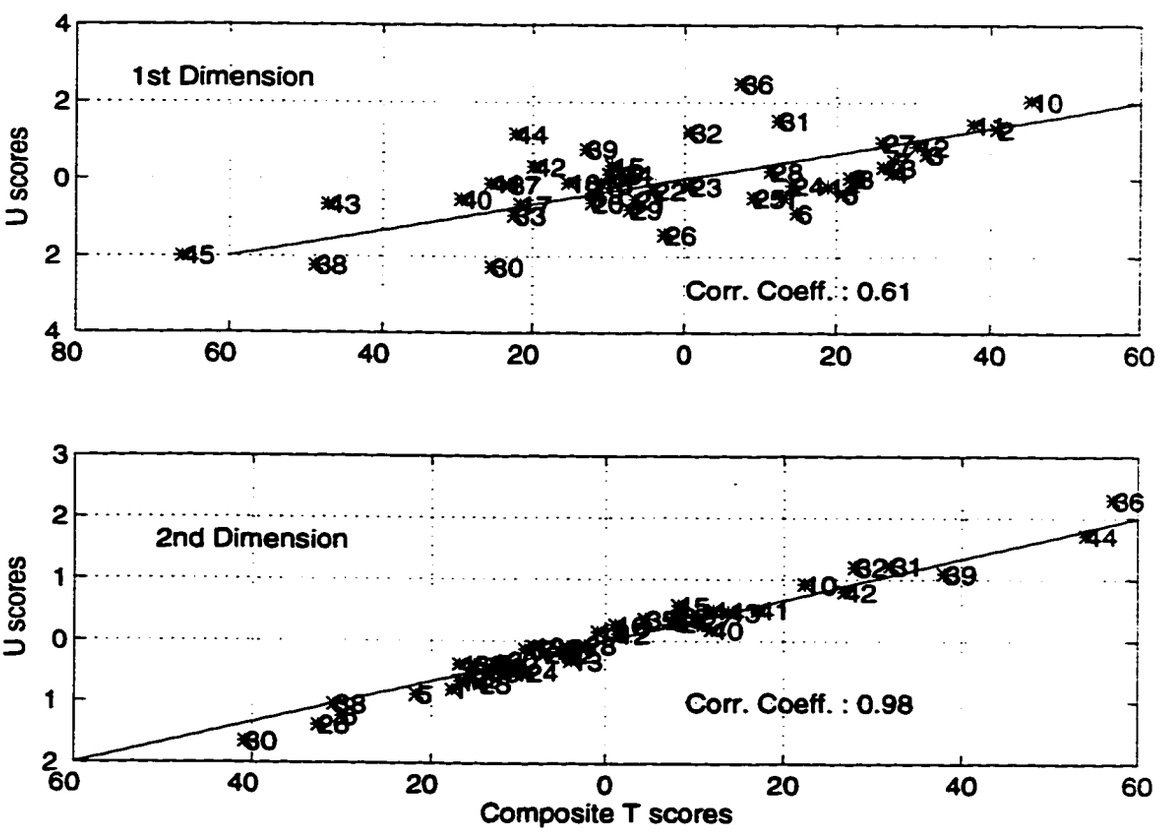

Figure 5.9: The inner relationship plots for the multiblock PLS model

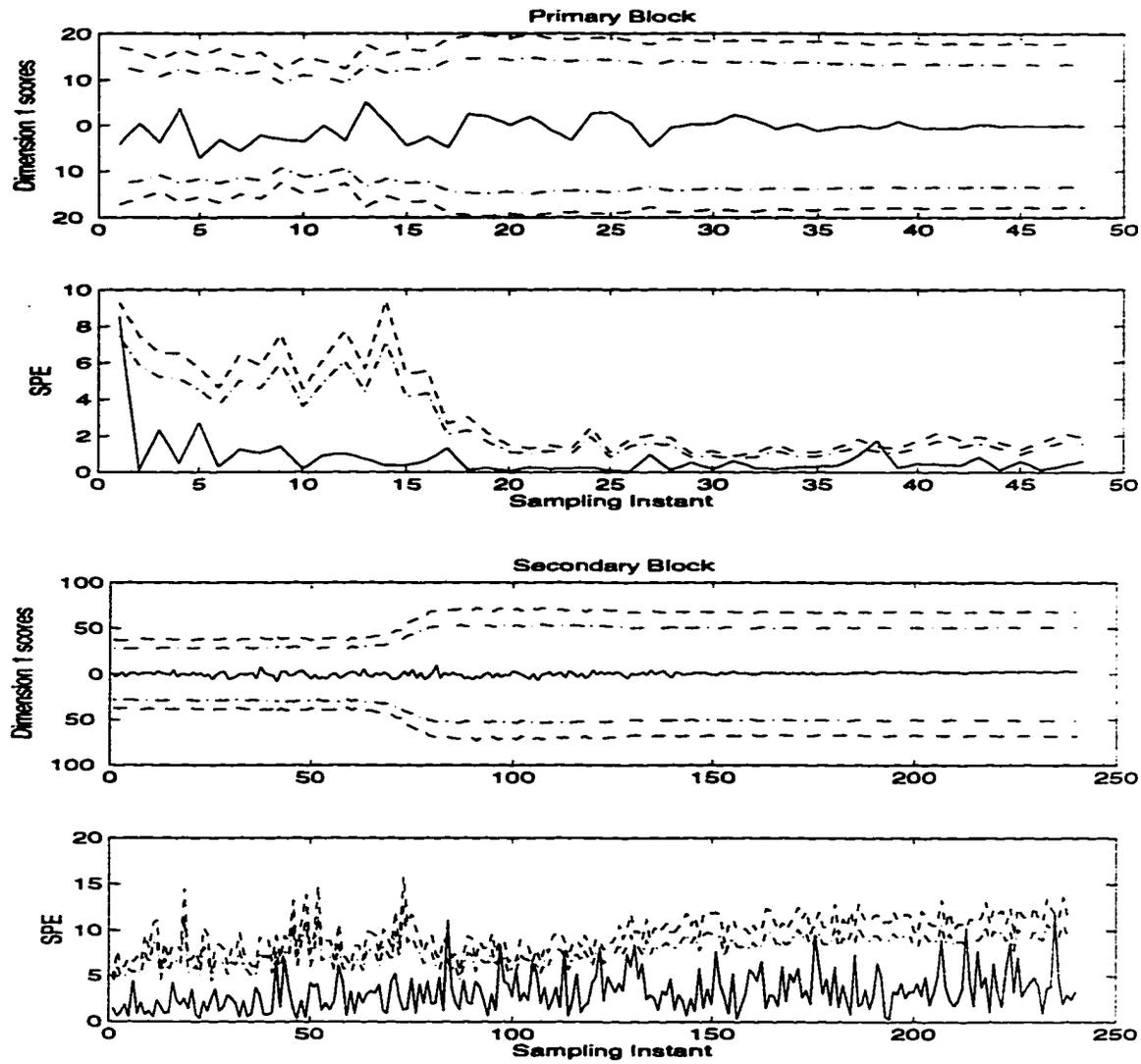

Figure 5.10: Online monitoring for a normal batch : Fermentation process

normal fermentation run. The scores and SPE trajectories (for the primary and secondary blocks) as well as the online predictions for the quality variable (final antibiotics concentration) are shown in Figures 5.10 and 5.11. It is seen that the trajectories lie within the 95% confidence limits throughout the duration of the batch. The final quality predictions are quite close to the actual value that was obtained at the end of the simulation run (indicated by the '*' symbol). In these Figures, the dash-dot line represents the 95% confidence limits. the dashed trajectories are the 99% confidence limits and the solid line shows the actual batch trajectory.

The following faults or deviations that are commonly encountered in fermentations were implemented in the simulations and the resulting data sets were presented to the algorithm to evaluate its capability to perform on-line fault detection and quality predictions.

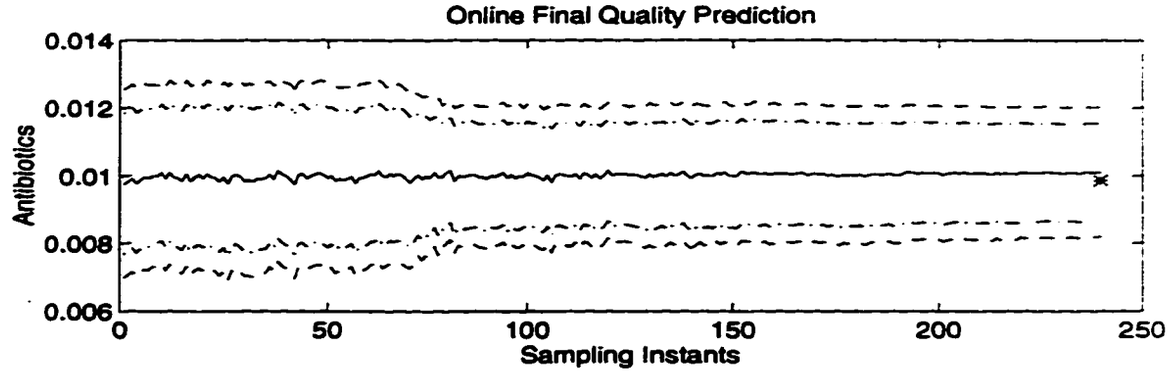

Figure 5.11: Online final quality predictions for a normal batch : Fermentation process

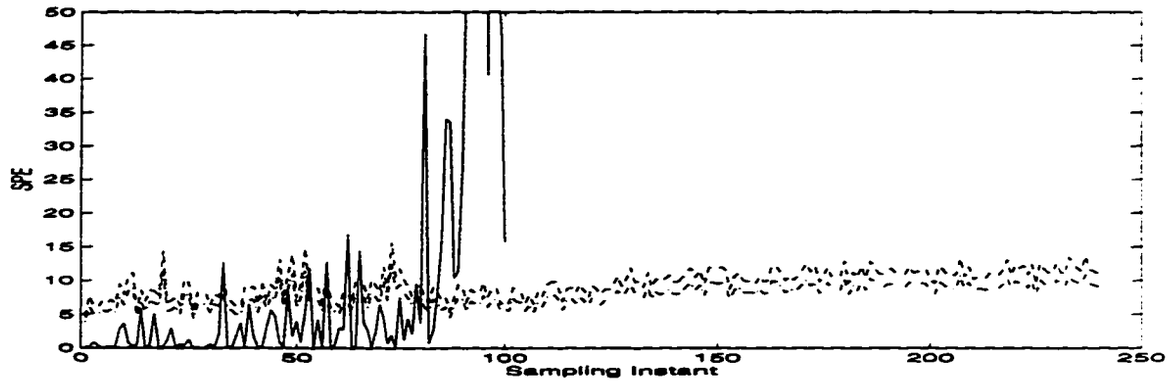

Figure 5.12: SPE trajectory of secondary variables block for abnormal run 1 : Fermentation process

1. Contamination by foreign microorganisms : An additional contaminant state to characterize the biomass evolved due to the growth of the foreign microorganism was introduced into the fermentor equations. The contaminant microorganism was assumed to grow at a constant specific growth rate that was higher than the maximum specific growth rate that could be attained in the normal fermentation (Chattaway and Stephanopoulos, 1989). The foreign microorganism affected the normal fermentation profiles by taking up nutrients, oxygen and by evolving CO_2. It also affected the final antibiotic titre through its effect on the environmental variables. The concentration of the contaminant microorganism also showed up in the offline measurements of the biomass. Figure 5.12 shows that the abnormality is detected almost immediately after the contamination is introduced (80^{th} sampling instant). The fermentation run was *discontinued* after the 100^{th} minor sampling instant.

2. Sparger disturbances : Changes in the environmental variables, such as the CER and dissolved oxygen, resulting from sparger disturbances were simulated by manipulating the gassed power to the fermentor. A disturbance was introduced at 20 hours of fermentation time and removed after 23 hours of fermentation. The fermentation

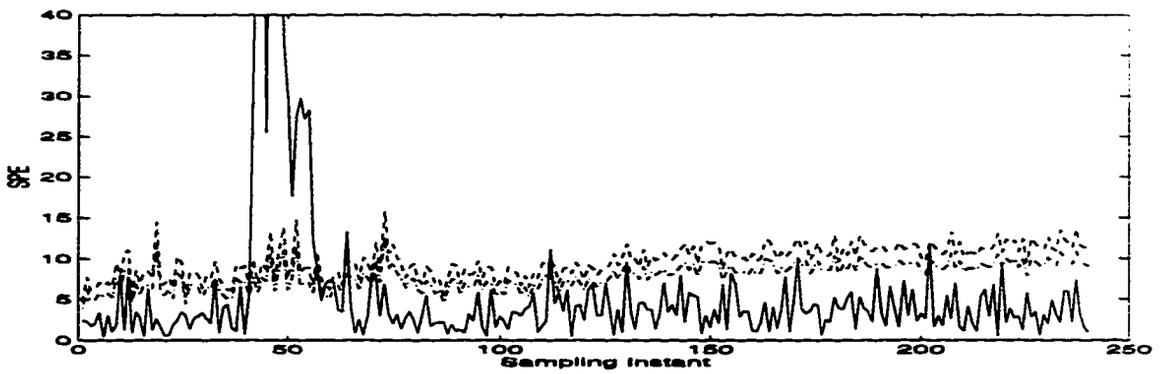

Figure 5.13: SPE trajectory of secondary variables block for abnormal run 2 : Fermentation process

operation returned gradually to the normal operating regime at about 50 hours of fermentation. Figure 5.13 shows the partial results of monitoring such a batch. The deviation from normal plant behavior on the introduction of the disturbance and the gradual return to normal plant operation after the disturbance is removed can be seen in the SPE plot of the secondary block.

5.5.2 Case Study 2 : The Batch Polymerization Reactor

As a second example, the monitoring of a semibatch reactor that produces Styrene Butadiene Rubber (SBR) will be illustrated. A detailed mechanistic model was used by Nomikos and MacGregor (1994a) to produce a database of normal operations consisting of 50 batches. Three more data sets that included a *normal batch* and two batches with faults were also generated by them. Autocorrelated variations and noise have been included in the data to impart real-life character to the data. These data sets were made available to other researchers on request and have since been used as a benchmark in related research (Dong and McAvoy (1994)). Frequent measurements of the feed rates of styrene and butadiene monomers, temperatures of the feed stream, the reactor contents, the cooling water and the reactor jacket are assumed. Latex density, total conversion and the instantaneous heat release from an energy balance are sampled at a lower sampling rate (ratio of frequencies 5:1). The primary variables block is a $(50 \times 3 \times 40)$ array and the secondary variables block is of dimension $(50 \times 6 \times 200)$. Initial condition data are not available for this case study. The quality block has 5 variables : composition (percent styrene), particle size, branching, crosslinking and polydispersity and is of dimension (50×5). Run lengths in the *normal database* were between 180 and 200 samples (minor sampling instants).

After the customary data pretreatment, the PLS model was constructed. Three PLS dimensions were found to be adequate via the process of cross validation. About 59% of the primary block sum of squares and 17.5% of the secondary block sum of squares accounted for approximately 69% of the sum of squares of the quality block. Amongst the

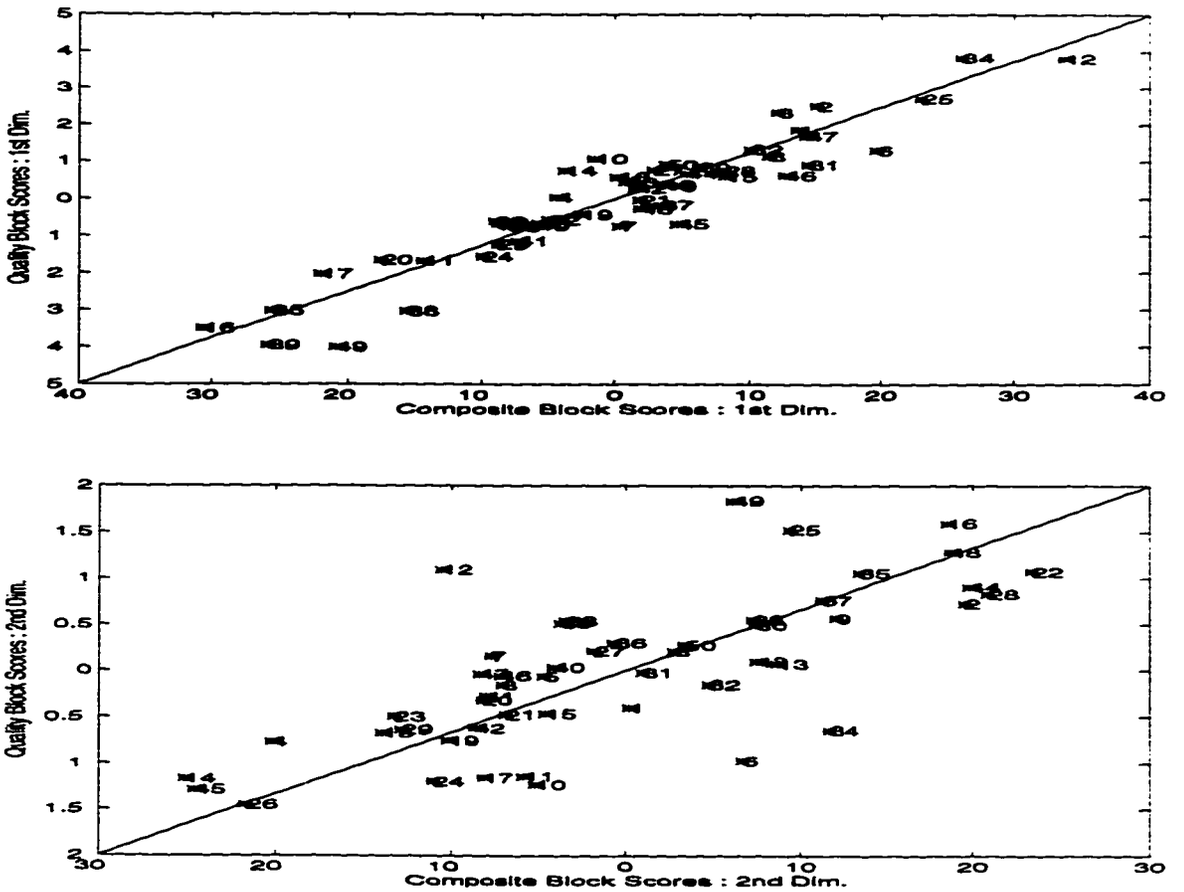

Figure 5.14: The PLS inner model : Polymerization process

quality variables, more than 80% of quality variables 3,4 and 5 (branching, crosslinking and polydispersity) were explained; 62% of quality variable 1 (percentage styrene) and only 35% of quality variable 2 (particle size) were accounted for. Inclusion of more PLS dimensions may be considered if the latter quality variables are to be fitted more closely.

The inner relationship plots given in Figure 5.14, indicates a good degree of correlation between the composite input scores and the output scores. The samples lie close to the diagonal line and implies that a model with good predictive capability has been obtained. No nonlinear trends are observed in these plots.

Figure 5.15 shows the monitoring charts (first dimension scores and SPE trajectories for the primary and secondary blocks) from the online monitoring of a good batch. The broken lines (in this and the following Figures) indicate the confidence limits and the '+' sign represents the value obtained from the current batch. There are no *statistically significant excursions* from normal plant behavior as is evident from the charts.

Data from an abnormal batch was also made available. In this case, impurities were introduced in the butadiene monomer feed to the reactor at the 100^{th} sampling instant. The results from the online monitoring strategy are given in Figure 5.16. Though the first

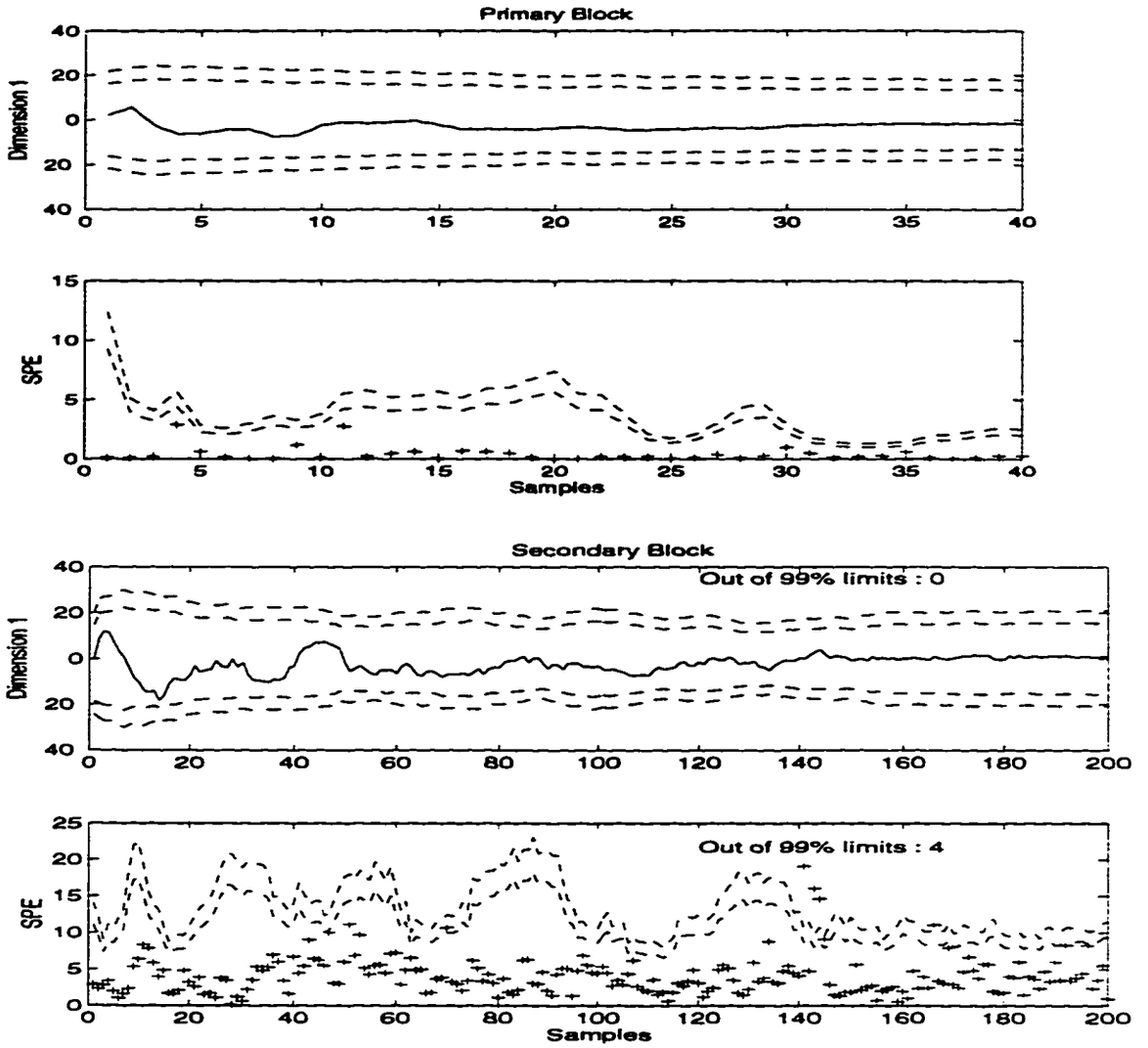

Figure 5.15: Online monitoring for a normal batch : Polymerization process

dimension scores are reluctant to pick up the fault, the SPE values for both the primary and secondary blocks flag this abnormality almost immediately. This once again brings into focus the utility of the SPE for process monitoring purposes (in tune with chapter 1). Once a fault has occurred, the predictions made for the final product quality becomes less reliable (the final quality predictions are not included because they convey very little).

It is common knowledge that the cooling water temperature and the jacket temperature are highly correlated and move together. For illustrative purposes, the cooling water temperature was increased and the jacket temperature was decreased taking care to see that the values were always within the maximum and minimum values found in the database representing normal plant operations. Such an exercise need not be necessarily viewed as a mathematical artifice; in real situations, this could represent a faulty sensor. Figure 5.17 shows the resulting trajectories for the resulting batch. The solid lines indicate the max-

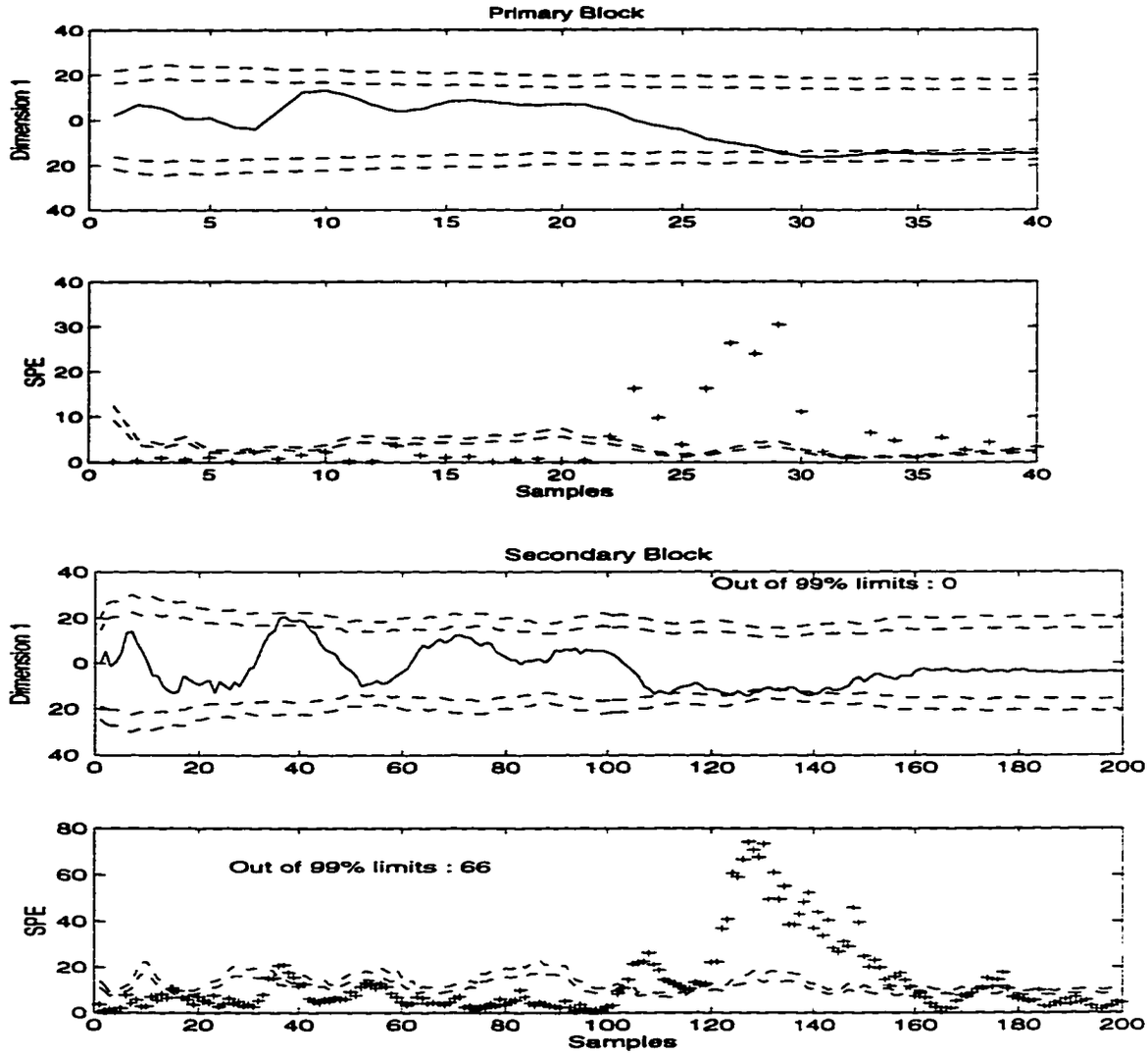

Figure 5.16: Online monitoring for faulty batch 1 : Polymerization process

imum and minimum trajectories obtained from the database of normal runs. The broken trajectory corresponds to the new operating batch. It is clear that if only the individual variables are monitored misleading conclusions can be drawn - the faulty sensor(s) may go unnoticed or the final product may not meet the specifications resulting in the loss of resources.

The PLS based multivariate monitoring strategy is able to detect the abnormality in the operation. Figure 5.18 shows the results; it is interesting to see that since the cooling water temperature and the jacket temperature belong to the secondary variables block. the primary block scores and SPE are insensitive to the fault.

To pinpoint the underlying reason for the fault, contribution plots were developed (see chapter 1) and presented in Figure 5.19. The SPE contributions clearly indicate the problem with secondary variables 5 and 6 (i.e.. the cooling water and jacket temperature). A look

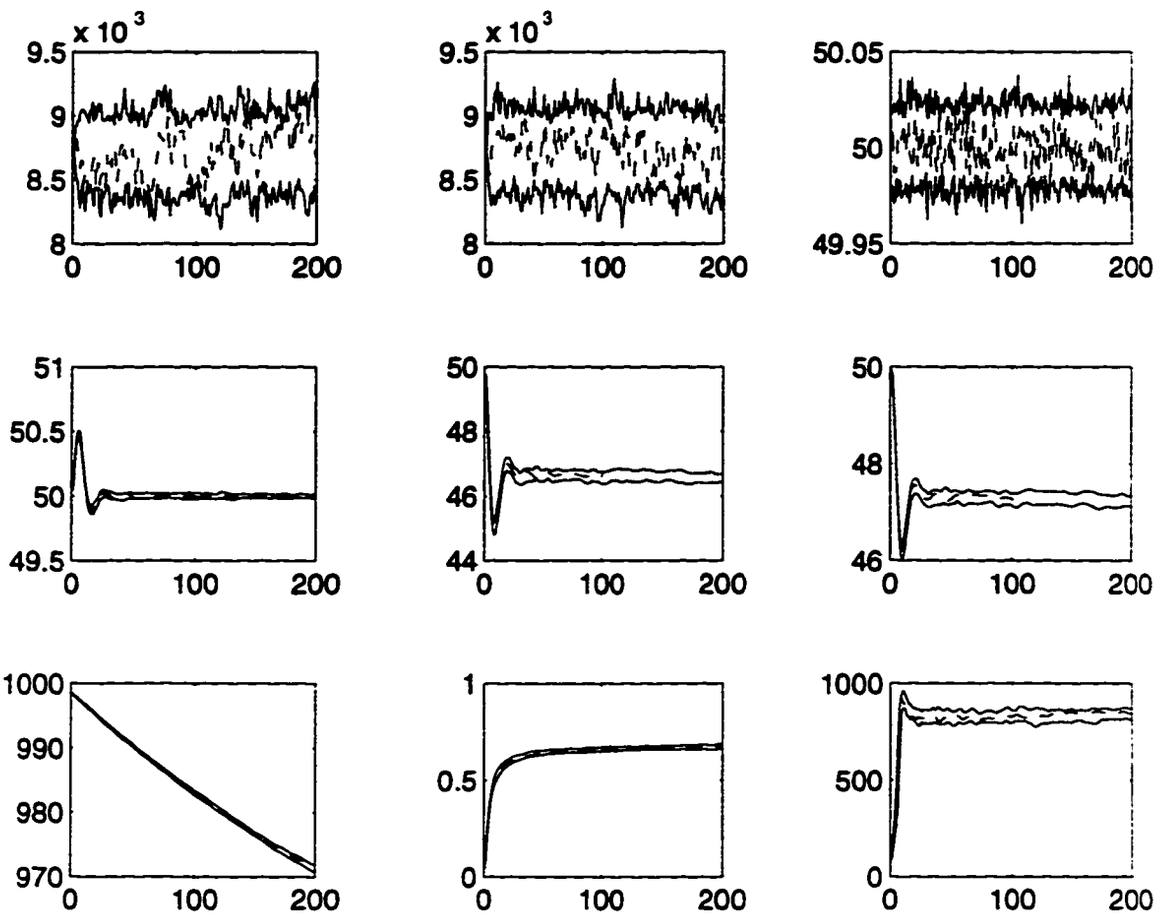

Figure 5.17: Trajectories of individual process variables over the entire duration of the batch run (200 samples)

at the contributions to the scores indicate that these variables have moved in opposite directions thus destroying the normal correlation between them. This indicates a problem with either of the two sensors.

5.6 Conclusions

This chapter reviewed a databased methodology for process monitoring and fault detection of batch and semibatch processes in a SPC framework. The strategy is very general and can be easily adapted to a variety of practical situations. The approach of Nomikos and MacGregor was extended to include the multirate sampling scenario. Some suggestions were also provided to handle *incomplete* databases. Rapid detection and isolation of process faults were illustrated using two simulation based case studies.

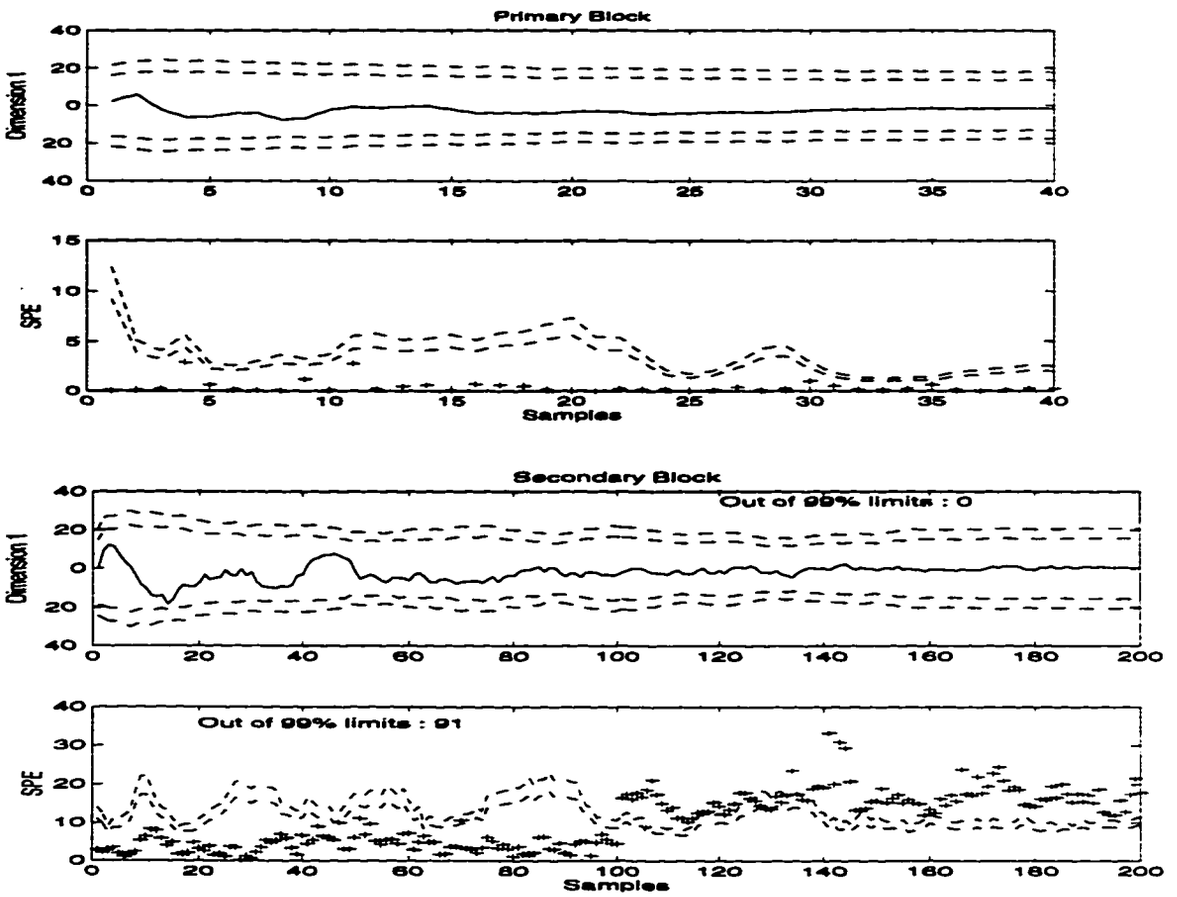

Figure 5.18: Online monitoring for faulty batch 2 : Polymerization process

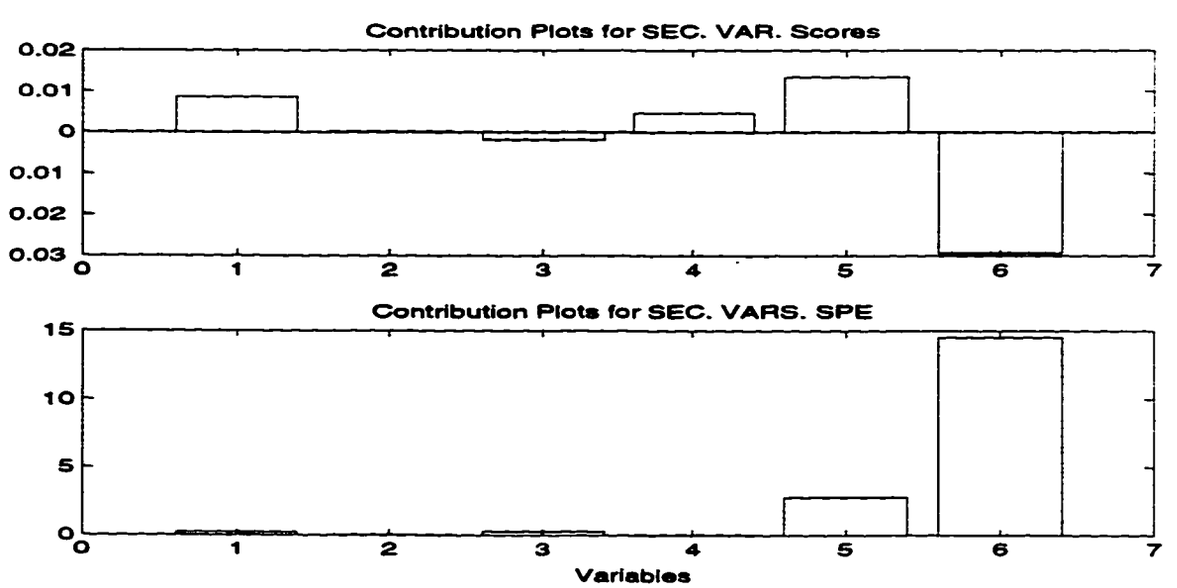

Figure 5.19: Contribution plots for the secondary variables : Scores and SPE

Chapter 6

Conclusions

This thesis has explored the application of multivariate statistical tools for a variety of process engineering situations - identification, control and monitoring. The suitability of principal components analysis and partial least squares for the monitoring and fault detection of continuous processes is well known (e.g. Kresta (1992), Wise (1991)) and is now a fairly mature area. Two industrial case studies were considered in Chapter 1 in order to gain experience in the use of these tools for real applications. The PLS based inferential model obtained in Chapter 1 for the distillation column has yielded better product quality at Mitsubishi chemicals, Japan. The CVA method of identifying black-box state space models was described in Chapter 2. The canonical correlations analysis technique was employed here to identify the *optimal* states of the model. It has been shown here and in other studies (via extensive simulations) that the CVA method is much reliable compared to other subspace identification methods. The CVA technique was also extended to the domain of a useful class of nonlinear systems, namely, the Hammerstein model. In Chapters 3 and 4. PLS based dynamic models were identified from plant data for both linear and nonlinear systems. The approach proposed by Kaspar and Ray (1992, 1993) served as the motivating factor for this study. The PLS loading matrices and the identified nonlinear components were used as compensating elements in the design of the PLS based control strategy. It was shown that, for multivariable processes, the identification and control tasks can be done solely based on the well understood SISO (single input single output) theory. To compensate for measured disturbances, the design of feedforward controllers (in terms of the latent space) was also considered for both linear and nonlinear systems. Very often, large improvements in regulatory control can be realized by employing only one PLS based feedforward compensator. Constrained model based predictive control was also implemented using the PLS models (integration of the PLS modelling and the DMC control strategy) for both linear and nonlinear systems to make it more suited for real world applications.

Finally, the PCA/PLS based batch process monitoring strategy proposed by Nomikos and MacGregor (1994a, 1994b, 1995) was extended to handle multirate process measurements and to obtain the PLS model from databases containing batches of varying run lengths.

6.1 Contributions of this thesis

The key contributions of this study can be listed as follows :

- Two industrial case studies involving PLS and PCA are presented to demonstrate the utility of these techniques for inferential model building and process monitoring
- Superiority of the CVA technique over N4SID was established via extensive simulations. The CVA technique was also used to model and control a laboratory stirred tank heater.
- The CVA technique was extended to identify multivariable Hammerstein systems. This work represents the first step towards extending the purview of the subspace methods to include these models - the MOESP algorithm (another subspace method) has been very recently extended (by its original proponents) to identify multivariable Hammerstein models. This also indicates the growing interest in the use of subspace identification methods.
- A PLS based modelling and control strategy that is applicable to both linear and nonlinear systems has been proposed. The theoretical developments were supported by simulation and experimental studies.
- Explicit expressions were provided for the design of feedforward controllers in the PLS latent space. Each element of the multivariable feedforward controller can be obtained as the ratio of two transfer function elements (similar to the SISO feedforward design case). These controllers are expected to provide *high performance* feedforward compensation to counter the effect of measured disturbances.
- The well known Cramer's rule (that expresses the solution of a square system of linear equations as the ratio of determinants) has been extended to include nonsquare equation systems. Though not of any practical utility, this spin-off result (from the design of PLS based feedforward controllers) has served to rekindle my passion for *old-fashioned mathematics.*
- Real process control applications must deal with constraints on process variables. Model predictive controllers have made this task simple (via online constrained optimization). The PLS based models with their *control friendly* structure would be useful for practical applications only if they were integrated with the model predictive control schemes. The investigation carried out in this direction has yielded positive

results indicating the viability of the PLS-DMC scheme. Of particular interest are the issues related to the mapping of constraints.

- Some practical extensions were made to the PLS based batch/semibatch process monitoring strategy pioneered by Nomikos and MacGregor. Provisions were made to handle the multiple rates of measurement as well as to obtain the PLS model from databases that contain batches of varying run lengths. Simulation examples were used to illustrate the concepts.

6.2 Recommendations for future work

As with any research project, the work presented here raises some questions and provides scope for further studies. These are briefly summarized below.

- The optimality of CVA has always been established (here and in other studies) using extensive simulations. It may be interesting to conduct a fundamental study into the theory of subspace identification methods and pinpoint the reasons for the optimality of CVA and the relative performance of the subspace methods.

- There has been considerable emphasis on the open loop identification problem using CVA. It remains to be seen if the power of CVA can be utilized for multivariable closed loop identification.

- Bilinear state space models serve as useful representation of processes such as the paper machine headbox and high purity distillation columns. Research can be initiated to extend the CVA method for the identification of bilinear systems.

- Incorporation of more complex nonlinear structures (e.g. nonlinear time series models. volterra kernels etc.) in the PLS inner relationship may be required to capture more severe dynamic and steady state nonlinearities that may be present in plant data.

- Design of control systems based on the *nominal* PLS model was illustrated in this thesis. It may be worthwhile to obtain a description of the uncertainty in the model and use it for robust control system design. Investigation of the robust stability and performance of this control scheme may be taken up as a research project.

- The PLS based feedforward controllers have not been tested on a physical system. Real-time implementation of the feedforward compensator will indicate if the proposed strategy lives up to its promise as a *high performance feedforward controller.*

- Determination of the interactor matrix, a multivariable generalization of the time delay, is critical for the performance assessment of multivariable control loops (based on the minimum variance controller as the benchmark). The PLS modelling strategy

may provide a convenient route to obtain the interactor matrix from input-output data.

- The PLS based monitoring strategy may also be extended to cover *event driven* operations

References

"ADAPTx". (1992). Adaptics, Inc., MA (USA).

Akaike, H. (1976). "Canonical Correlation Analysis of Time Series and the use of an Information Criterion". *Pages 27–96 of:* Mehra, R., and Lainiotis, D. (eds). *"System Identification : Advances and Case Studies "*. Academic Press, NY, USA.

Anbumani, K., Patnaik, L.M., and Sharma, I.G. (1981). "Self-Tuning Minimum Variance Control of Nonlinear Systems of the Hammerstein Model". *IEEE Trans. autom. Control*, **AC-26**, 959–961.

Bajpai, R.K., and Reuss, M. (1980). "A Mechanistic Model for Penicilin Production". *Journal of Chemical Technology and Biotechnology*, **33**, 332.

Ben-Israel, A. (1982). "A Cramer Rule for Least-norm Solution of Consistent Linear Equations". *Linear Algebra and its Applications*, **43**, 223–226.

Bhat, J., Chidambaram, M., and Madhavan, K.P. (1990). "Adaptive Feedforward Control of Hammerstein Nonlinear Systems". *Int. J. Control*, **51**, 237–242.

Biegler, L.T., and Rawlings, J.B. (1991). "Optimization Approaches to Nonlinear Model Predictive Control". *In:* Arkun, Y., and Ray, W.H. (eds), *"Chemical Process Control - CPCIV"*. AIChE, NY, USA.

Billings, S.A., and Fakhouri, S.Y. (1979). "Nonlinear System Identification using the Hammerstein Model". *Int. J. Syst. Sci.*, **10**, 567–578.

Box, G.E.P., and Jenkins, G.M. (1976). *"Time Series Analysis, Forecasting and Control"*. Holden-Day, San Francisco, USA.

Candy, J.V., Bullock, T.E., and Warren, M.E. (1979). "Invariant Description of the Stochastic Realization". *Automatica*, **15**, 493–495.

Chang, F.H.I., and Luus, R. (1971). "A Noniterative Method for Identification using Hammerstein Model". *IEEE Trans. autom. Control*, **AC-16**, 464–468.

Chattaway, T., and Stephanopoulos, G. (1989). "Adaptive Estimation of Bioreactors : Monitoring Plasmid Instability". *Chemical Engng. Sci.*, **44**(1), 41.

Cinar, A. (1994). "Nonlinear Time Series Models for Multivariable Dynamic Processes". *In: INCINC '94*.

Clarke, D.W., Mohtadi, C., and Tuffs, P.S. (1987). "Generalized Predictive Control - Part I. The Basic Algorithm". *Automatica*, **23**, 137–148.

Cutler, C.R., and Ramaker. B.L. (1980). Dynamic Matrix Control - A Computer Control Algorithm. *In: Proceedings of American Control Conference.* American Control Conference, San Francisco. CA, USA.

Dayal, B. (1996). *"Multivariate Statistical Regression Methods for Process Monitoring and Experimental Design"*. Ph.D. thesis, McMaster University.

de Jong, S. (1993). "SIMPLS : an alternative approach to Partial Least Squares Regression". *Chemometrics and Intelligent Laboratory Systems*, **18**, 251–263.

Deistler, M., Peternell, K., and Scherrer, W. (1995). "Consistency and Relative Efficiency of Subspace Methods". *Automatica*, **31**(12), 1865–1875.

Dong D., and McAvoy, T.J. (1994). "Batch Tracking via Nonlinear Principal Components Analysis". *In: Proceedings of the AIChE National Meeting, San Francisco.*

Eskinat, E., Johnson, S.H., and Luyben, W.L. (1991). "Use of Hammerstein Models in Identification of Nonlinear Systems". *A.I.Ch.E. J.*, **37**(2), 255–268.

Gabriel Cramer. (1750). *"Introduction à l'Analyse des Lignes Courbes Algébriques"*. Geneva.

Gallman, P.G. (1976). "A Comparison of Two Hammerstein Model Identification Algorithms". *IEEE Trans. autom. Control*, **AC-21**, 124–126.

García, C.E., and Morari, M. (1982). "Internal Model Control 1. A Unifying Review and Sone New Results". *Ind. Eng. Chem. Process Des. Dev.*, **21**, 308–323.

García, C.E., Prett, D.M., and Ramaker, B.L. (1990). "Fundamental Process Control". *In:* Prett D.M., García, C.E., and Ramaker, B.L. (eds). *"The Second Shell Process Control Workshop : Solutions to the Shell Standard Control Problem"*. Butterworth Publishers, MA, USA.

Geladi, P., and Kowalski, B.R. (1986a). "Partial least-squares regression : A tutorial". *Analytica Chimica Acta*, **185**, 1–17.

Geladi, P., and Kowalski, B.R. (1986b). "An Example of 2-Block Predictive Partial Least-Squares Regression with Simulated Data". *Analytica Chimica Acta*, **185**, 19–32.

Gevers, M., and Wertz, V. (1982). "On the Problem of Structure Selection for the Identification of Stationary Stochastic Processes". *In: Proceedings of the Sixth IFAC Symp. Ident. System Param. Estimation.*

Gudi, R.D. (1995). *"Multirate Estimation, Control and Monitoring of Fed-batch Fermentations"*. Ph.D. thesis, University of Alberta.

Hahn, G.J., and Meeker, W.Q. (1991). *"Statistical Intervals : A Guide for Practitioners"*. John Wiley and Sons, NY.

Haist, N.D., Chang, F.H.I., and Luus, R. (1973). "Nonlinear Identification in the Presence of Correlated Noise using a Hammerstein Model". *IEEE Trans. autom. Control*, **18**, 552.

Henson, M.A., and Seborg, D.E. (1994). "Adaptive Nonlinear Control of a pH Neutralization Process". *IEEE Trans. Control Syst. Tech.*, **2**(3), 169–182.

Hernández, E., and Arkun, Y. (1993). "Control of Nonlinear Systems using Polynomial ARMA Models". *A.I.Ch.E J.*, **39**(3), 446–460.

Höskuldsson, A. (1988). "PLS Regression Methods". *Journal of Chemometrics*, **2**, 211–228.

Hsia, T.C. (1968). "Least Squares Method for Nonlinear Discrete System Identification". *Pages 423–426 of: Proceedings of the Second Asilomar Conference on Circuits and Systems.*

Hsia, T.C. (1976). "A Multi-Stage Least Squares Method for Identifying Hammerstein Model Nonlinear Systems". *Pages 934–936 of: Proceedings of the IEEE Conference on Decision and Control.*

Hurvich, C.M., Shumway, R., and Tsai, C.L. (1990). "Improved Estimators of Kullback Leibler Information for Autoregressive Model Selection in Small Samples". *Biometrika*, **77**, 709–719.

Jackson, J.E., and Mudholkar, G.S. (1979). "Control Procedures for Residuals Associated with Principal Components Analysis". *Technometrics*, **21**, 341–349.

Kaspar, M.H. (1992). *"Model Identification for Chemical Process Control"*. Ph.D. thesis. University of Wisconsin - Madison.

Kaspar, M.H., and Ray, W.H. (1992). "Chemometric Methods for Process Monitoring and High-Performance Controller Design". *A.I.Ch.E J.*, **38**(10), 1593–1608.

Kaspar, M.H., and Ray, W.H. (1993). "Dynamic PLS Modelling for Process Control". *Chemical Engng. Sci.*, **48**(20), 3447–3461.

King, R. (1986). "Early Detection of Hazardous States in Chemical Reactors ". *In: Proceedings of the IFAC Conference on Dynamics and Control of Chemical Reactors and Distillation Columns.*

Klein, R.E. (1990). "Teaching Linear Systems Theory Using Cramer's Rule". *IEEE Transactions on Education*, **33**(3), 258–267.

Kortmann, M., and Unbehauen, H. (1987). "Identification of Nonlinear MISO Systems". *Pages 225–230 of: Proceedings of the 10th IFAC World Congress.*

Kravaris, C., and Arkun, Y. (1991). "Geometric Nonlinear Control : An Overview". *In:* Arkun, Y., and Ray, W.H. (eds), *"Chemical Process Control - CPCIV"*. AIChE. NY. USA.

Kresta, J. (1992). *"Applications of Partial Least Squares Regression"*. Ph.D. thesis, McMaster University.

Kresta, J.V., MacGregor, J.F., and Marlin, T.E. (1991). "Multivariate Statistical Monitoring of Process Operating Performance". *Can. J. Chem. Eng.*, **69**(1), 35–47.

Kreyzig, E. (1979). *"Advanced Engineering Mathematics"*. 4 edn. John Wiley and Sons. NY.

Lakshminarayanan, S., Shah, S.L., and Nandakumar, K. (1995). "Identification of Hammerstein Models using Multivariate Statistical Tools". *Chemical Engng. Sci.*, **50**(22), 3599–3613.

Lang Zi-Qiang. (1994). "On Identification of the Controlled Plants Described by the Hammerstein System". *IEEE Trans. on autom. Control*, **39**(3), 569–573.

Larimore, W.E. (1990). "Canonical Variate Analysis in Identification, Filtering and Adaptive Control". *In: Proceedings of the 29th IEEE Conference on Decision and Control.*

Larimore, W.E. (1994). The Optimality of Canonical Variate Identification by Example. *Pages 151–156 of: Proceedings of the 10th IFAC Symposium on System Identification*, vol. 2.

Larimore, W.E., Mahmood, S., and Mehra, R.K. (1984). "Multivariable Adaptive Model Algorithmic Control". *Pages 675–680 of: Proceedings of the Conference on Decision and Control*, vol. **2**.

Lee, P.L., and Sullivan, G.R. (1988). "Generic Model Control (GMC)". *Comput. Chem. Engng*, **12**, 573–580.

Lindgren, F., Geladi, P., Rannar, S., and Wold, S. (1994). "Interactive Variable Selection for PLS : Part I". *Journal of Chemometrics*, **8**, 349–363.

Ljung, L. (1987). *"System Identification : Theory for the user"*. Prentice-Hall, NJ.

Ljung, L., and Soderstrom, T. (1983). *"Theory and Practice of Recursive Identification"*. MIT Press, Cambridge (MA), USA.

Luyben, W.L., and Eskinat, E. (1994). "Nonlinear Auto-tune Identification". *Int. J. Control*, **59**(3), 595–626.

MacGregor, J.F., Jaeckle, C., Kiparissides, C., and Koutoudi, M. (1986). "State Estimation for Polymerization". *In: Proceedings of the IFAC Conference on Dynamics and Control of Chemical Reactors and Distillation Columns.*

MacGregor, J.F., Nomikos, P., and Kourti, T. (1994a). Multivariate Statistical Process Control of Batch Processes using PCA and PLS. *Pages 525–530 of: Proceedings of the IFAC ADCHEM'94 Meeting.*

MacGregor, J.F., Jaeckle, C., Kiparissides, C., and Koutoudi, M. (1994b). "Process Monitoring and Diagnosis by Multiblock PLS Methods". *A.I.Ch.E J.*, **40**(5), 826–838.

Manne, R. (1987). "Analysis of two partial-least-squares algorithms for multivariate calibration". *Chemometrics and Intelligent Laboratory Systems*, **2**, 283–290.

Marlin, T.E. (1995). *"Process Control : Designing Processes and Control Systems for Dynamic Performance"*. McGraw-Hill, Inc., NY.

Martens, H., and Naes, T. (1989). *"Multivariate Calibration"*. John Wiley and Sons, NY.

"MATLAB©". (1992). The Mathworks, Inc., MA (USA).

Mejdell, T., and Skogestad, S. (1991). " Estimation of Distillate Composition from Multiple Temperature Measurements using Partial Least Squares Regression ". *Ind. Engng. Chem. Res.*, **30**, 2543–2555.

Miller, P., Swanson, R.E., and Heckler, C.F. (1993). "Contribution Plots : The Missing Link in Multivariate Quality Control". *In: 37th Annual Fall Conference, ASQC. Rochester. NY.*

Morari, M. (1990). "Process Control Theory : Reflections on the Past and Goals for the Next Decade". *In:* Prett D.M., García, C.E., and Ramaker, B.L. (eds), *"The Second Shell Process Control Workshop : Solutions to the Shell Standard Control Problem"*. Butterworth Publishers, MA, USA.

Morari, M., and Zafiriou, E. (1989). *"Robust Process Control"*. Prentice-Hall, Englewood Cliffs, NJ.

Moteki, Y., and Arai, Y. (1986). "Operation Planning and Quality Design of a Polymer Process". *In: Proceedings of the IFAC Conference on Dynamics and Control of Chemical Reactors and Distillation Columns.*

Narendra, K.S., and Gallman, P.G. (1966). "An Iterative Method for the Identification of Nonlinear Systems using the Hammerstein Model". *IEEE Trans. autom. Control.* **12**, 546–550.

Neergaard, L.J. (1993). "Use of Cramer's Rule to Simplify Batch Calculation". *Journal of Materials Education*, **15(3)**, 167–178.

Niu, S. (1994). *"The Augmented UD Identification for Process Control"*. Ph.D. thesis. University of Alberta.

Niu, S., and Fisher, D.G. (1994). "Simultaneous Structure Identification and Parameter Estimation of Multivariable Systems". *Int. J. Control*, **59**(5), 1127–1142.

Nomikos, P., and MacGregor, J.F. (1994a). "Multi-way Partial Least Squares in Monitoring Batch Processes". *In: INCINC '94.*

Nomikos, P., and MacGregor, J.F. (1994b). "Monitoring of Batch Processes using Multi-way Principal Components Analysis". *A.I.Ch.E J.*, **40**(8), 1361–1375.

Nomikos, P., and MacGregor, J.F. (1995). "Multivariate SPC Charts for Monitoring Batch Processes". *Technometrics*, **37**(1), 41–59.

Norquay, S.J., Palazoglu, A., and Romagnoli, J.A. (1996). *"Unconstrained and Constrained Model Predictive Control using Wiener Model s : An Application to pH Neutralization"*. Submitted to the IFAC ADCHEM '97 Meeting.

Ogunnaike, B.A., and Ray, W.H. (1994). *"Process Dynamics, Modeling, and Control"*. Oxford University Press, Inc. NY.

Orr, J.W. (1989). "A Geometric Approach to Cramer's Rule". *Mathematics Magazine*, **62(1)**, 35–37.

Parwardhan, R.S. (1996). *"Constrained Model Predictive Control using Hammerstein and Wiener Models"*. Tech. rept. 96RSP1. University of Alberta.

Phatak, A. (1993). *"Evaluation of Some Multivariate Methods and their Applications in Chemical Engineering"*. Ph.D. thesis, University of Waterloo.

Phatak, A., Reilly, P.M., and Penlidis, A. (1993). "An Approach to Interval Estimation in Partial Least Squares Regression : A tutorial". *Analytica Chimica Acta*. **277**. 495–501.

Qin, S.J. (1993). "Partial Least Squares Regression for Recursive System Identification". *In: Proceedings of the 32nd Conference on Decision and Control*.

Qin, S.J., and McAvoy, T.J. (1992a). "A Data-Based Process Modeling Approach and its Applications". *In: Proceedings of the IFAC Conference on Dynamics and Control of Chemical Reactors (DYCORD+'92)*.

Qin, S.J., and McAvoy, T.J. (1992b). "Nonlinear PLS Modelling using Neural Networks". *Comput. Chem. Engng*, **16**(4), 379–391.

Richalet, R., and Mehra, R.K. (1982). "Model Algorithmic Control (MAC) : Basic Theoretical Properties". *Automatica*, **18**, 401–414.

Ricker, N.L. (1988). "The use of biased least-squares estimators for parameters in discrete-time pulse response models". *Industrial and Engineering Chemistry Research*. **27**. 343–350.

Robinson, S.M. (1970). "A short proof of Cramer's Rule". *Mathematics Magazine*. **43**. 94–95.

Rouhani, R., Rault, A., Testud, J.L., and Papon, J. (1978). "Model Predictive Heuristic Control : Application to Industrial Processes". *Automatica*, **14**, 413–428.

Sanchez, E., and Kowlaski, B.R. (1990). "Tensorial Resolution : A Direct Trilinear Decomposition". *Journal of Chemometrics*, **4**, 29–45.

Schaper, C.D., Larimore, W.E., Seborg, D.E., and Mellichamp, D.A. (1994). "Identification of Chemical Processes using Canonical Variate Analysis". *Comput. chem. Engng.* **18**(1), 55–69.

Searle, S.R. (1982). *"Matrix Algebra Useful for Statistics"*. John Wiley and Sons. NY.

Seborg, D.E., Edgar, T.F., and Mellichamp, D.A. (1989). *"Process Dynamics and Control"*. John Wiley and Sons, NY.

Sharma, Subhash. (1996). *"Applied Multivariate Techniques"*. John Wiley and Sons. NY.

Shen, S.H., and Yu, C.C. (1992). "Indirect Feedforward Control : Multivariable Systems". *Chemical Engng. Sci.*, **47**(12), 3085–3097.

Sripada, N.R., and Fisher, D.G. (1985) (June). "Multivariable Optimal Constrained Control Algorithm. Part I : Formulation and Application". *In: Proceedings of the International Conference on Industrial Process Monitoring and Control*.

Stanley, G., Marino-Galarraga, M., and McAvoy, T.J. (1985). "Shortcut Operability Analysis 1. The Relative Disturbance Gain". *Ind. Engng. Chem. Process Des. Dev.*, **24**. 1188.

Steiglitz, K.. and McBride, L.E. (1965). "A Technique for the Identification of Linear Systems". *IEEE Trans. autom. Control*, **AC-10**, 461–464.

Stone, M., and Brooks, R.J. (1990). "Continuum Regression : Cross-validated Sequentially Constructed Prediction Embracing Ordinary Least Squares, Partial Least Squares and Principal Components Regression". *Journal of the Royal Statistical Society*, **52**(2). 237–269.

"System Identification Toolbox for use with MATLAB". (1992). The Mathworks, Inc.. MA (USA).

Van Overschee, P., and De Moor, B. (1994). " N4SID : Subspace algorithms for the identification of combined deterministic-stochastic systems ". *Automatica*, **30**, 75–93.

Van Overschee, P., and De Moor, B. (1995). "An unifying theorem for Three Subspace Identification Algorithms". *Automatica*, **31**(12), 1853–1864.

Venkatsubramanian, V.. and Chan, K. (1989). "A Neural Network Methodology for Process Fault Diagnosis". *A.I.Ch.E J.*. **35**, 1993–2002.

Verghese, G.C. (1982). "A "Cramer rule" for Least-norm Least-square-error Solution of Inconsistent Linear Equations". *Linear Algebra and its Applications*, **48**, 315.

Verhaegen, M. (1994). " Identification of the deterministic part of MIMO state space models given in innovations form from input-output data ". *Automatica*, **30**, 61–74.

Verhaegen, M.. and Westwick, D. (1996a). " Identifying MIMO Hammerstein Systems in the context of Subspace Model Identification Methods ". *In: Proceedings of the IFAC '96 Triennial World Congress, San Francisco.*

Verhaegen, M.. and Westwick, D. (1996b). *"Identifying MIMO Hammerstein Systems in the context of a Subspace Model Identification Strategy "*. To appear in the Int. Journal of Control.

Vogel, E.F., and Edgar, T.F. (1980). "A New Dead Time Compensator for Digital Control". *In: ISA/80 Proceedings.*

W. H. Ray Research Group. (1989). *"CONSYD : Computer-Aided Control System Design"*. Prof. W. Harmon Ray, Department of Chemical Engineering, University of Wisconsin. Madison, WI 53706.

Wang, G. (1986). "A Cramer Rule for Minimum-norm (T) Least-squares (S) Solution of Inconsistent Linear Equations". *Linear Algebra and its Applications*. **74**, 213–218.

Wang, Y., Seborg, D.E., and Larimore, W.E. (1996). *"Process Monitoring using Canonical Correlation Analysis and Principal Components Analysis"*. Submitted to the IFAC ADCHEM '97 Meeting.

Wangen, L.E., and Kowlaski, B.R. (1988). "A Multiblock PLS Algorithm for Investigating Complex Chemical Systems". *Journal of Chemometrics*. **3**, 3–20.

Weber. R.. and Brosilow, C. (1972). "The Use of Secondary Measurements to Improve Control". *A.I.Ch.E. J.*. **18**. 614.

Wise, B.M. (1991). *"Adapting Multivariate Analysis for Monitoring and Modeling Dynamic Systems"*. Ph.D. thesis, University of Washington.

Wold, H. (1966). "Estimation of principal components and related models by iterative least squares". *Pages 391–420 of:* Krishnaiah, P.R. (ed), *Multivariate Analysis*. Academic Press, NY.

Wold, S. (1978). "Cross-Validatory Estimation of the Number of Components in Factor and Pricipal Components Models". *Technometrics*, **20**, 397–405.

Wold, S., Kettaneh-Wold, N., and Skagerberg, B. (1989). "Nonlinear PLS modelling". *Chemometrics and Intelligent Laboratory Systems*, **7**, 53–65.

Wood, R.K., and Berry, M.W. (1973). "Terminal Composition Control of a Binary Distillation Column". *Chemical Engng. Sci.*, **29**, 1808.

Zhu, X., and Seborg, D.E. (1994). "Nonlinear Predictive Control Based on Hammerstein Models". *Pages 995–1000 of: Proceedings of PSE '94.*

Appendix A

Formulation of PCA, PLS and CCA as Eigenvalue-Eigenvector Problems

The multivariate statistical techniques that are employed in this thesis can also be cast in a eigenproblem framework. Here, we present a quick overview of the same. Assume that the data blocks X and Y have been autoscaled.

- Principal Components Analysis

  The goal in PCA is to maximize the amount of variance explained in each of the principal components (latent variables). Denoting the first latent variable by $T_1 = X j_1$, we may express the variance measure of this latent variable as :

$$T_1' T_1 = j_1' X' X j_1 \tag{A.1}$$

  where ' indicates the transpose operator. Some restriction or normalization is required on j_1 in order to pose the problem correctly. The following condition is used in the PCA technique.

$$j_1' j_1 = 1 \tag{A.2}$$

  Using a Lagrangian multiplier λ, the corresponding objective function to be maximized is :

$$\Theta = j_1' X' X j_1 - \lambda (j_1' j_1 - 1) \tag{A.3}$$

Taking partial derivatives.

$$\frac{\partial \Theta}{\partial j_1'} = 2X'Xj_1 - 2\lambda j_1 \tag{A.4}$$

Setting the derivative equal to zero, we get

$$\frac{\partial \Theta}{\partial j_1'} = 0 \Rightarrow 2X'Xj_1 = 2\lambda j_1 \tag{A.5}$$

Using equation (A.5), we may write

$$X'Xj_1 = \lambda j_1 \tag{A.6}$$

Equation (A.6) is the standard form in which the eigenvalue-eigenvector problem is usually expressed. j_1 is the eigenvalue of the matrix $X'X$ corresponding to the largest eigenvalue λ (notice that we are interested in maximizing the variance and therefore λ must correspond to the largest eigenvalue). The first principal component (latent variable) is $T_1 = Xj_1$ and $var(T_1) = var(Xj_1) = \lambda$.

The k^{th} principal component may be expressed as $T_k = Xj_k$, where j_k is the eigenvector corresponding to the k^{th} largest eigenvalue of $X'X$.

- Partial Least Squares

  In partial least squares, the goal is to maximize the covariance between the X and Y block latent variables. Let T_1 and U_1 denote the latent variables of the X and Y blocks respectively. The covariance between these latent variables can be expressed as :

$$Covariance = T_1'U_1 = j_1'X'Yl_1 \tag{A.7}$$

The covariance given above is to be maximized subject to the constraints which are as follows :

$$j_1'j_1 = 1 \text{ and } l_1'l_1 = 1 \tag{A.8}$$

After introducing the lagrangian multipliers λ_1 and λ_2, we may write the objective function in the following form

$$\Theta = j_1'X'Yl_1 - \lambda_1(j_1'j_1 - 1) - \lambda_2(l_1'l_1 - 1) \tag{A.9}$$

Differentiating equation (A.9) partially with respect to j_1' and l_1' respectively, we get

$$\frac{\partial \Theta}{\partial j_1'} = X'Yl_1 - 2\lambda_1 j_1 \tag{A.10}$$

$$\frac{\partial \Theta}{\partial l_1'} = Y'Xj_1 - 2\lambda_2 l_1 \tag{A.11}$$

On equating the partial derivatives given by equations (A.10) and (A.11) to zero. we get

$$X'Yl_1 = 2\lambda_1 j_1 \tag{A.12}$$

$$Y'Xj_1 = 2\lambda_2 l_1 \tag{A.13}$$

Employing equations (A.12) and (A.13) and after some simple algebraic manipulations, we arrive at equations (A.14) and (A.15).

$$X'YY'Xj_1 = 4\lambda_1\lambda_2 j_1 \tag{A.14}$$

$$Y'XX'Yl_1 = 4\lambda_1\lambda_2 l_1 \tag{A.15}$$

Equations (A.14) and (A.15) represent the eigenproblem corresponding to the PLS technique. Using (A.9), (A.12) and (A.13) and the constraints (i.e.. equation (A.8)). it is possible to show that λ_1 equals λ_2; further, the product $4\lambda_1\lambda_2$ corresponds to the largest eigenvalues of $X'YY'X$ and $Y'XX'Y$ respectively with j_1 and l_1 being the corresponding eigenvectors.

- Canonical Correlations Analysis

  As the name suggests, the Canonical Correlations Analysis technique is concerned with maximizing the correlation between the latent variables of the X and Y blocks. Let us denote these latent variables (called *canonical variates* in CCA related literature) by T_1 and U_1 respectively.

  The correlation measure that is sought to be maximized is given by equation (A.16).

$$Correlation = \frac{T_1'U_1}{\sqrt{T_1'T_1}\sqrt{U_1'U_1}} \tag{A.16}$$

  In terms of the original X and Y blocks. the above equation may be expressed as :

$$Correlation = \frac{j_1' X' Y l_1}{\sqrt{j_1' X' X j_1}\sqrt{l_1' Y' Y l_1}} \tag{A.17}$$

In CCA, the constraints on the optimization problem are specified as follows :

$$j_1' X' X j_1 = 1 \text{ and } l_1' Y' Y l_1 = 1 \tag{A.18}$$

With the introduction of the lagrangian multipliers (λ_1 and λ_2) and utilizing the constraints posed above, we may write the CCA maximization function in the following manner :

$$\Theta = j_1' X' Y l_1 - \lambda_1 (j_1' X' X j_1 - 1) - \lambda_2 (l_1' Y' Y l_1 - 1) \tag{A.19}$$

Performing the usual task of taking the partial derivatives of the above equation with respect to j_1' and l_1' respectively and setting them equal to zero, we get

$$X' Y l_1 = 2\lambda_1 X' X j_1 \tag{A.20}$$

$$Y' X j_1 = 2\lambda_2 Y' Y l_1 \tag{A.21}$$

Using equation (A.21), it is possible to express l_1 as

$$l_1 = \frac{(Y'Y)^{-1} Y' X j_1}{2\lambda_2} \tag{A.22}$$

Substituting equation (A.22) into equation (A.20) and performing some algebraic manipulations results in

$$(X'X)^{-1} X' Y (Y'Y)^{-1} Y' X j_1 = 4\lambda_1 \lambda_2 j_1 \tag{A.23}$$

Similarly, it is possible to show that

$$(Y'Y)^{-1} Y' X (X'X)^{-1} X' Y l_1 = 4\lambda_1 \lambda_2 l_1 \tag{A.24}$$

It can be proved that λ_1 equals λ_2 and that the product $4\lambda_1\lambda_2$ represents the largest eigenvalue of the matrices $(X'X)^{-1}X'Y(Y'Y)^{-1}Y'X$ and $(Y'Y)^{-1}Y'X(X'X)^{-1}X'Y$ respectively. Further, the eigenproblems represented by equations (A.23) and (A.24) suggest that j_1 and l_1 are the eigenvectors corresponding to the largest eigenvalues of $(X'X)^{-1}X'Y(Y'Y)^{-1}Y'X$ and $(Y'Y)^{-1}Y'X(X'X)^{-1}X'Y$ respectively.

Although the solutions of the multivariate techniques covered here are provided in terms of eigenproblems, it is relatively straightforward to cast them in terms of singular value decomposition (SVD) of appropriate matrices (see Chapter 1). Links between the eigenproblem and the SVD technique is available in several linear algebra textbooks.

Appendix B

Cramer's Rule for Nonsquare Systems

The solution of a linear system of n equations in n variables can be expressed as quotients of determinants. Gabriel Cramer (1704-52) published, in 1750, this celebrated rule which he had obtained by the "*science of algebra.*" Though computationally efficient algorithms such as the LU decomposition are preferred for large numerical problems. Cramer's rule is of interest if the equations contain parameters and if analytical expressions are required (Kreyzig, 1979). Cramer's rule continues to be used in a variety of fields as evidenced by some recent publications (Klein (1990), Neergaard (1993)) and serves as a valuable tool to illustrate basic concepts of linear algebra.

Consider the following n by n system :

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n \end{aligned}$$

In matrix notation, we may write A x = b where

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & \cdots & a_{nn} \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} \text{ and } b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}$$

Let A_i denote the i^{th} column of A (i.e., $A = [A_1 \mid A_2 \mid \cdots \mid A_{i-1} \mid A_i \mid A_{i+1} \mid \cdots \mid A_n]$). If A is a matrix of full rank (has *linearly independent rows and columns*), then Cramer's rule gives the solution of x_i (i = 1, ..., n) as :

$$x_i = \frac{\det(A_i^*)}{\det(A)} \tag{B.1}$$

where $A_i^* = [A_1 \mid A_2 \mid \cdots \mid A_{i-1} \mid b \mid A_{i+1} \mid \cdots \mid A_n]$ is obtained by replacing the i^{th} column of A with the vector b. A simple proof of Cramer's rule can be found in Robinson (1970). Orr (1989) provides a geometric interpretation.

Such a rule is easily extended to the case where it is desired to find the solution to the equation $A_{n\times n}X_{n\times r} = B_{n\times r}$. Now, X and B are matrices rather than vectors. Element [i,j] (i = 1, .. , n ; j = 1, .. , r) of the matrix X can be written by generalizing equation (B.1) as :

$$X_{ij} = \frac{\det(A_{ij}^*)}{\det(A)} \tag{B.2}$$

Here, the matrix A_{ij}^* is obtained by replacing the i^{th} column in matrix A by the j^{th} column of B (i.e. $A_{ij}^* = [A_1 \mid A_2 \mid \cdots \mid A_{i-1} \mid B_j \mid A_{i+1} \mid \cdots \mid A_n]$).

In practical situations we often need to deal with nonsquare system of equations. For example, the linear model A X = B is developed from experimental data - the number of experimental observations is seldom exactly equal to the number of variables. Usually, there are more observations than variables though the reverse is also possible. Under such circumstances, the least squares solution is preferred.

Consider the nonsquare system represented as $A_{m\times n}X_{n\times r} = B_{m\times r}$ where A is assumed to be of full (column or row) rank. The least squares solution of this system is represented as $X = A^{\dagger}B$ where $A^{\dagger}$ is the *pseudoinverse* of matrix A. Depending on whether matrix A is narrow (m > n ; more observations than variables) or broad (m < n : less observations than variables), $A^{\dagger}$ can take two forms :

- Case (a) : A is narrow (m > n)

$$X = A^{\dagger}B = (A^TA)^{-1}A^TB \tag{B.3}$$

$(A^TA)^{-1}A^T$ is the *left inverse* of matrix A. The symbol T indicates the transpose operator. We now provide an extended Cramer's Rule to determine an analytical expression for a particular element X_{ij} (i = 1, .. , n ; j = 1, .. , r) of matrix X.

$$X_{ij} = \frac{\det(A^TA_{ij}^*)}{\det(A^TA)} \tag{B.4}$$

Illustration 1

$$\text{Let } A = \begin{bmatrix} 5 & 3 \\ 7 & 6 \\ 2 & 1 \end{bmatrix}, B = \begin{bmatrix} 2 & 4 & 5 \\ 3 & 7 & 1 \\ 1 & 2 & 6 \end{bmatrix}.$$

Here, m = 3, n = 2 and r = 3. To find the element X_{22}, we first compute $A^{=}_{22}$ as

$$A^{=}_{22} = \begin{bmatrix} 5 & 4 \\ 7 & 7 \\ 2 & 2 \end{bmatrix}$$

The matrix products and the determinant values are obtained as :

$$A^T A^{=}_{22} = \begin{bmatrix} 78 & 73 \\ 59 & 56 \end{bmatrix} \text{ and } \det(A^T A^{=}_{22}) = 61$$

$$A^T A = \begin{bmatrix} 78 & 59 \\ 59 & 46 \end{bmatrix} \text{ and } \det(A^T A) = 107 \text{ (shows that A is of full rank).}$$

From equation (B.4), we obtain $X_{22} = \frac{61}{107} = 0.5701$. The reader can verify that this is indeed the least squares solution of X_{34} using any computational tool such as MATLAB©.

- Case (b) : A is broad (m < n)

$$X = A^{\dagger} B = A^T (A A^T)^{-1} B \tag{B.5}$$

$A^T(AA^T)^{-1}$ is the *right inverse* of matrix A. The *extended* Cramer's Rule to determine a particular element X_{ij} of matrix X is :

$$X_{ij} = \frac{[\det(A^{=}_{ij} A^T) - \det(A^0_i A^{0^T}_i)]}{\det(AA^T)} \tag{B.6}$$

where $A^0_i = [A_1 \mid A_2 \mid \cdots \mid A_{i-1} \mid A_{i+1} \mid \cdots \mid A_n]$ is obtained by simply deleting the i^{th} column in matrix A.

Illustration 2

$$\text{Let } A = \begin{bmatrix} 5 & 7 & 2 \\ 3 & 6 & 1 \end{bmatrix}, B = \begin{bmatrix} 6 & 2 & 1 & 7 \\ 7 & 6 & 4 & 2 \end{bmatrix}.$$

Here m = 2, n = 3 and r = 4. To find the element X_{34}, we first compute $A^{=}_{34}$ and A^0_3. Using the definitions we get,

$$A^{=}_{34} = \begin{bmatrix} 5 & 7 & 7 \\ 3 & 6 & 2 \end{bmatrix} \text{ and } A^0_3 = \begin{bmatrix} 5 & 7 \\ 3 & 6 \end{bmatrix}$$

The matrix products and the determinant values are obtained as :

$$A^{=}_{34} A^T = \begin{bmatrix} 88 & 64 \\ 61 & 47 \end{bmatrix} \text{ and } \det(A^{=}_{34} A^T) = 232$$

$$A_3^0 A_3^{0^T} = \begin{bmatrix} 34 & 53 \\ 53 & 85 \end{bmatrix} \text{ and } \det(A_3^0 A_3^{0^T}) = 81$$

$$AA^T = \begin{bmatrix} 78 & 59 \\ 59 & 46 \end{bmatrix} \text{ and } \det(AA^T) = 107 \text{ (implies A is nonsingular).}$$

Using equation (B.6), we obtain $X_{34} = \frac{(232-81)}{107} = 1.4112$. This is indeed the minimum norm solution for this element.

Concluding Remarks

It is easy to see that if B is chosen as an identity matrix of appropriate size, the pseudoinverse of A can be computed, element by element, using equations (B.2), (B.4) and (B.6) as long as A is of full rank. Ben-Israel (1982), Verghese (1982) and Wang (1986) provide Cramer's rule for the linear system $Ax = b$ when A is rank deficient. However, their expressions do not reduce explicitly to those given here for a nonsingular A.

In the process of developing compact representations for a multivariable feedforward controller, we noticed certain *patterns* in the solution. These patterns helped us to extend Cramer's rule to the domain of nonsquare-nonsingular systems (equations (B.4) and (B.6)). While the proof of equation (B.4) is straightforward (omitted here for the sake of brevity), proving equation (B.6) may be a bit formidable and has been proposed as a challenge problem in *The American Mathematical Monthly* .