

# NOTE TO USERS

This reproduction is the best copy available.

**UMI**<sup>®</sup>



BATCH PROCESS IMPROVEMENT  
USING LATENT VARIABLE METHODS

By

SALVADOR GARCÍA MUÑOZ, M. Sc.

A Thesis

Submitted to the School of Graduate Studies

In Partial Fulfillment of the Requirements

For the Degree

Doctor of Philosophy

McMaster University

© Copyright by Salvador García Muñoz, September. 2004

## BATCH PROCESS IMPROVEMENT USING LATENT VARIABLE METHODS

Doctor of Philosophy (2004)  
(Chemical Engineering)

McMaster University  
Hamilton, Ontario

TITLE: Batch Process Improvement using Latent Variable Methods

AUTHOR: Salvador García Muñoz, B.Sc. M.Sc. (Instituto Tecnológico y de Estudios Superiores de Monterrey, ITESM – Campus Monterrey, Monterrey, NL. México)

SUPERVISORS: Dr. John F. MacGregor.

Dr. Theodora Kourti

NUMBER OF PAGES: *xiv*, 227

## Abstract

This thesis deals with the following four topics:

1. Multivariate statistical methods are used to analyze data from an industrial batch drying process. Principal Component Analysis (PCA) and Partial least-squares (PLS) methods were able to isolate which group of variables from the initial conditions and the process variables were related to a poor-quality product. The use of a novel approach to the time warping of the trajectories for batches, and the subsequent use of the time-warping information, is presented.

2. In the procedure to monitor a new batch using the method proposed by Nomikos and MacGregor (1994), an assumption about the unknown future samples in the batch has to be taken. This work demonstrates that using the *missing data* (MD) option and estimating the score with an appropriate method are equivalent to the use of an adaptive-expansive multivariate time series model in the forecasting for the unknown future samples. The benefits of using the MD option are analyzed on the basis of (i) the accuracy of the forecast, (ii) the quality of the score estimates, and (iii) the detection performance during monitoring.

3. Jaeckle and MacGregor (1998) introduced a technique to estimate operating conditions in order for a process to yield a product with a desired set of characteristics. This thesis provides a detailed study of the application of such technique in designing the operation of a batch process. The original technique is modified to include constraints and other optimal criteria onto the desired quality and the trajectories. A parallel approach based on derivative-augmented models is proposed to avoid the analysis of the null space.

4. An extension to the work by Jaeckle and MacGregor (2000) in solving the product transfer problem is proposed. The early technique does not consider all the data structures involved in the problem and particularly the operating conditions from the *source* plant. The Joint-Y PLS model is presented as an alternative to solve this problem using all the available data.

## Acknowledgements

“In the beginning was the Word, and the Word was with God, and the Word was God. He was in the beginning with God. All things came to be through him, and without him nothing came to be. What came to be through him was life, and this life was the light of the human race; the light shines in the darkness, and the darkness has not overcome it.”

The Bible - *John, 1, 1*

Thank you Lord for sending the light of Your Holy Spirit that kept me going in this journey.

Thank you Betty, Andrea and Isabel for being the light that always shined in the darkest of the moments. You are my reason to live and be a better man.

Thank you Dr. MacGregor and Dr. Kourti, Dr. Viveros and Dr. Quinn for your patience and your teachings which went beyond academics.

Thank you Mom and Dad (Mr. Salvador García López and Mrs. Maria Cristina Muñoz de García) for your support in this wild project.

Thank you to the rest of the Chemical Engineering faculty and my peers at the advanced control group at McMaster University for all the valuable discussions we had. Special thanks also to Marc Champagne, Francesca Apruzzese and Joe Ranking from Tembec; Debashis Neogi from Air Products and Chemicals; and Arthur Matheos formerly with FMC Chemicals (now with Merck) for the data provided to this research project and for the time you gave me to transmit your experiences and knowledge.

# Table of Contents

Page

1	<b>Chapter 1: Introduction</b>
6	<b>Chapter 2: Analysis of Batch Processes using Multivariate Methods and Alignment Techniques</b>
6	2.1 Introduction
8	2.2 The drying process
10	2.3 The data set
11	2.4 Multivariate classification of quality
14	2.5 Alignment of the batch data
14	2.5.1 Alignment technique
21	2.5.2 Inclusion of “time usage” as one extra trajectory per batch
22	2.6 Analysis of historical batch data
23	2.6.1 Impact of initial conditions and warping information ( $Z$ ) on final quality ( $Y$ )
27	2.6.2 Impact of process variable trajectories on the final quality
32	2.7 The Anomalous Batches 420, 411, 412 and 422
34	2.8 Conclusions
36	<b>Chapter 3: Model Predictive Monitoring for Batch Processes</b>
36	3.1 Introduction
41	3.2 Forecast and filling in mechanisms
41	3.2.1 Notation

42	3.2.2 Data Sets
43	3.2.3 Future trajectory predictions and assumptions for batch monitoring
51	3.2.4 The PCA forecast model interpreted as a time series forecast
54	3.3 Statistics for Comparing Future Predictions
63	3.4 Quality of the score estimates
69	3.5 Monitoring Performance
75	3.6 Conclusions
77	<b>Chapter 4: Product Design for Batch Processes using Latent Variable Model Inversion via Constraint Optimization</b>
77	4.1 Introduction
78	4.1.1 Background
81	4.1.2 Design of operating conditions by LVR model inversion
83	4.1.3 Extensions to the model inversion approach
84	4.1.4 Contributions of this work
85	4.2 First Industrial Example Description
89	4.3 Batch Trajectory Estimation using latent variable models via optimization with multivariate constraints
90	4.3.1 Solving for $\hat{\tau}_{\text{new}}$ that will yield $\mathbf{y}_{\text{des}}$
92	4.3.2 Re-constructing the operating conditions $\hat{\mathbf{x}}_{\text{new}}$ from $\hat{\tau}_{\text{new}}$
98	4.3.3 Application to an Industrial Pulp Digester
112	4.4 Model Augmentation with Derivatives
114	4.4.1 Building the augmented multi-way PLS Model
115	4.4.2 Re-Formulating the Objective Function
117	4.4.3 Application to industrial Pulp Digester
118	4.5 Optimal Trajectory Design - Minimize Batch Length

120	4.6 Industrial Example: Optimal Trajectory Estimation for an Emulsion Polymerization Process
128	4.7 Batch Modeling and Troubleshooting using Augmented Models with Partial Derivatives
131	4.8 Conclusions and Future Work
132	<b>Chapter 5: Product Transfer using Joint-Y PLS</b>
132	5.1 Introduction
134	5.1.1 Product Transfer using Model Inversion
138	5.1.2 Motivation for a technique which uses all data structures
140	5.2 The Joint-Y PLS (JYPLS) Model
143	5.2.1 Parameter Estimation Approaches
149	5.3 JYPLS Model Diagnostics
152	5.4 Modifications to the JYPLS Model
152	5.4.1 Multiple Sites
153	5.4.2 Multi-Block JYPLS Models (MBJYPLS)
155	5.4.3 Non-Linear JYPLS
155	5.5 Product Transfer using JYPLS
156	5.5.1 Transfer Technique
157	5.5.2 Null Space in JYPLS
158	5.6 Product Transfer using EPCR vs. JYPLS
165	5.7 Examples
166	5.7.1 Simulation of an LLDPE Process
173	5.7.2 Scale-up of an Industrial Pulp Digester from Pilot Plant Experiments
181	5.8 Assessing Parallel Plants using JYPLS
185	5.9 Conclusions

187	<b>Chapter 6: Conclusions</b>
193	Appendix 1
203	Appendix 2
205	Appendix 3
209	Appendix 4
211	Appendix 5
214	Appendix 6
216	Appendix 7
221	References

## List of Captions

Page	Figure Number	Title
7	2.1	Unfolding of the three way batch data set
7	2.2	Multi-block multi-way PLS matrices
9	2.3	Critical Variables and two critical steps in operation
12	2.4	SPE after 2 components in a PCA for Y variables
13	2.5	PCA score plot for product quality data (Y)
13	2.6	SPE for the Y matrix multivariate classification model
16	2.7	Stages and new variables defined during the batch
18	2.8	Variable 3 for all batches before (top) and after (bottom) alignment
19	2.9	Collector tank level variable for all batches before (top) and after (bottom) alignment
20	2.10	Dryer-temperature variable for all batches before (top) and after (bottom) alignment
21	2.11	Distortion of clock time with the alignment of batch data
25	2.12	VIP Plot for Z -Y PLS model
26	2.13	Score plot for the Z – Y PLS model
26	2.14	Loadings plot for the $Z_w$ Variables in the Z – Y PLS model
26	2.15	Loadings plot for the $Z_{ch}$ variables in the Z – Y PLS model
27	2.16	Score plot for the X space in the X-Y MPLS
28	2.17	Percentage captured per variable in the X matrix
29	2.18	Weights for 1st component in the X space for the XY MPLS
31	2.19	Differences in the trajectory of the collector tank level between batches 72 and 2
33	2.20	Contribution plots to $t_1$ in the XY-MPLS model for bad batches 415 and 410 (top), and anomalous batches 420 and 412 (bottom)
34	2.21	Contribution plots to the scores $t_1$ in the ZY-PLS model from batch 415 to batch 420 (left) and from 410 to 412 (right)
38	3.1	Unfolding of the 3D batch data matrix
38	3.2	Unknown samples in monitoring a new batch
42	3.3	Notation and definitions
44	3.4	Future trajectory prediction when using raw current deviations (x) and mean centered and scaled current deviations (dotted line), raw value scale (top) and MCS scale (bottom)
46	3.5	TS Forecast for the unknown samples
50	3.6	PCA Forecast for the unknown samples using $PP_{PLS}$
54	3.7	Coefficient $\Phi_j^k(2,2)$ as it adapts and expands with time (k)

55	3.8	Future prediction error in batch trajectory forecasting
58	3.9	FPRESS and FPMSE for variable 3 in data set A, using zeros option
59	3.10	Mean FPMSE for variable 1 in data set A, using missing data and solving with SCP
60	3.11	Mean FPMSE (top) and FPRESS (bottom) for variable 4 in data set A
62	3.12	Mean FPRESS (top) and mean FPMSE (bottom) for variable 1 in dataset B
64	3.13	Monitoring score plots for different “filling in” options and missing data handling methods for dataset A
65	3.14	Variance and covariance of the scores from K models throughout the batch, dataset A
67	3.15	Time evolving variance-covariance of the scores for CMR, PPPLS, CD Zeros, TS and II for dataset A
68	3.16	Evolving variance of $t_1$ for dataset B, using all the methods and options
82	4.1	Kappa and Viscosity for a Product Family of Pulp
86	4.2	Temperature Profiles for Different Grades of Pulp
87	4.3	Aligned Temperature Profiles for Different Grades of Pulp
89	4.4	Super Score plot ( $t_1-t_2$ ) and square prediction error (SPE) for a MBPLS model with $Z_{PBD}$ $X_{PBD}$ and $Y_{PBD}$
99	4.5	Grades of pulp being produced by the pilot batch digester
100	4.6	Score Plots ( $\circ$ model, $\square$ $\tau_{new}$ , $+\tau_{xnew}$ ) and Design Estimates (dotted line) known solution (solid black) and model building grades (gray lines), for grade C (top), A (bottom left) and F (bottom right)
102	4.7	Null Space and Minimum Euclidian Distance (+) and Mahalanobis Distance ( $\blacklozenge$ ) Solution along the Null Space in the Design of Grade A in the Pilot Batch Digester
106	4.8	Univariate null spaces, pseudo-null space and several solutions for the design exercise in section 4.3.2
107	4.9	Temperature Trajectories for the several solutions on the null and pseudo-null space for the design exercise of grade A in the pilot batch digester
109	4.10	Combined SPE for solutions along the pseudo-null space in the design of grade A for the pilot digester, dashed line is 95% confidence limit from the model (symbols as in Fig 4.8)
110	4.11	Score Plots ( $\circ$ model, $\square$ $\tau_{new}$ , $+\tau_{xnew}$ ) and Design Estimates (dotted line), known solution (solid black) and model building grades (gray lines) for grades A (left) and F (right) in the pilot batch digester, imposing bounds on the solution
111	4.12	Null space oriented away from the origin
113	4.13	Data Structure and re-ordered X to be used in the augmented model
116	4.14	Segmentation of $H_a$
117	4.15	Score Plots ( $\circ$ model, $\square$ $\tau_{new}$ , $+\tau_{xnew}$ ) and Design Estimates (dotted line), known solution (solid black) and model building grades (gray lines) for grades A (left) and F (right) in the pilot batch digester.

121	4.16	Raw Process Trajectories for the Emulsion semi-batch process
122	4.17	Aligned Process Trajectories and Time Usage for the emulsion semi-batch process
124	4.18	Different Optimal Solutions for different values of $\eta$
125	4.19	Detail of the Different Optimal Solutions for different values of $\eta$
126	4.20	Unaligned Different Optimal Solutions for different values of $\eta$
129	4.21	R2Y and Q2Y as partial derivatives are included in the model
130	4.22	FPRESS for all variables in the emulsion polymerization data
135	5.1	Product transfer from plant A to plant B
139	5.2	Schematic use of PLS to directly estimate the conditions in plant B from those in plant A if enough common grades exist, with no use of the Y matrices
140	5.3	Loadings and Scores for two separate PLS models
141	5.4	Loadings and Scores for the Joint-Y PLS model
146	5.5	NIPALS modified to compute loadings the JYPLS model
152	5.6	JYPLS built with multiple sites
154	5.7	Multi-block JYPLS model data structures
162	5.8	Case $Y_a \rightarrow B$ : Known and estimated values for $X_{Bnew}$ for EPCR and JYPLS for three observations at random
163	5.9	Case $Y_a \rightarrow B(10 \text{ obs. in } B)$ : Known and estimated values for $X_{Bnew10}$ for EPCR and JYPLS for three observations at random
164	5.10	Case $Y_a \rightarrow C$ : Known and estimated values for $X_{Cnew}$ for EPCR and JYPLS for three observations at random
169	5.11	Operating conditions and quality produced for simulated LLDPE transfer case 1
170	5.12	New operating conditions $X_{Anew}$ and $X_{Bnew}$ (top), and quality after transfer ( $Y_{AinB}$ middle, $Y_{BinA}$ bottom) for LLDPE case 1
171	5.13	Operating conditions and quality for LLDPE case 2
172	5.14	Operating conditions (top) and quality obtained after transfer for LLDPE case 2
174	5.15	Grades of pulp produced at a full scale digester and a pilot plant
175	5.16	Temperature trajectories for the pilot plant
175	5.17	Temperature profiles for the full scale digester
177	5.18	Score spaces for the transfer of grade D for EPCR (top) and JYPLS (bottom)
178	5.19	Temperature trajectory for full scale digester when the score ( $\tau_{b_{new}}$ ) is moved along the null space
179	5.20	Temperature trajectories design estimate for JYPLS (dotted) and EPCR (dashed) for grades D (top), F (middle) and I (bottom)
182	5.21	Loadings plot for the temperature in a JYPLS model for the pulp digester example used in Section 5.7.2
184	5.22	Loadings plot for two simulated batch plants producing SBR latex

## List of Tables

Page	Table Number	Title
11	2.1	Variables measured per batch
24	2.2	Percentage variance of Y explained by various PLS models with Z
70	3.1	Standard deviation of the disturbances added to the simulation results
71	3.2	Percentage of samples outside of 95% confidence limits for the validation and training data sets
73	3.3	Comparison of the Mean Time To Detection (MTD) and the standard deviation of the MTD ( $\sigma(\text{MTD})$ ) of different methods
75	3.4	Two Way ANOVA Table for the Times to Detection
86	4.1	Grades of Pulp produced in the Pilot Digester
101	4.2	Percentage captured variance of X and Y in individual PCA models
127	4.3	Square prediction error for the five different optimal solutions
149	5.1	Values for $w_a$ and $w_b$ obtained with modified-NIPALS and with SVD for a JYPLS model fitted to the test set in Appendix 6
150	5.2	Diagnostics to be considered in the JYPLS model
159	5.3	Data structures used to compare EPCR with JYPLS
160	5.4	Notation to refer the estimates in the JYPLS-EPCR comparison exercise
161	5.5	Mean square prediction error for all the estimates in Table 5.4
167	5.6	Characteristics of the sites for the LLDPE example
180	5.7	Viscosity and kappa prediction and square prediction error (SPE) for the analytical solution and minimal distance to the origin along the null space solution.

## List of Schemes

<b>Page</b>	<b>Scheme Number</b>	<b>Title</b>
79	1	Classification of the different approaches to batch operation design
96	2	Steps to follow and computational options in the design of batch process operating conditions
97	3	Additions to the design technique done in sections 4.4 and 4.5

# Chapter 1

## Introduction

Batch processes play a key role in today's manufacturing of high value added chemicals like pharmaceuticals and polymer resins. Process systems engineering (PSE) has played a crucial role in developing techniques to improve the operation for batch units.

Such improvements can be done at several stages of the life-cycle of the batch process. At the design phase where the geometry and nominal operation is determined, fundamental models are traditionally used to aid the design exercise and estimate the best possible design for the intended use. Once the batch is in production, the improvements can be done in the scheduling (usually done with optimization techniques); or in the production itself (e.g. by including new control strategies into the system). Finally at a certain time into the life cycle of the unit, the operation of the batch might be reconsidered and purposefully changed perhaps due to changes in the desired specifications of the final product.

Once the batch is into the production stage of its life cycle, a certain amount of unwanted disturbances will enter the process, and the original design may not be suitable to protect the process against such disturbances (or the nature of disturbances changes from the ones assumed in the original design exercise). Improving a batch process in order for its operation to deal with these new (or modified) sources of uncertainty is a

difficult task due to the nature of these disturbances and the complex chain of consequences that they may bring to the operation.

This thesis contains new ideas and techniques to aid the improvement of a batch process in its production stage, when it will suffer the presence of such uncertainties. This research focuses particularly in *i)* troubleshooting the operation; *ii)* building of a real-time monitoring scheme for the batch run; and *iii)* the re-designing of the operational conditions for the batch.

The proposed techniques in this work use data based driven latent variable models, which have been shown to be an attractive alternative to fundamental deterministic approaches. The four main chapters of this thesis provide solutions to different needs from industry in the improvement of batch processes. Chapters 2 and 3 deal with the troubleshooting and monitoring of the operation of batch processes. Chapter 4 covers the optimal re-design of operating conditions. And chapter 5 presents a novel regression technique to be used in the re-design of process operation; when the desired new product has been already produced at another site. A brief summary of each chapter is given below.

Chapter 2 elaborates on the troubleshooting and analysis of batch process operation by presenting a practical review of the techniques introduced by Nomikos and MacGregor (1994;1995a). This is done by applying such techniques to a set of data from an industrial drying process and interpreting the model parameters in great detail. The troubleshooting exercise is performed to the conclusion of the controller in place being the principal cause of product being off specification. The main contribution of this chapter is in the application itself; how the batch trajectories were synchronized; and how the time variable was included into the model for the analysis.

The techniques for batch process monitoring proposed by Nomikos and MacGregor (1995b) are poorly understood by some researchers when it comes to the monitoring of a new batch. Monitoring a new run at sample  $k$  (from  $K$  total samples) requires an assumption on how the variable trajectories will evolve for the rest  $K-k$  samples for the current batch run. Nomikos and MacGregor (1995b) propose three options:

- a) Consider the batch will return to nominal operation (referred to as Zeros because the value for the  $K-k$  unknown samples is set to zero)
- b) Assume that the current deviations from the mean trajectories are kept throughout the rest of the  $K-k$  samples (referred to as the Current Deviation option)
- c) Use the missing data handling features of a PCA model (referred to as the Missing Data option)

Chapter 3 is a thorough study of the implications in using missing data when it comes to forecasting the  $K-k$  unknown samples in the batch, and how this impacts into the fault detection performance. This study is done along with a comparison among the three mentioned options. And for the missing data case, several missing data handling methods are also included in the comparison exercise. Doing so uncovers practical differences among these methods (even though some of them are theoretically equivalent). The comparison is done from three different perspectives:

- i)* Accuracy of the forecast for the remaining  $K-k$  samples in the batch.
- ii)* Quality of the score estimates as the batch evolves, and
- iii)* Fault detection performance

This chapter includes many new contributions to the area: It uncovers the implicit forecasting model embedded into missing-data-handling mechanism for a PCA model. Such forecasting model is derived by understanding the mechanisms behind the missing

data handling techniques. Then it is mathematically manipulated to the form of a multivariate time series; to understand its properties and compare them with a fixed traditional time series model. The comparison exercise is a novel contribution as well since it shows explicitly the accuracy of the forecasts done by the PCA model. Finally, the monitoring comparison presented in the later part of the chapter showed results (initially unexpected) about the on-line monitoring performance given by the various options and missing data handling methods.

In the author's opinion, this chapter represents the missing piece in the technique for batch process analysis and monitoring as proposed by Nomikos and MacGregor (1994;1995a;1995b). It also gives theoretical foundations to the application of these multiway models for control purposes (Flores-Cerrillo and MacGregor, 2004).

The optimal design of the operating conditions for a batch system using latent variable models is discussed in Chapter 4, where the design methodology proposed by Jaeckle and MacGregor (1998) is applied to the design of complete batch process operation. A parallel technique is proposed to introduce more information about the trends of the trajectories into the multiway model; which is later used in estimating the design. This technique is later reformulated to include an optimal criterion into the operation design.

The main contributions of this chapter are: the proposal of derivative-augmented multiway models for the batch process operation design. The reformulation of the design equations which allows the solution to include operational constraints, and also allowing the optimization of the design estimate. This last technique is illustrated with the estimation of the operating conditions for an industrial emulsion polymerization process with minimal time consumption. Finally, new ideas on how to include partial derivatives of the trajectories with respect to each other in the multiway model are presented. The

benefits of doing so are a noticeable increase in the predictability of the characteristics of the final product as it is illustrated with industrial data.

When the design of operating conditions is being done for a grade of product which has already been produced at another site, then the problem is referred to as the product transfer problem, this is dealt with in Chapter 5. Jaeckle and MacGregor extended their product design approach (1998) to solve the product transfer problem (2000). However, as the authors of this technique admit, this is a suboptimal solution since it does not use all the data structures available in the problem.

Chapter 5 introduces a novel latent variable regression model (referred to as the Joint-Y PLS model) which makes possible the effective transfer of information from the *source* site, onto the *target* one. An industrial scale-up example from the pulp and paper industry is presented.

This main contribution of this chapter is the JYPLS model, its mathematical foundations and parameter estimation options; and its application in solving the product transfer problem by including all the different data sets available. This chapter also provides a study on the scenarios in which it is desirable to include all data structures into the model. This chapter is perhaps the most innovative one in this research work.

Each chapter of this thesis presents an original idea with tremendous potential in its application to industrial problems (4 industrial cases are studied throughout this thesis). A general summary of the overall findings is presented in Chapter 6.

## Chapter 2

### Analysis of Batch Processes using Multivariate Methods and Alignment Techniques

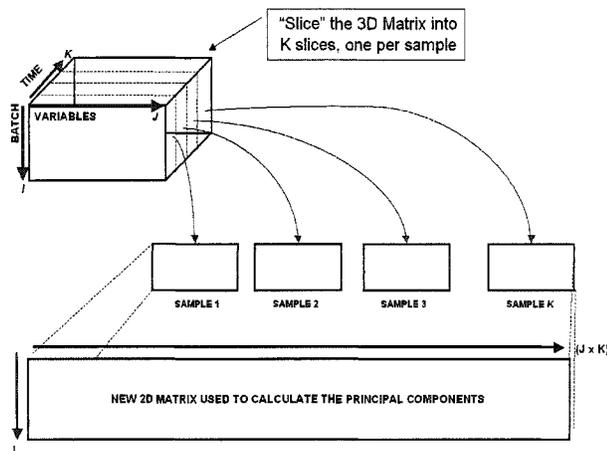
This Chapter presents an application of the methodology introduced by Nomikos and MacGregor (1994) for batch process analysis, in particular to troubleshoot an industrial drying unit from which the product is often off-specification. The data is augmented by integrating the information obtained from the synchronization of the batch trajectories, as suggested by (Westerhuis *et al.*, 1999).

#### 2.1. Introduction

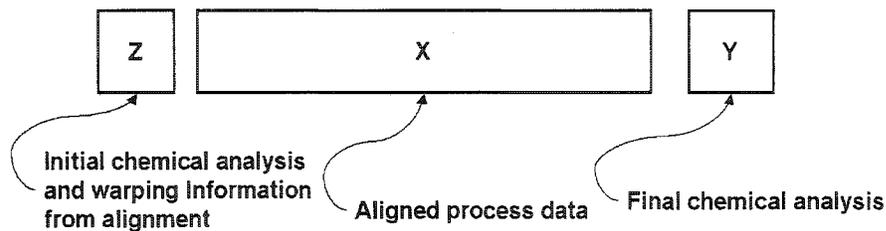
Batch processes play an important role in the pharmaceutical, semiconductor, polymer, and specialty chemical industries. Data collected over the duration of the batch has a time dimension for each process variable. As a result a data set from a batch process can be arranged in a three dimensional array (Fig. 2.1) consisting of data from a number of batches ( $I$ ), with  $J$  process variables measured at  $K$  time intervals over the course of each batch.

Nomikos and MacGregor presented methods for applying multivariate statistical process control (MSPC)(MacGregor and Nomikos, 1992; Nomikos and MacGregor, 1994; 1995b) by using multi-way PCA (Wold *et al.*, 1987) to an “unfolded” matrix of the batch data (Fig. 2.1). This strategy models the deviations of each batch from the mean

trajectory since it mean-centers the unfolded matrix column wise and scales each column to have a variance of one. These multi-way statistical models capture the correlation structure among all the measured variables over the entire time history of the batch, and compress this information down into a low dimensional latent variable space in which it is easy to compare and monitor batches. Nomikos and MacGregor (1995a) extended the methods using multi-way PLS (Wold *et al.*, 1987) to incorporate the final product quality data ( $Y$ ) collected at the end of each batch. Kourti *et al* (1995) further extended the method to incorporate data on initial conditions and other discrete variables, such as operators and information from upstream processes, by using a multi-block, multiway PCA and PLS (Fig. 2.2).



**Figure 2.1 Unfolding of the three way batch data set**



**Figure 2.2 Multi-block multi-way PLS matrices**

The methods proposed by Nomikos and MacGregor have been applied widely in industry, with some applications reported in the literature (Kosanovich, Dahl, and Piovoso, 1996; Neogi and Schlags, 1998; Nomikos, 1996; Schlags and Popule, 2001; Tates *et al.*, 1999) but most are still unpublished. Other variations of these multivariate methods for monitoring batch systems have also been reported (Boque and Smilde, 1999; Wold *et al.*, 1998).

This work presents an industrial application of the method proposed by Nomikos and MacGregor with the additional inclusion of “time usage”, a new trajectory that appears in the alignment exercise.

This chapter is organized as follows: Sections 2.2 to 2.4 describe the process, its operation and its variables, and how the product is classified into on-spec and off-spec product using principal component analysis (PCA) on the product quality data. Section 2.5 describes how the batch trajectories are synchronized, and presents a new approach to utilizing the time warping information from the synchronization. Sections 2.6 and 2.7 describe the PLS modeling studies performed on the data and the use of these models to uncover the major operating problems with the process that are related to the production of poor quality product. Some conclusions are then given in Section 2.8.

## **2.2. The drying process**

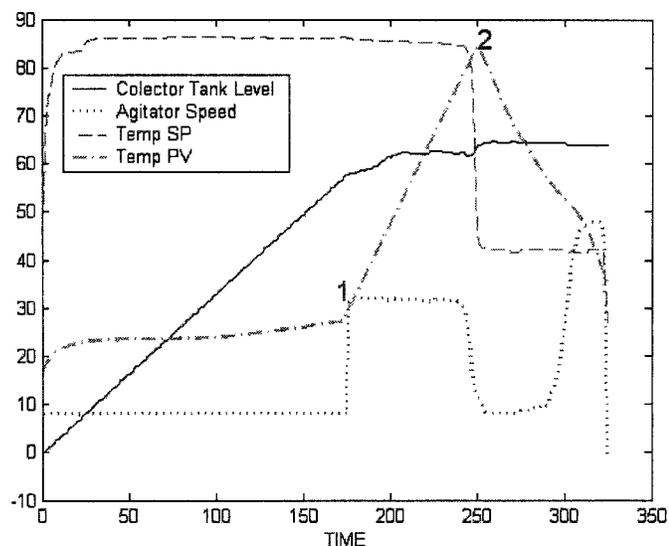
The unit is used to evaporate and collect the solvent contained in the initial charge (wet cake) and to dry the product to a target residual solvent level. During the drying, important chemical structural changes can occur that can lead to unacceptable product quality. With the aid of Figure 2.3, the operation is described as follows:

1. The batch is charged with a mass of wet cake that varies from batch to batch. The weight of the wet cake fed to the dryer is measured for each batch, but the solvent content is unknown

2. At the beginning of the batch, the agitator is running at low speed (~ 8 RPM), and the heating jacket is already running with hot medium. As a result, the temperature in the dryer starts increasing slowly.

3. At a certain point determined by the control system (marked as “1” in Fig. 2.3) the agitator is switched to high-speed (~30 RPM) and the temperature increases rapidly until it reaches its peak (point “2” in Fig. 2.3). The agitator is triggered down to slow-speed just before point “2” is reached.

4. After the temperature peak (point “2”) the product in the batch is cooled down, and then towards the end of the batch, the agitator is turned to high-speed for some time.



**Figure 2.3 Critical Variables and two critical steps in operation**  
(Level in % fill, Temperatures in Celsius, Speed in RPM)

As the batch evolves, the solvent that is being extracted from the wet cake is collected in a tank that is emptied after each batch. The time at which the agitator triggers from low to high speed, and vice versa is different for each batch; also the maximum temperature that the batch reaches is not the same for all batches because the operators adjust this set-point (peak temperature) from time to time to correct the quality of the product in a batch-to-batch manual feedback.

### 2.3. The Data Set

There are three sets of variables measured for each batch: a) An initial chemical analysis on the wet cake done before each batch, and the weight of the wet cake fed to the dryer (11 chemical variables and one weight), b) 10 process variable trajectories as they evolve throughout the batch, and c) 11 product quality variables measured at the end of the batch.

The data on the initial chemical analysis and on the weight of the cake will be referred to as the initial condition matrix ( $Z$ ), the process variable trajectories will be referred to as the process matrix ( $X$ ), and the final chemical properties will be referred to as the quality matrix ( $Y$ ) (Fig. 2.2). A more detailed description as well as the names used for each measured variable is presented in Table 2.1. The whole dataset consists of 71 batches.

Matrix	Variable Name	Description
Z	WGTCAKE	Total weight of the wet cake fed initially to the dryer
Z	Z1, Z2,... Z10	9 organic group concentrations and pH
X	CTANKLVL	Level of the collector tank, always starts in zero (empty)
X	DIFF-PRESS	Differential pressure in the dryer
X	X1	Pressure in the dryer
X	X2	Power to the agitator
X	X3	Torque resistance for the agitator
X	AGITSPEED	Agitator Speed
X	JTEMPsp	Set Point for the jacket heating medium
X	JTEMP	Temperature of the jacket heating medium
X	DTEMPsp	Set Point for the temperature inside the dryer
X	DTEMP	Temperature inside the dryer
Y	Y1,Y2,... Y10	7 organic group concentrations and 3 physical properties
Y	SOLVENT-CONC	Weight percentage of solvent in the final product

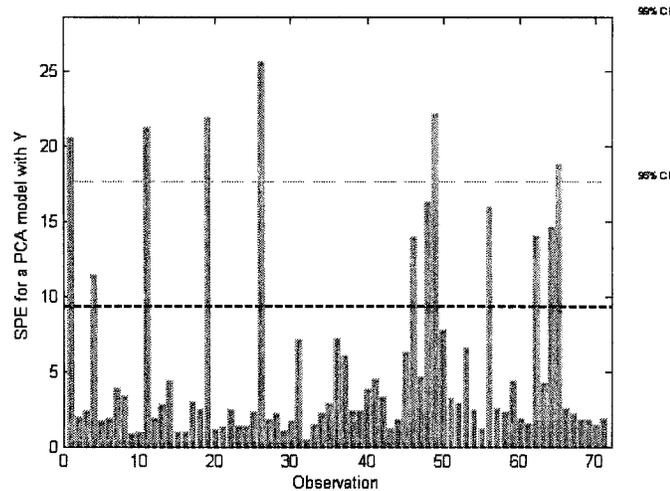
**Table 2.1 Variables measured per batch**

#### 2.4. Multivariate classification of quality

The quality of a product has to be analyzed in a multivariate way, since “*quality is a multivariate property, requiring the correct combination of all the measured characteristics*” (Duchesne and MacGregor, 2003). Therefore to uncover the natural multivariate product classification within the data, we first build a PCA model on the final product properties (Y).

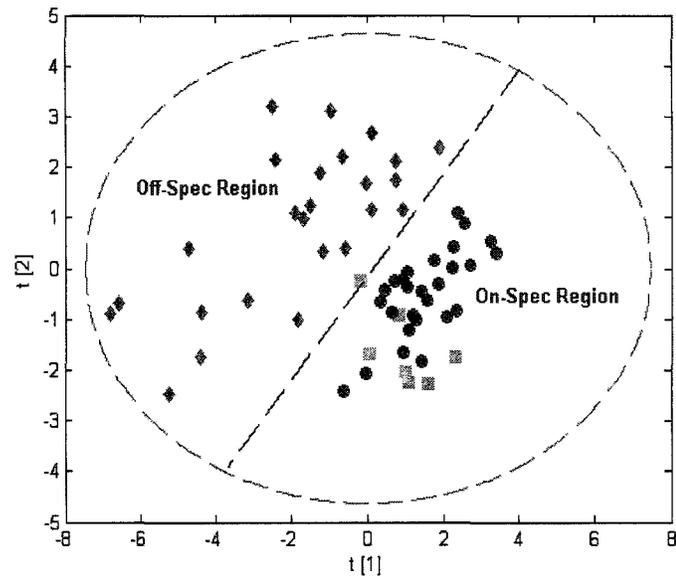
After removing 12 obvious outliers (Fig. 2.4), a final PCA model for Y with 2 components captures 70.0% of the variability in the data (51.2% captured by the 1<sup>st</sup> component, and 18.8% by the 2<sup>nd</sup> component). The t1-t2 score plot (Fig. 2.5) shows a natural clustering of what the company had classified as on-spec and off-spec products,

the dashed line ellipse represents the approximate 99% confidence region in this model and contains the 59 used observations.

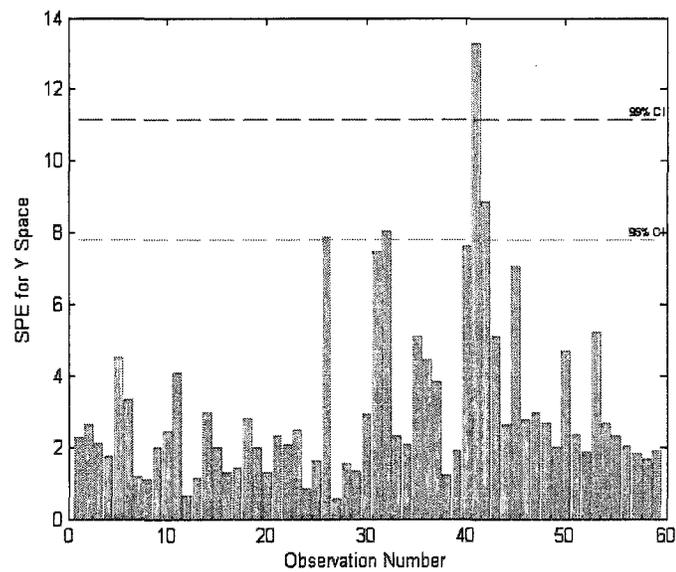


**Figure 2.4 SPE after 2 components in a PCA for Y variables  
(Observations above the bold dashed line were considered outliers)**

The dashed line in Fig. 2.5 (drawn by eye) is drawn to show the separation between good and poor quality product, and it clearly shows the separation between good and bad quality product much more clearly than looking at the 11 Y variables separately. The model can be easily used to discriminate the quality of the final product by its position in the  $t_1$ - $t_2$  plane. Figure 2.6, shows the square prediction error (residuals of the PCA model) for each batch, together with their 95% and 99% confidence limits (Nomikos and MacGregor, 1995b). Aside from batch 41, the PCA model appears to explain the quality variation from the batches.



**Figure 2.5 PCA score plot for product quality data (Y).  
Off-spec (◆), on-spec (●), on-spec but high residual solvent (■)**



**Figure 2.6 SPE for the Y matrix multivariate classification model**

## 2.5. Alignment of the batch data

One of the issues still under investigation in the application of multivariate methods for batch process analysis is how to handle a set of batches with different duration; this is referred to as the “alignment” or “synchronization” problem. To “align” or “synchronize” a data set of batch trajectories means to perform a certain transformation on each trajectory in such way that at the end of the “alignment” all the batch trajectories will line up and “evolve” similarly, and have the same number of samples.

### 2.5.1. Alignment Technique

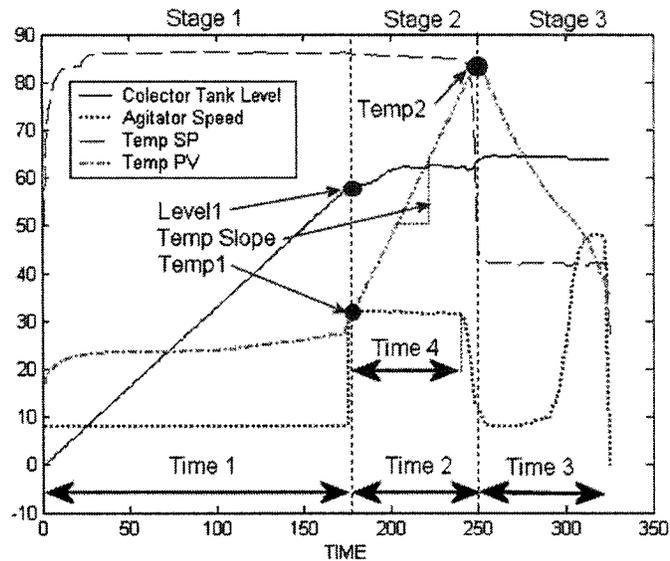
In this industrial drying process, the amount of wet cake, the initial concentration of solvent and some operating conditions for each of the batches are different. As a result, the duration of each batch, and the duration of different stages within each batch, are different; thus the need for performing an alignment on the trajectories in order to make them comparable. Nomikos and MacGregor (Nomikos and MacGregor, 1995b) proposed the usage of an indicator variable in order to re-sample the trajectories and perform the alignment (e.g. sample at 1% increments on conversion). Kassidas *et al* (Kassidas, MacGregor, and Taylor, 1998) proposed the use of Dynamic Time Warping, a technique used in speech recognition, in order to align the batch data when an indicator variable is not obvious.

In these two suggested techniques, the authors emphasized the importance of not discarding the information that comes out of an alignment (e.g. the time required to complete any given stage). This information will be called “warping information” since it is a new piece of data for each batch, and it is not available until the alignment is performed.

There are many ways to use an indicator variable to align batch trajectories. If there exists a monotonically increasing variable that always starts each batch at the same value and ends the batch at another given value, then alignment can often be achieved simply by re-sampling at constant intervals of the variable. Industrial examples of such indicator variables are conversion information (Neogi and Schlags, 1998), and the cumulative weight of a key monomer added during each batch (Kourti, Lee, and MacGregor, 1996). If such an indicator variable does not exist over the whole duration of the batch, one may exist within each stage of the batch, allowing for a stage by stage alignment.

The discrete events that, most of the time, determine the transition between stages within a batch, are triggered by operators or by the automation system in response to the achievement of certain conditions during the preceding stage. Kaistha and Moore (Kaistha and Moore, 2001) proposed a mathematical filter to extract the events from the batch trajectories. However, in practice we often already know the events and there is no need to identify them from the trajectory.

In this industrial study, no one “key” monotonically increasing or decreasing indicator variable existed during the batch. Therefore, to align the batches, we used prior knowledge about how the process was operated to define 3 clear stages in the batch (Fig. 2.7). Stage 1 runs from the beginning of the batch and up to the time where the agitator is turned to high-speed; stage 2 runs from this point and up to the time where the temperature reaches its peak; and stage 3 is from this point until the end of the batch.



**Figure 2.7 Stages and new variables defined during the batch  
(Level in % fill, Temperatures in Celsius, Speed in RPM)**

To align the batches we use 3 different indicator variables (one per stage). For stage 1 the chosen indicator variable is the level in the collector tank. The value of this level at the end of the stage is very different for each batch so it is not proper to simply re-sample using a fixed  $\Delta Level$  for all batches. Therefore, for any batch  $i$ , we assume that stage 1 is 0% complete when the tank is empty (time zero), and it is 100% complete when the level reaches the  $Level1_i$  value at the end of the stage; to achieve  $n$  samples between 0 and 100% of completion, each batch  $i$  is re-sampled at level increments given by  $\Delta Level_i = Level1_i / (n - 1)$ , and linear interpolation is used when needed. Aligning this way assures an equal number of samples, and a “line up” in the percentage of completion for all batches in stage 1.

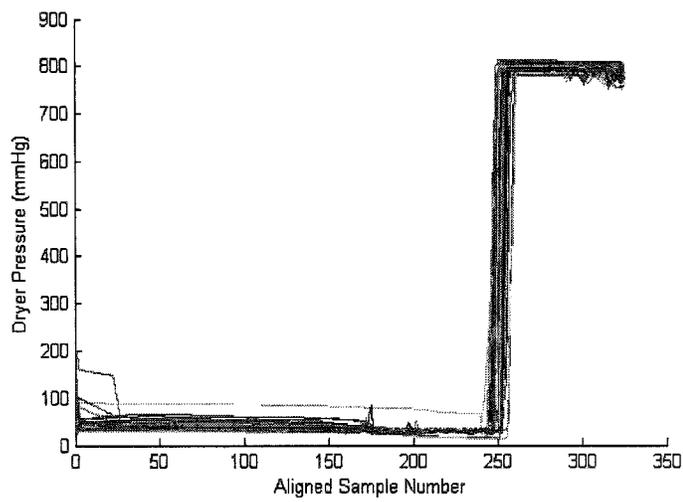
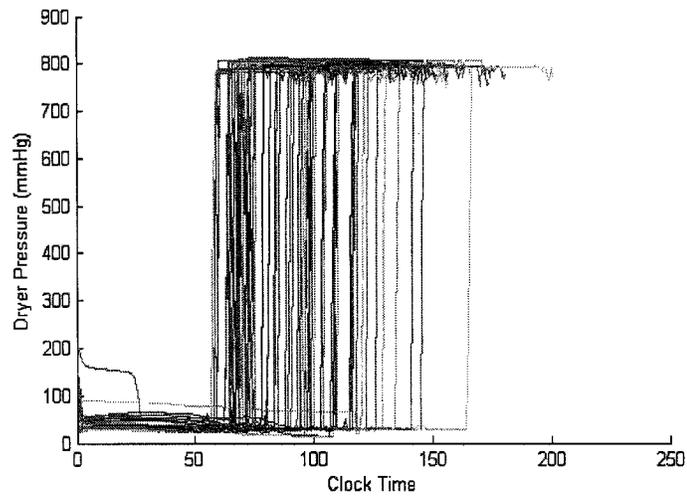
For stage 2 the indicator variable is the dryer temperature. The alignment strategy is similar to that for stage 1, because the initial and final temperatures in this stage ( $Temp1$  and  $Temp2$ ) are very different for each batch. In this case stage 2 is completed 0% when the temperature is at its  $Temp1_i$  value, and is completed 100% when the temperature

reaches its peak value  $Temp2_i$ . For each batch we calculate a  $\Delta Temp_i$  and all variables (for stage 2) are re-sampled using this increment.

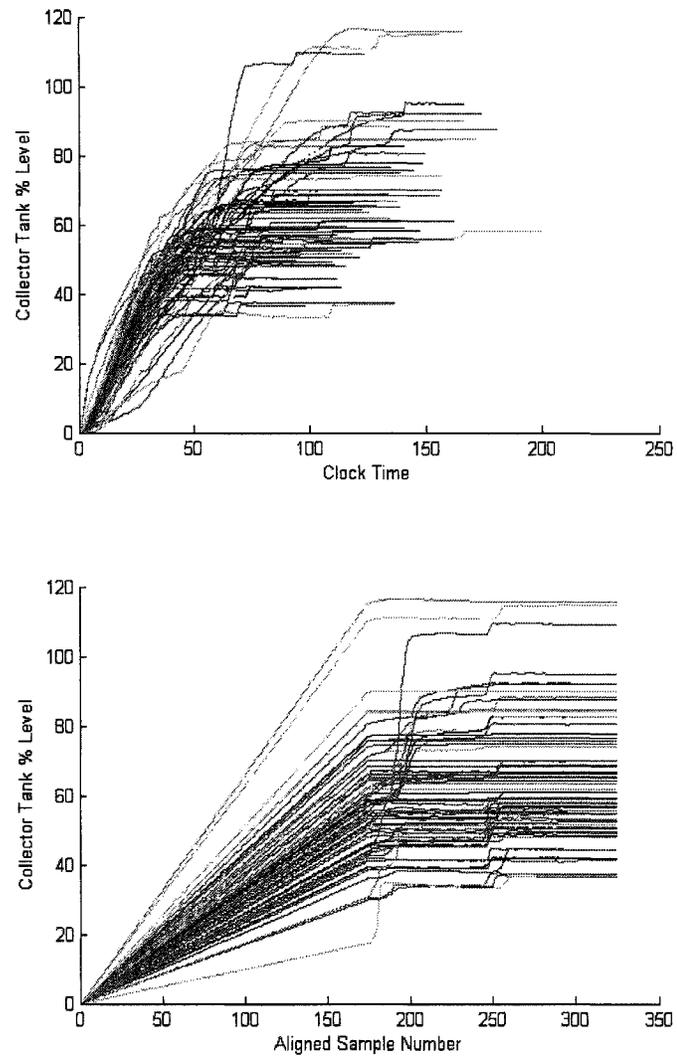
In stage 3 there is no variable that can be used as indicator (the fact that the temperature in this stage appears monotonic in Fig. 2.7 is because this is a great mean from all batches), so we re-sample linearly with time in order to obtain 75 samples total from the start of stage 3 to the end of the batch. However, the total time in stage 3 for each batch is recorded in the  $Z_w$  and, as discussed later in Section 2.4.2, detailed warping information is also introduced as a new variable in the X matrix. In this way information on the duration of this stage and on the variation of the time usage throughout the batch is retained in the next PLS models developed in further Sections.

The alignment obtained can be appreciated by plotting some variables for all batches (Fig. 2.8.) All batches now have 325 samples each, and all the stages are synchronized making them comparable. Figures 2.9 and 2.10 show the aligned trajectories of the two indicator variables used: the level in the collector tank and the dryer temperature. With the definition of these stages we can calculate eight new discrete variables per batch that define the alignment information of the batches. This data is added to the Z matrix (Fig. 2.2) and will be included in the analysis in the later Sections:

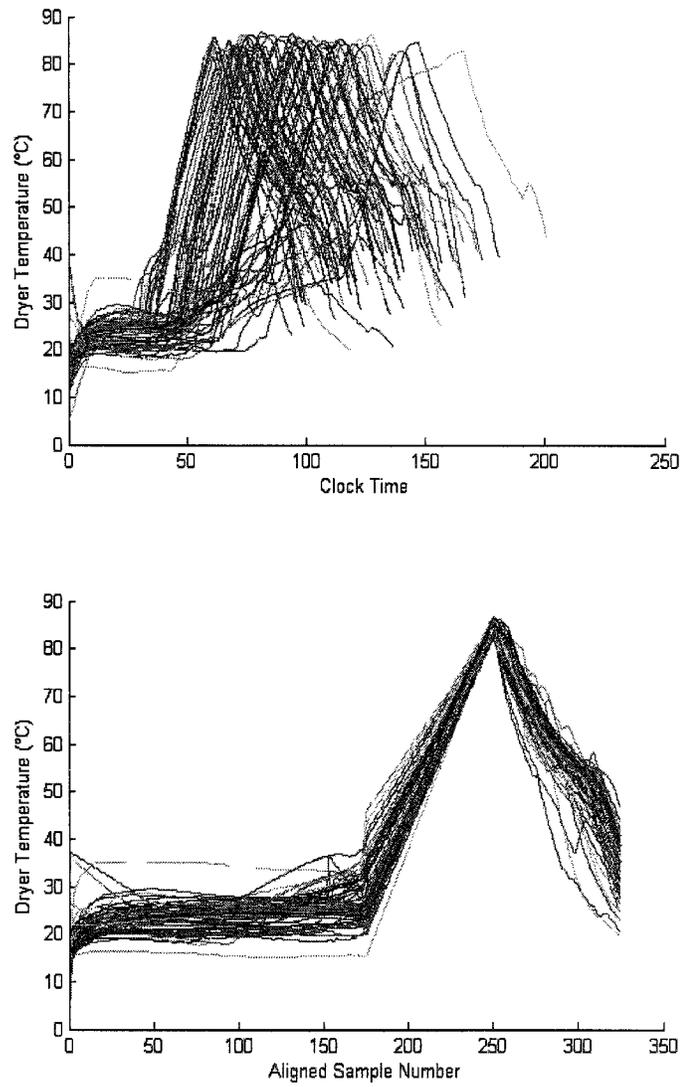
2. *Level1*: The level in the collector tank at the end of stage 1.
3. *Temp1*: The temperature in the dryer at the end of stage 1.
4. *Temp2*: The temperature in the dryer at the end of stage 2 (peak).
5. *Time1*: Total length of stage 1 (time samples).
6. *Time2*: Total length of stage 2 (time samples).
7. *Time3*: Total length of stage 3 (time samples).
8. *Time4*: Total length of high-speed agitation (time samples).
9. *Temp Slope*: Slope of the dryer temperature trajectory in stage 2.



**Figure 2.8 Variable 3 for all batches before (top) and after (bottom) alignment**



**Figure 2.9 Collector tank level variable for all batches before (top) and after (bottom) alignment**



**Figure 2.10 Dryer-temperature variable for all batches before (top) and after (bottom) alignment**

### 2.5.2. Inclusion of “Time usage” as one extra trajectory per batch

From the new  $Z$  variables obtained from the alignment, four of them have the “signature” on how the batch “used” time (Time1, Time2, Time3, and Time4). These are indeed four discrete values of time that summarize important information on timing differences among the batches. However, there is a more detailed way that we can extract timing information from the aligning exercise. In the original trajectories, time has a linear evolution for all batches. However, in the alignment process, this time is stretched and/or compressed as we re-sample. If we include “Time” as an extra variable for each batch, and we re-sample this variable along with all the others; at the end we will have a very useful trajectory that will tell us how each batch was “using” time as it evolved (Westerhuis *et al.*, 1999). Therefore, batches, which evolve faster or slower at different stages, are readily apparent. Furthermore, if a fault is present in a particular batch, we will be able to say at what percentage of completion of the batch it happened. Figure 2.11 shows how the “time” is distorted when we re-sample the batch trajectories for this particular data set. These trajectories are added to the  $X$  matrix (as the 11<sup>th</sup> variable) and will be analyzed together with the other process variables in later Sections.

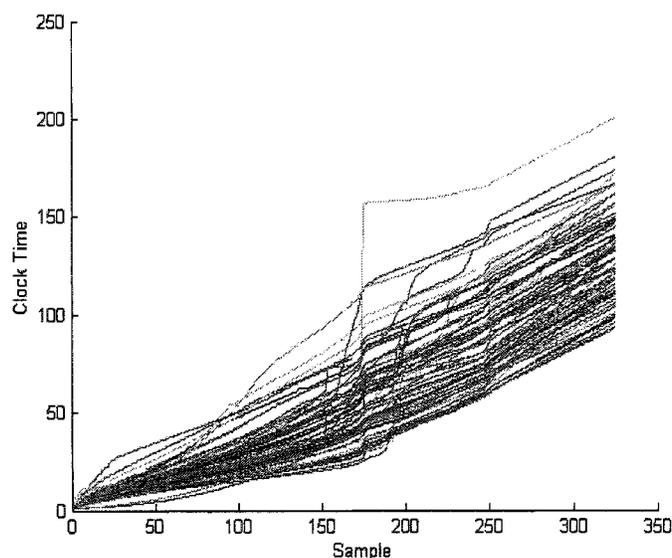


Figure 2.11 Distortion of clock time with the alignment of batch data

From an operational perspective, this *time usage* is of tremendous importance since, in most cases time represents the most expensive resource being consumed by the batch. Numerous deterministic studies are found in the literature, where the batch length is part of the overall objective function to minimize and compute an optimal trajectory (Biegler, Cervantes, and Wachter, 2002; Floudas and Pardalos 2001; Vassiliadis, Sargent, and Pantelides, 1994a; Vassiliadis, Sargent, and Pantelides, 1994b). This particular topic will be treated in the later Sections of chapter three of this thesis.

## **2.6. Analysis of historical batch data**

The objective of this Section is to uncover possible causes for the occurrence of such a large number of batches that are producing off-specification product. PLS regression models are built to relate the final product quality ( $Y$ ) to the aligned process trajectory data ( $X$ ), and to the matrix ( $Z$ ) containing the initial conditions for the batches (chemistry and charge) and the discrete timing variables. These PLS models project the information contained in the very large data matrices down into low dimensional latent vector spaces where it becomes much easier to visualize and compare the behavior of the batches and to diagnose the operational problems.

The block nature of the data ( $Z, X, Y$ ) matrices available in this study naturally makes this problem suitable for analysis by multi-block PLS methods (Wangen and Kowalski, 1989; Westerhuis and Smilde, 2001). Multi-block PLS methods give the same prediction models for  $Y$  as a single block PLS if the same weighting and scaling is applied to each variable (Westerhuis, Kourti, and MacGregor, 1998); but additionally allow for estimation of the separate effect of each block (see Appendix 1 for a brief example of a multi-block modeling exercise). In this study, with only two regression blocks ( $Z, X$ ), separate single block PLS models relating  $Z$  to  $Y$ , and  $X$  to  $Y$  were found to provide essentially the same information as the multi-block approach. Therefore, for simplicity, only the former approach will be presented.

The analysis is presented in two Sections; the first one is to understand the impact that the variables in the  $Z$  matrix have on quality, and the second one to understand the impact that the trajectory variables in the  $X$  matrix have on quality.

### **2.6.1. Impact of initial conditions and warping information ( $Z$ ) on final quality ( $Y$ )**

The  $Z$  matrix, at this point of the analysis, contains two different types of variables: a) The information from the alignment and b) the initial chemical and physical analysis. It was decided to partition the  $Z$  matrix into two blocks  $Z_w$  and  $Z_{ch}$ .

$Z_w$  is made upon 9 columns of  $Z$  representing the variables arisen from the alignment (or warping), plus the wet cake weight (Level1, Temp1, Temp2, Time1, Time2, Time3, Time4, TempSlope and WgtCake); and  $Z_{ch}$  is made upon the 11 columns of  $Z$  representing the chemical analyses, ( $Z_1, Z_2, Z_3 \dots Z_{11}$ ). It was decided to place the wet cake weight with the warping information because then all operating variables of the batch are together and separated from all the chemical analyses.

The first step taken in this analysis is to remove from the data the 12 outliers found in Section 2.3, and 15 more observations for which the whole  $Z_{ch}$  information was missing. The final data set consists of 44 batches and 21 variables grouped in two blocks. These 44 batches are now used in the rest of the chapter.

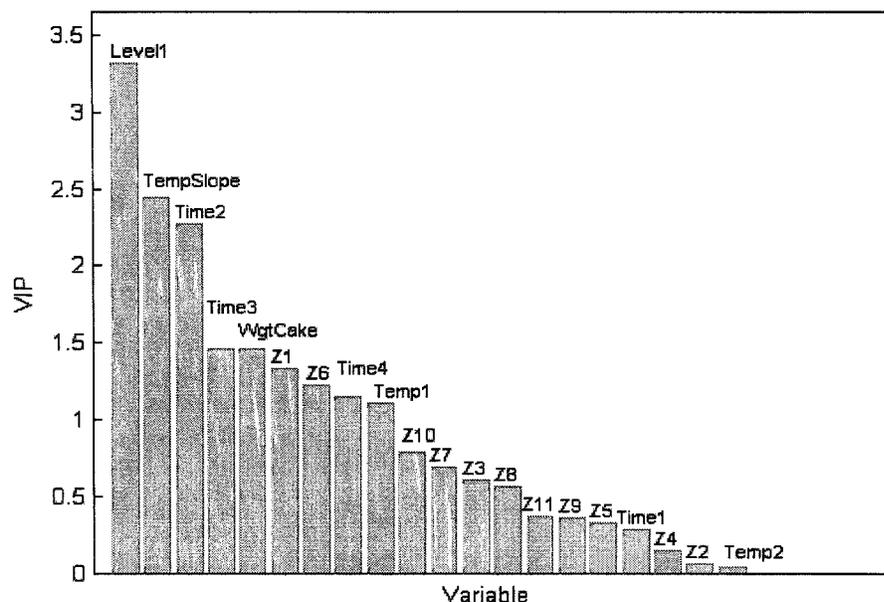
A PLS is fitted with the whole  $Z$  matrix, and for illustrative purposes, two more models are built individually for each block of  $Z$  ( $Z_w$  and  $Z_{ch}$ ). The percentage of the variance that each model captures is listed on Table 2.2; all models are built using 2 latent variables, as determined by cross-validation. As expected; none of the individual models captures as much variance as they do together. The fact that the combined

captured variance (50%) is less than the sum of both individual captured variances ( $37.14+21.52=58.66$ ) is an indication that both blocks are not entirely orthogonal to each other. This is because some of the variables in the  $Z_w$  block are correlated with some of those in the  $Z_{ch}$  block. This analysis also shows that the process operating variables  $Z_w$  have a much greater impact on the product quality (Y) than the initial charge chemistry. This is an important result as will be confirmed later.

Model	%Variance of Y Explained
Z-Y PLS	50.49
$Z_w$ - Y PLS	37.14%
$Z_{ch}$ - Y PLS	21.52%

**Table 2.2 Percentage variance of Y explained by various PLS models with Z**

The PLS of the whole Z matrix versus Y may be used to show the relative importance of each of the variables in the model. The VIP (variable importance to the projection) (Eriksson *et al.* 1999) plot in Fig. 2.12 lists the variables in Z in their order of importance. Several important conclusions arise from this VIP plot; (i) it confirms the previous conclusion that the most important variables are those variables related to the operation itself, and not those related to the initial chemistry; (ii) the most important variable is the initial amount of solvent in the wet-cake; (iii) the variable “Temp2” appears to be the least important of all the variables, in spite of the fact that this variable was the one “adjusted” in the plant in order to correct the quality. Since this manipulated variable is involved in some sort of feedback loop to control some aspect of the final quality, its relationship with Y will be some combination of the open loop effect and the close loop effect on Y, and such result is not surprising.



**Figure 2.12 VIP Plot for Z -Y PLS model**

Additional important results can be obtained by looking at the score space of this model (Fig. 2.13). It is clear from this plot, that most of the good quality batches have a positive value of  $t_1$ , while the poor quality one have a negative value of  $t_1$ .

If the loadings for  $Z_w$  (Fig. 2.14) and  $Z_{ch}$  (Fig. 2.15) are now analyzed in the light of this new piece of information (good quality batches have high positive values of  $t_1$ ), we can tell much more about the process. From the loadings map for the  $Z_w$  variables (Fig. 2.14) it is clear that high quality product is associated with a high temperature slope in stage 2 and low values of the length of stage 2 (Time 2) and low high-speed agitation time (Time 4). All these imply that fast evaporation stage 2 is desirable. High quality product is also associated with low solvent level in the collector tank at the end of stage 1 (Level 1) and a low initial charge of wet cake (Wgtcake), both implying that a smaller charge of wet cake (and hence low solvent) is desirable. A slow cool-down in stage 3 (Time 3) and possibly a low Temp 1 might also seem desirable. From the loadings for the  $Z_{ch}$  variables (Fig. 2.15) it also appears that high quality might also be related to higher levels of chemicals Z6 and Z7 lower levels of Z1, in the wet cake.

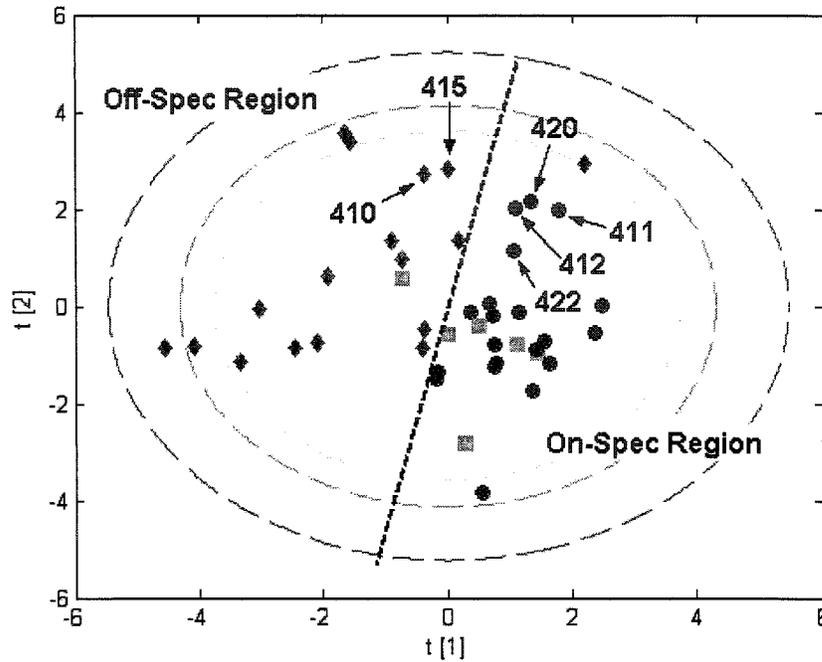


Figure 2.13 Score plot for the Z – Y PLS model

Off-spec ( $\blacklozenge$ ), on-spec ( $\bullet$ ), on-spec but high residual solvent ( $\blacksquare$ )  
 (Numbered batches are references for analysis in Section 4.3)

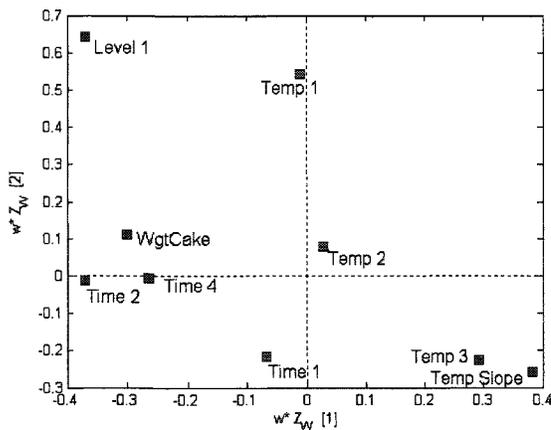


Figure 2.14 Loadings plot for the  $Z_w$  Variables in the Z – Y PLS model

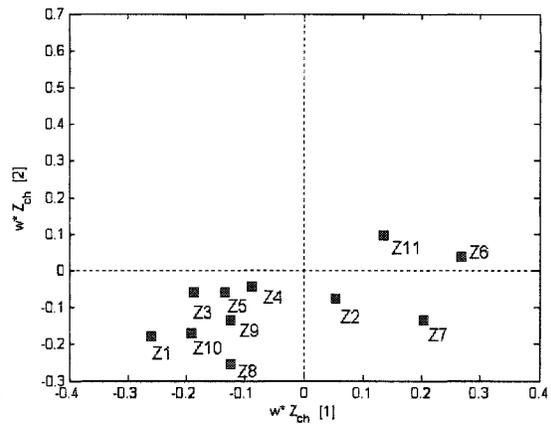
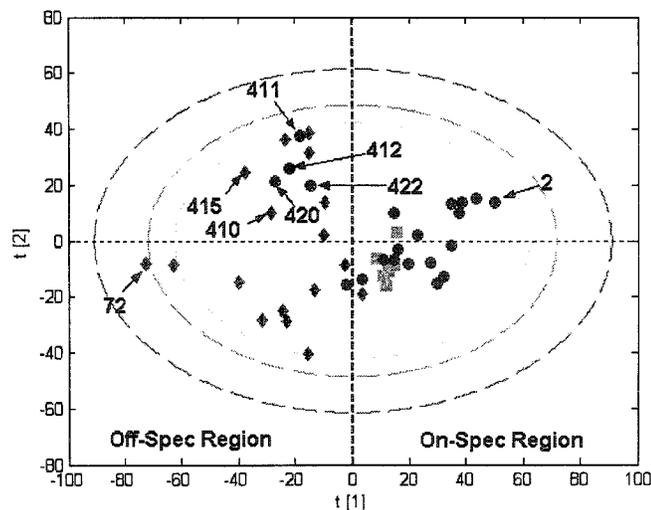


Figure 2.15 Loadings plot for the  $Z_{ch}$  variables in the Z – Y PLS model

### 2.6.2. Impact of process variable trajectories on the final quality

This Section covers the analysis on how the shape and timing of the process variable trajectories (X) affect the final quality. It is important to keep in mind that the variables are evolving with time and the effect of a variable at different points of time is not the same. The analysis will not be done on single variables, but on their trajectories summarized in the X matrix. The X matrix has 11 variables – 10 process variables and the clock time from the alignment – each with 325 samples per batch giving a final unfolded X matrix with 3575 columns (variables at all time intervals) and 44 rows (batches). The multi-way PLS model captures 36% of the variance of the X matrix, and explains 48% of the Y matrix with 2 latent variables.

The score-plot for the X space is shown in Fig. 2.16. Most of the batches with similar quality do cluster together except for batches 412, 411, 420 and 422; these batches will be discussed later in Section 2.7. Before doing so, we need to interpret the model in order to better understand the position of the batches in the score plot.



**Figure 2.16** Score plot for the X space in the X-Y MPLS.

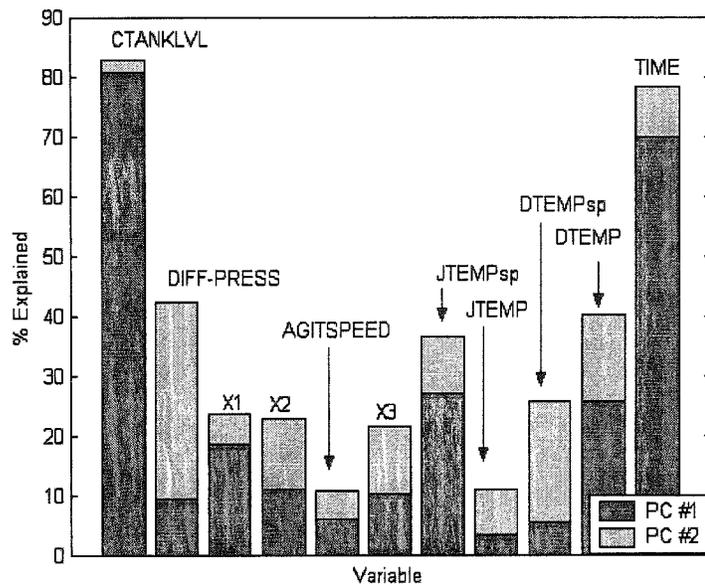
Off-spec batches ( $\blacklozenge$ ), on-spec batches ( $\bullet$  and  $\blacksquare$ )

(Numbered batches are references for analysis in Section 2.6)

### Interpretation of the multi-way PLS model

#### *Captured Variance*

The way the model is capturing the variable trajectory information can be analyzed in different ways since each variable in the X matrix varies with time, and has a different importance in each time interval. Although the model uses only 36% of the information in the X, this is the average for all variables at all time samples: Figure 2.17 shows the total percentage explained per variable, per component; the variables CTANKLVL, DIFF-PRESS, JTEMPsp, DTEMP and TIME are variables that are used much more fully by the model and hence are the most important ones in describing quality. Notice the high importance of the time usage; this re-enforces our conclusion of the past Section about the importance of timing. In order to better understand the importance of each of the process variables over their time history; we will analyze the weights of each variable trajectory as it changes with time.

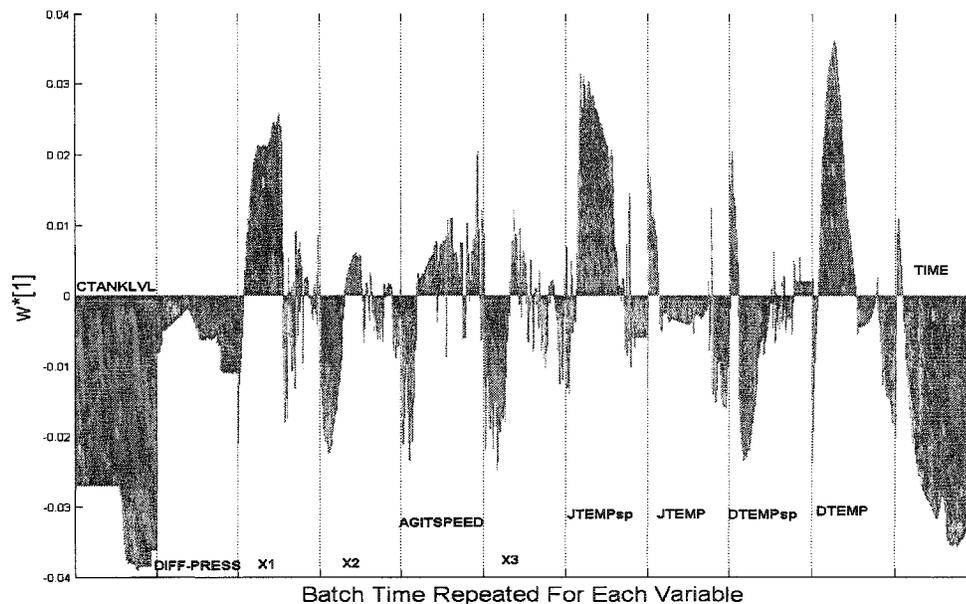


**Figure 2.17 Percentage captured per variable in the X matrix**

### *Analysis of the PLS Model Weights*

Since each variable in the X matrix is a trajectory, their loadings (or weights in the case of PLS) are not single values but are trajectories indicating the importance of each variable at each point of time to the latent variables. If, for the first component, we plot the weight trajectory for variable 1 from sample 1 to sample 325 and beside it we plot the weight trajectory for variable 2 from sample 1 to sample 325 and we continue in a similar manner for all variables, we end up with Figure 2.18. The magnitude of the weight, for a certain variable, at a certain point in time, is an indication of how important that variable is, at that point in time, for the first component ( $t_1$ ).

It is clear from the score plot in Fig. 2.16 that a high positive value of  $t_1$  leads to high quality product. With this in mind we can therefore interpret the trajectory loadings in Fig. 2.18.



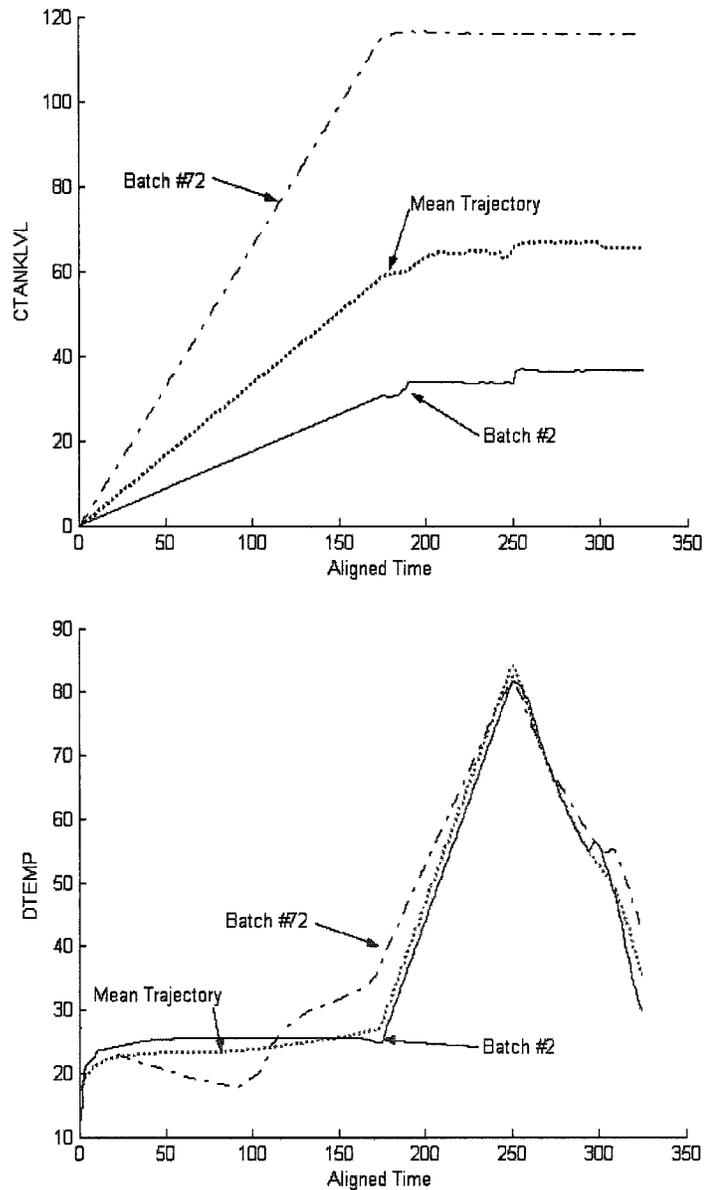
**Figure 2.18 Weights for 1<sup>st</sup> component in the X space for the XY MPLS**

A high quality batch will have the following characteristics:

1. It will have low levels of the solvent collector tank level (CTANKLVL) with respect to its mean trajectory from the model, throughout the entire batch. This result is clearly consistent with the importance found previously in the PLS analysis of the Z variables, that a low wet cake fed to the dryer (wgtcake) and a low level in the collector tank at the end of stage 1 (Level 1) are key factors for on-spec product quality.
2. It will have high levels of pressure in the dryer (X1), the set point for the jacket heating medium (JTEMPsp), and the temperature inside the dryer (DTEMP) with respect to each of their mean trajectories from the model, for the first stage of the batch. These all appear to stress the importance of applying a high level of heating during stage 1.
3. It will have low values of the time variable (TIME) relative to its mean trajectory from the model, throughout most of the batch. This implies that batches that progress faster tend to be those with high product quality

To illustrate this, consider two batches on opposite sides of the score plot in the  $t_1$  dimension (Fig. 2.16), batch 72 (which has a high negative value of  $t_1$ ), and batch 2 (which has a high positive value of  $t_1$ ). For these batches we plot the collector tank level (CTANKLVL) for each of them along with the mean trajectory (Figure 2.19 top) and we see that the trajectory for this variable in batch 2 is always below the mean as opposed to batch 72. If we plot the temperature DTEMP for these batches (Figure 2.19 bottom) we see that batch 2 has a value of DTEMP above the mean for the first stage and below the mean for the second and third stages of the batch; as opposed to batch 72 which has a value below the mean for the first stage and above the mean for the second and third stages of the batch.

The above analysis performed on the variable trajectory weighting for the  $t_1$  variable clearly provided insights into the operating policies that appear to lead to good or poor quality product.



**Figure 2.19 Differences in the trajectory of the collector tank level and temperature between batches 72 and 2**

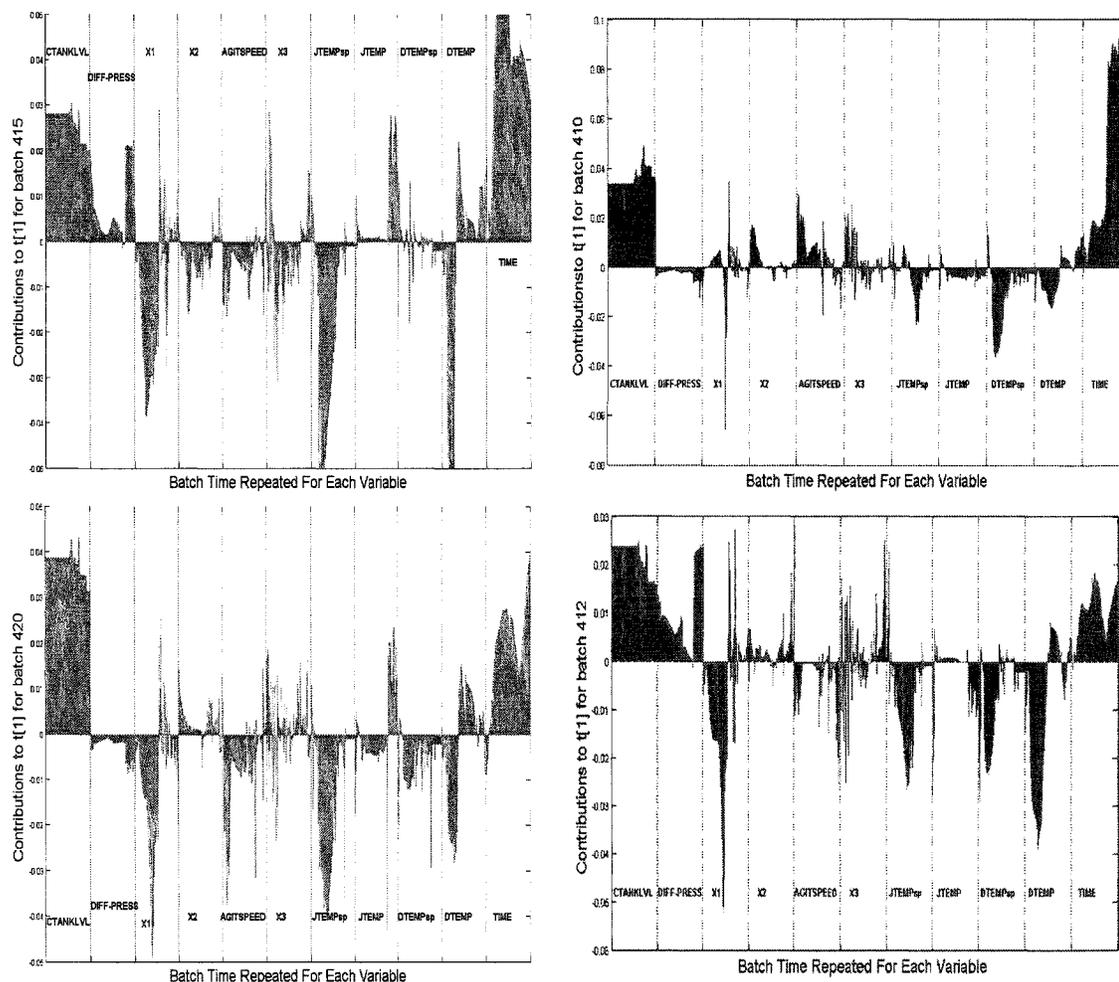
### 2.7. The Anomalous Batches 420, 411, 412 and 422

In Figure 2.16 four batches were observed to lie in the off-spec region of the  $t_1$ - $t_2$  score plot for the X-Y MPLS model. These same four batches clustered in the on-spec region in the Z-Y PLS score plot (Fig. 2.13) and eventually yielded good product at the end. The fact that these four batches cluster in the off-spec region in Figure 2.16 means that the process variable trajectories of these batches have the characteristics of other “bad” batches. However, it appears that some conditions in the Z matrix were sufficient to counter-balance these bad trajectories.

To reassure the fact that the trajectories of these four batches have the same characteristics of any other ‘bad’ batch, we compared (looking for similarities) the contributions plots (Kourti and MacGregor, 1996; MacGregor *et al.*, 1994; Miller, Swanson, and Heckler, 1998), in  $t_1$  (using the X-Y PLS model) for these four batches with the contribution plots in  $t_1$  of other ‘bad’ batches in the vicinity of these batches.  $-t_1$  is used because this component defines the on-spec/off-spec division, as illustrated in Figure 2.16.

Figure 2.20 shows the contribution plots for  $t_1$  for batches 420 and 412 (anomalous batches) and 415 and 410 (bad batches) in the XY-MPLS model. Both plots show the same characteristics: the trajectory for the collector tank level (CTANKLVL) is above the mean during the entire trajectory; the variable X1, JTEMPsp, and DTEMP are all below their mean during the first stage, implying too slow heating and this contributing to a long CLOCK-TIME. All these conditions are totally contrary to the ones related with good On-Spec quality product, according to our earlier conclusions. These contributions are very similar with other nearby batches in the score plot (like batches 411 and 422).

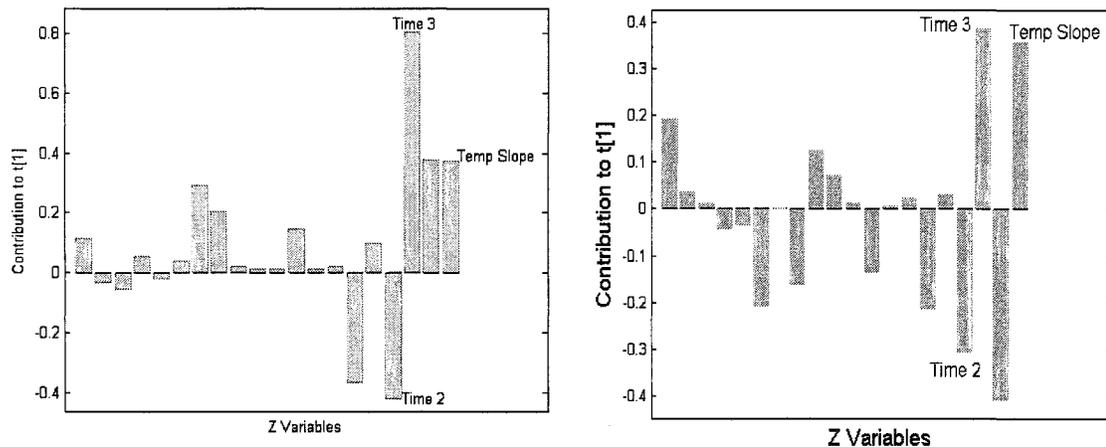
Now consider the Z-Y PLS model score plot (Fig. 2.13). Most of the ‘bad’ batches that are near the anomalous ones in the X-Y score space (Fig. 2.16), appear apart in the Z-Y score space (Fig. 2.13) e.g. batches 415 and 420, 410 and 412.



**Figure 2.20 Contribution plots to  $t_1$  in the XY-MPLS model for bad batches 415 and 410 (top), and anomalous batches 420 and 412 (bottom)**

Figure 2.21 is a contribution plot for  $t_1$  in the score space for the ZY-PLS model, *from* bad baches (415 and 410) *to* anomalous batches (420 and 412). This contribution

plot shows that anomalous batches had a steeper slope for the temperature in stage 2 (high value of Temp Slope and low value of Time 2) and a much larger cool-down time (Time 3). These two characteristics (high evaporation speed in stage 2, large cool-down time) also appear in all the contribution plots between the other anomalous batches and the bad batches; indicating that a fast evaporation in stage 2 and a large cool-down time are “strong” enough characteristics to compensate all the other adverse conditions in the trajectories, and made these batches yield a good product at the end.



**Figure 2.21 Contribution plots to the scores  $t_1$  in the ZY-PLS model from batch 415 to batch 420 (left) and from 410 to 412 (right)**

## 2.8. Conclusions.

Multivariate statistical methods were used to analyze historical data from an industrial batch drying process to uncover possible causes for a high level of poor quality product. The methods were able to show that the initial chemistry of the charge had little impact on the quality, and that it was variation in the operating policies of the batches that were the dominant contributions to the poor quality product. These methods were further able to isolate individual variables such as the initial wet cake charge, timing

variables such as the duration of the second and third stage, and process variables trajectories such as the temperature and pressure profiles during the first stage of the batch that were highly related to poor final product quality. This study illustrates the potential of these multivariate methods for analyzing historical batch data and for suggesting process improvements. Some new approaches to the alignment of the batch trajectories and the subsequent use of the alignment information in the analysis were introduced.

# Chapter 3

## Model Predictive Monitoring for Batch Processes

The purpose of this chapter is to uncover the implications and forecasting mechanism behind a batch-monitoring scheme built with the method proposed by Nomikos and MacGregor (1994,1995a, 1995b) when the future unknown part of the trajectory (for the ongoing batch) is treated as *missing data*. This is contrasted with other forecasting options such as a multivariate time series model, the unconditional mean assumption and the current deviation from the mean. Furthermore the differences and implications of the several methods available to handle *missing data* are exposed and discussed in the context of batch process monitoring. All the forecasting options and missing data methods considered are then put in perspective in a fault detection comparison exercise.

### 3.1. Introduction

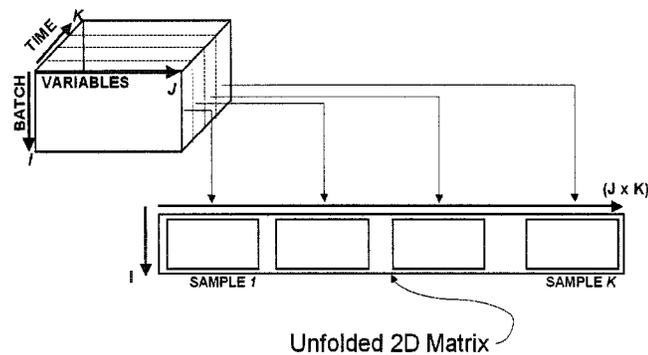
Multivariate methods based-on multi-way PCA and PLS (Wold et al., 1987) have proven their usefulness in batch analysis and monitoring (Kourti, Nomikos, and MacGregor, 1995;Nomikos, 1996;Nomikos and MacGregor, 1994;1995a;1995b). These methods have been widely studied and extensively applied in industry (Kosanovich, Dahl, and Piovoso, 1996;Nomikos, 1996;Ramaker et al., 2002;Schlags and Popule, 2001;Tates et al., 1999). They are essentially the only powerful approaches for the analysis and monitoring batch processes when there is no deterministic model available. Other variations of these multivariate approaches (Boque and Smilde, 1999;Wold et al., 1998) are found in the literature, and have been compared (van Sprang et al.,

2002;Westerhuis, Kourti, and MacGregor, 1999). This chapter focuses on the approach proposed by Kourti, Nomikos and MacGregor (1995;Nomikos, 1996;Nomikos and MacGregor, 1994;1995a;1995b), and in particular in the model predictive estimation of the missing trajectory information needed in the monitoring phase of the approach, and in batch control problems (Flores-Cerrillo and MacGregor, 2003).

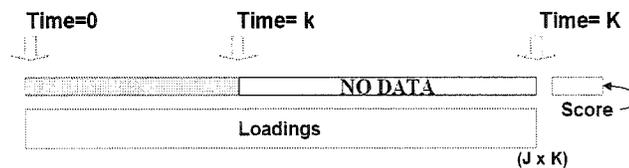
The methods introduced by Nomikos and MacGregor and later developments are very well covered in the review by Kourti (2003). In general, there are two stages: The first stage involves an analysis of a historical set of data to troubleshoot the process and correct for identified sources of unwanted deviation by modifying the process or the operating strategy. A recent industrial example of this analysis stage can be found in the work by Garcia *et al* (2003). Once the process has been analyzed and possibly corrected for any problems, then the second stage involves collecting a new set of data that defines the “common-cause variations” of the “Normal Operating Conditions” (NOC) for the process and this is used to build the monitoring model and to establish control limits for monitoring. The progress of new batches can then be monitored by testing if their operation remains within the established control limits. To assess a new batch in this monitoring stage there are two options: a) to wait until the end of the batch run and then assess its operation or b) to assess the batch in real-time at every time point throughout its execution. Industrial applications of each of these options for monitoring are found in early work by Nomikos (1996) and Nomikos and MacGregor (1995b). The focus of the present work is on the latter of these two monitoring approaches, namely on the real-time monitoring of the batch run as it evolves in time.

In the Nomikos-MacGregor method the 3 dimensional batch dataset consisting of  $J$  variables,  $I$  batches and  $K$  time periods is rearranged as shown in Fig. 3.1. A natural problem to overcome when monitoring a new batch at time= $k$ , where  $k < K$  is the fact that there are  $K-k$  unknown samples (Fig. 3.2) and therefore, the need to estimate or “fill in” these unknown samples for the ongoing batch in order to calculate a score value (note

that this problem does not exist when building the PCA model, when analyzing historical data, or when monitoring only at the end of each new batch). However, for real-time monitoring the quality of the current batch will depend strongly upon how events up to the current time will impact process variable trajectories for the remainder of the batch. Therefore, inference on these future trajectories is unavoidable in any efficient monitoring scheme.



**Figure 3.1 Unfolding of the 3D batch data matrix**



**Figure 3.2 Unknown samples in monitoring a new batch**

In their work, Nomikos and MacGregor (1994;1995b) acknowledge this issue and propose three different ways to handle this problem: to complete the observation vector with zero deviation from the nominal trajectories (zero option: Z); complete the observation assuming that the mean centered and scaled current deviation from the nominal trajectory will be maintained throughout the rest of the batch (i.e. current deviations option: CD), or estimate the “missing” values using the PCA model (missing

data option: MD). The control limits of the resulting monitoring charts are dependent upon the used option (Nomikos and MacGregor, 1995b).

This “filling in” need of the method has been criticized by several authors (Cho and Kim, 2003; Meng, Morris, and Martin, 2003; Undey, Ertunc, and Cinar, 2003; van Sprang et al., 2002) without any convincing analysis to support their arguments. This has also led to alternative approaches to monitoring by unfolding the three way data array variable wise (i.e. as a  $IK \times J$  matrix), then performing PCA or PLS (Undey, Ertunc, and Cinar, 2003; Wold et al., 1998), and using the scores of this model for monitoring. The sole advantage of this approach is that it does not require any “filling in” mechanism for the remainder of the batch, because the scores only model the correlation among the variables at the local point in time and do not take into account the dynamic nature of the batch. Nomikos (1995) originally suggested this approach as an alternative, but dismissed it after a thorough investigation because it captures the wrong covariance structure in the data. Since unfolding in this direction followed by mean centering and scaling does not remove the mean trajectories from the data, this unfolding approach was shown to capture mainly the covariance among the mean trajectories of the variables, which is not the interest in monitoring. Rather, it is the covariance structure of the deviations of the variables about these mean trajectories that is of interest. This alternative unfolding approach also gives only a “steady-state” model that captures only the local covariance structure among the variables, and does not account for the dynamic behavior of the batch process (hence the reason for not requiring any estimate of the effect of disturbances on the future trajectories). Finally, it does not incorporate the time varying covariance structures that normally exist in batch processes (each component models an effect which is assumed constant over the entire course of the batch). Its deficiencies in modeling were found to greatly out-weight any advantage of not having to estimate the missing data for the remainder of the batch (Westerhuis, Kourti, and MacGregor, 1999). Some of the deficiencies of this alternative unfolding method for monitoring have been overcome to some extent by building multiple models for different stages of the batch,

within which the covariance structure of the variables is more constant, and by monitoring about the mean trajectories of the resulting scores (UMETRICS AB 2002). Another alternative which avoids the need to estimate future trajectories is the adaptive modeling approach of Rannar *et al.* (1998). As in the Nomikos and MacGregor (1994) approach it unfolds batch-wise and mean centers from the average trajectories. It accounts for the time varying dynamic behavior through the use of an adaptive PCA model.

In this work we demonstrate that using the missing data (MD) approaches to estimate the future unknown trajectories in a new batch is in fact a powerful model predictive monitoring scheme where the future trajectories are predicted with good accuracy even from the very beginning of the batch run. The missing data in the future trajectories are predicted much in the same way as an adaptive multivariate time series forecast, but with optimal use of all the available measurements up to current time and detailed knowledge about the time varying correlation structure among the variables throughout all the rest of the batch. This model predictive monitoring scheme gives much better estimates of the final overall batch scores and of the process variable trajectories for the remainder of the batch than the other options.

In this investigation the three different “filling in” methods (MD, Z, and CD) are contrasted with each other, and with the use of a multivariate auto-regressive (AR) time series. Furthermore, in the MD case, several methods for handling missing data samples are included in the comparison: single component projection (SCP), projection to the plane using PLS or ordinary least squares (PP<sub>PLS</sub>, PP<sub>OLS</sub>), conditional mean replacement (CMR), trimmed score regression (TSR), and iterative imputation (II) (Arteaga and Ferrer, 2002; Nelson, Taylor, and MacGregor, 1996). This last method has proven to be equivalent to the projection to the plane (Arteaga and Ferrer, 2002), but we include both methods because even though the analytical estimator is theoretically the same, the

computations and error propagation are different. (A nomenclature Section is included at the end of this chapter for the many acronyms used)

The comparisons are made from different perspectives: *i)* accuracy in the prediction of the unknown samples in the batch, *ii)* the quality of the score estimates made during the progress of the batch and *iii)* fault detection reliability of the resulting monitoring schemes.

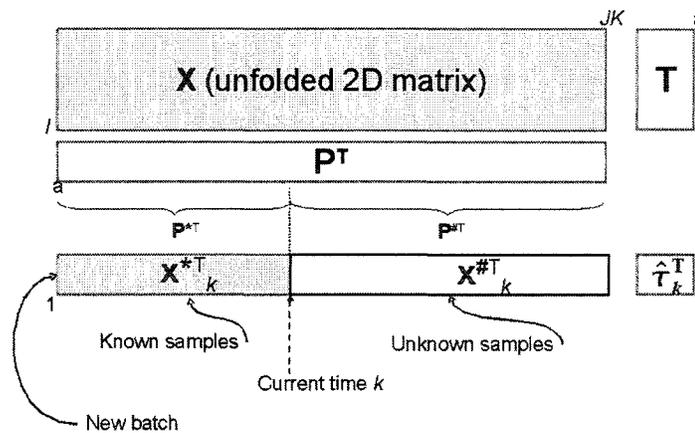
This work is organized as follows: in Section 3.2 the “filling in” methods are discussed for the missing data case and it is shown how the future samples are predicted with the PCA model. Section 3.3 compares the accuracy of these predictions with the ones obtained with the other options considered. In Section 3.4 the score estimates are discussed and compared as they evolve throughout the execution of the batch. Section 3.5 compares the ability of some of the approaches for monitoring the process and detecting various faults. Section 3.6 provides some remarks and a summary of our main conclusions.

## 3.2. Forecast and filling in mechanisms

### 3.2.1. Notation

In the following sections,  $a$  is used to represent the total number of components in a PCA model;  $\mathbf{X}$  represents the original data set used to build the model;  $\mathbf{T}$  represents the known scores from the original model;  $\mathbf{P}$  represents the loadings of the PCA model,  $\mathbf{x}$  is used to represent a vector of data from a new observation;  $\hat{\boldsymbol{\tau}}$  is used to represent the estimate of the score vector corresponding to  $\mathbf{x}$  in the presence of incomplete data and  $\boldsymbol{\tau}$  is the value of the score vector with the complete data vector. The dataset  $\mathbf{X}$  contains  $I$  batches, each batch has  $J$  variables sampled  $K$  times throughout the batch run.  $\mathbf{X} \in \mathcal{R}^{I \times JK}$ ,  $\mathbf{T} \in \mathcal{R}^{I \times a}$ ,  $\mathbf{P} \in \mathcal{R}^{JK \times a}$ ,  $\mathbf{x} \in \mathcal{R}^{JK \times 1}$ ,  $\boldsymbol{\tau} \in \mathcal{R}^{a \times 1}$ ,  $\hat{\boldsymbol{\tau}} \in \mathcal{R}^{a \times 1}$ . Lower case bold letters ( $\mathbf{x}$ ) denote a

vector; upper case bold letters ( $\mathbf{X}$ ) denote a matrix. For a certain observation  $\mathbf{x}$  with missing elements, we will group all the missing elements at the *end* of the observation, the known samples in the observation will be denoted by  $\mathbf{x}^*$ , and the missing samples will be denoted by  $\mathbf{x}^\#$ . So, for a certain observation  $\mathbf{x}$  with missing elements all at the end of the vector, we denote as  $\mathbf{x}^T = [\mathbf{x}^{*T} \ \mathbf{x}^{\#T}]$ . This notation is applied to other matrices or vectors as well, e.g.  $\mathbf{P}^*$  refers to those rows of a loading matrix  $\mathbf{P}$  that correspond to the known variables (elements) of  $\mathbf{x}$  ( $\mathbf{x}^*$ ); and  $\mathbf{P}^\#$  refers to those rows of  $\mathbf{P}$  corresponding to the missing elements of  $\mathbf{x}$  ( $\mathbf{x}^\#$ ). These definitions are illustrated in Figure 3.3.



**Figure 3.3** Notation and definitions

### 3.2.2. Data sets

Two data sets are used in this work: The industrial batch data set used by Nomikos and MacGregor (1995b) from an industrial nylon polymerization process; and a simulated set of batches for an emulsion polymerization process used by Nomikos and MacGregor (1994). The first data set will be referred to as the data set A, the second will be referred to as the data set B.

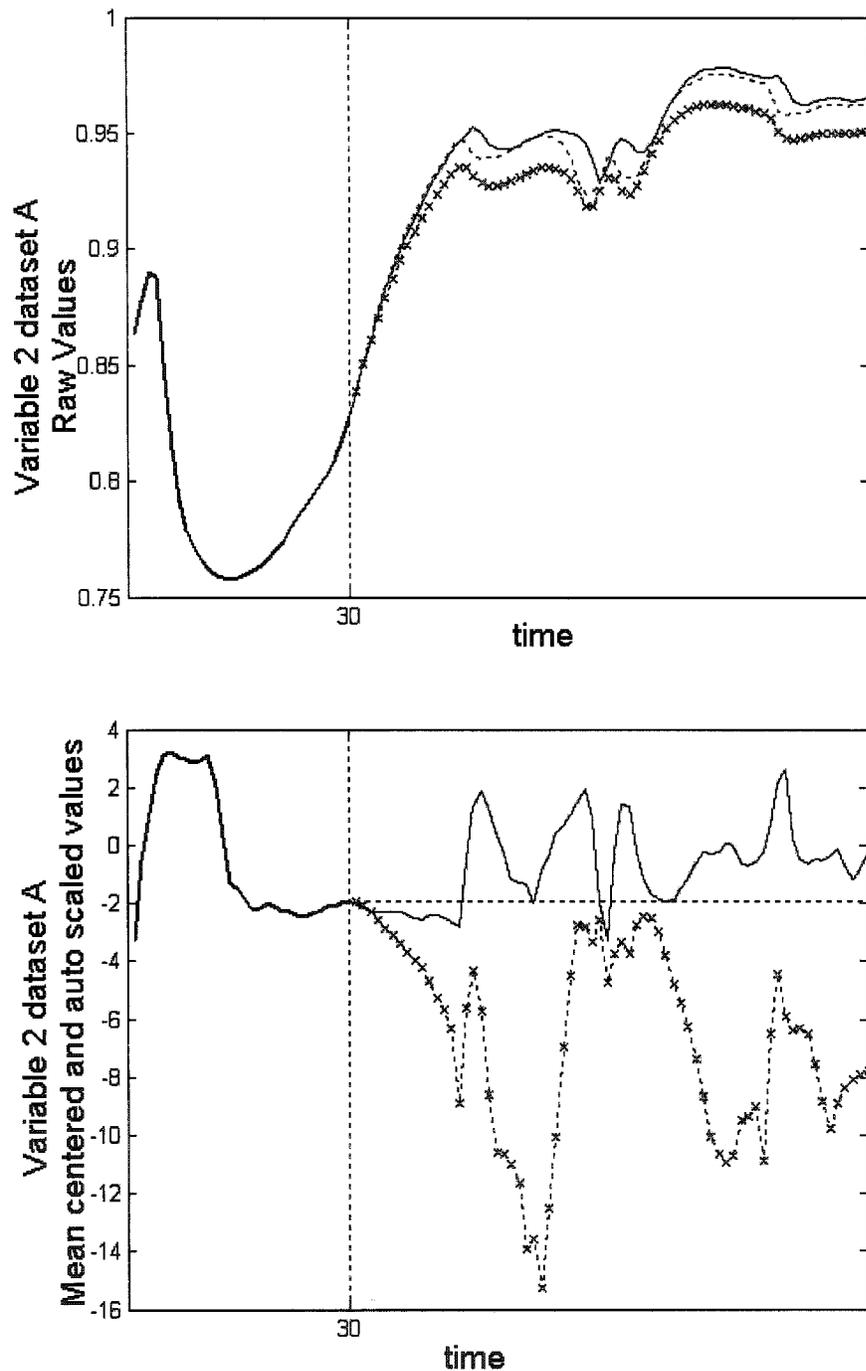
### 3.2.3. Future trajectory predictions and assumptions for batch monitoring

From the different options to “fill in” the future unknown observations; the *zeros* and *current deviation* options use an assumption about the future trajectory that is easy to visualize; and AR time series model approach is also easy to illustrate, but what various *missing data* options yield for the unknown future part of the trajectories, is not obvious. The following sections aim to clarify and illustrate this.

#### Zeros, current deviation and time series

Filling the rest of the trajectory with zeros assumes that the batch will follow the nominal or mean trajectory from sample  $k+1$  to sample  $K$ . *Current deviation* assumes that the deviation from the mean centered and scaled trajectories from sample  $k+1$  to sample  $K$  will be the same as the deviation at sample  $k$  (Fig. 3.4) in its mean centered and scaled value. The current deviation approach is similar to the approach taken in the Dynamic Matrix Controller (DMC) algorithm where the deviation from the set point in the future horizon ( $p$ ) is considered the same as in the present sample (Nomikos, 1995), doing so is equivalent to assume a random-walk disturbance in the future measurements (Kozub and MacGregor, 1992a).

A detail not mentioned in some previous works with CD approach (Meng, Morris, and Martin, 2003; van Sprang et al., 2002) is to clearly define which deviations are being kept constant: the mean centered and scaled (MCS) deviations (as in Nomikos and MacGregor (1995)) or the deviations in the raw unscaled variables (Cho and Kim, 2003). Figure 3.4 compares these two assumptions on variable #2 from dataset A. It will be shown that the latter approach ( $CD_{RAW}$ ) is not acceptable.



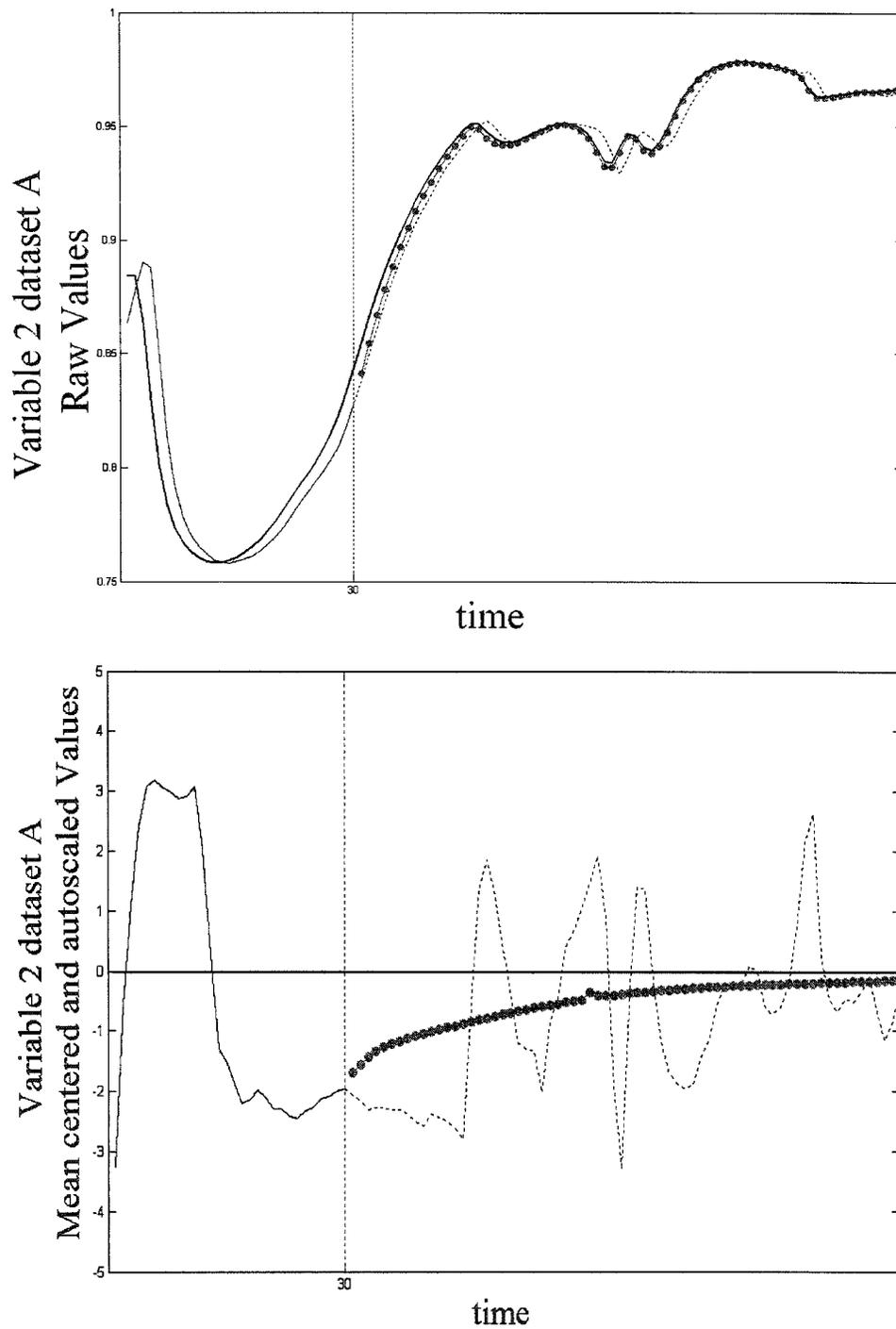
**Figure 3.4** Future trajectory prediction when using raw current deviations (x) and mean centered and scaled current deviations (dotted line), raw value scale (top) and MCS scale (bottom). Solid bold line: known measurements up to time 30; solid line: true unknown trajectory

To represent a batch trajectory with an AR multivariate time series we represent each batch as a set of vectors of measured variables at each time point  $k$  ( $\chi_k$ ) such that  $\mathbf{x}^T = [\chi_1^T, \chi_2^T \dots \chi_K^T]$ . At time  $k$ , the model will be able to make the forecast for the values of the samples at time  $k+1$  using Eq. 3.1 where  $q$  is the order of the model.

$$\hat{\chi}_{k+1} = \Phi_1 \chi_k + \Phi_2 \chi_{k-1} + \Phi_3 \chi_{k-2} + \dots + \Phi_q \chi_{k-q+1} \quad (3.1)$$

Several multivariate AR models can be built, one for each phase in the batch data (within which the autocovariance structure is reasonably constant). The order of each model is obtained by using the Akaike Information Criterion (AIC) (Akaike, 1974). The models are built using the mean-centered and auto scaled values of the batch trajectories in order to make the model comparable with the PCA forecast. Figure 3.5 illustrates the forecast made at time 30 for one variable in data set  $A$  using the corresponding AR model. Notice the forecast asymptotically returns to the mean, since an AR model is stationary in the mean.

The time series forecast for the unknown part of the trajectory is fairly simple to understand, and the replacement of the unknown samples in the batch is easy to visualize. In the following Section an adaptive AR time series analogy is used to illustrate the *missing data* estimation option of PCA/PLS models, in order to establish similarities with the time series forecast for better understanding of their properties.



**Figure 3.5** TS Forecast for the unknown samples. Solid line: known measurements up to time 30; solid bold line: mean trajectory; dashed line: true unknown trajectory; marked line (●): forecast by the time series

### Missing Data (MD) Options

For the previous *filling in* options, it is easy to illustrate the values that are being used to replace the unknown samples at time  $k$ ; however, in the MD approach, it is not obvious. In their earlier work, Nomikos and MacGregor (1995b) clearly state that, when using the MD approach, the missing unknown values will be *replaced* by predictions using the PCA model.

The simplest way to estimate the score vector when an observation contains missing data is to use the single component projection method (SCP) (Nelson, Taylor, and MacGregor, 1996). This method will calculate each of the scores independently and sequentially as

$$\hat{\tau}_{ki} = \mathbf{z}^{*T} \mathbf{p}_i^* / \mathbf{p}_i^{*T} \mathbf{p}_i^* \quad (3.2)$$

where  $\mathbf{z}^*$  is  $\mathbf{x}^*$  at time  $k$  deflated by the first  $i-1$  components. This method is the simplest but generally the poorest performing of the missing data handling methods (Nelson, Taylor, and MacGregor, 1996).

One natural improvement to the method is to use an initial estimate of the final scores such as given by SCP method (Eq. 3.2) to forecast the remainder of the trajectories,  $\hat{\mathbf{x}}_k^\#$ , (using their corresponding loadings) (Eq. 3.3)

$$\hat{\mathbf{x}}_k^\# = \mathbf{P}_k^\# \hat{\boldsymbol{\tau}}_k \quad (3.3)$$

create a new augmented vector using the known measurements and the new predicted values (Eq. 3.4); and recalculate the score estimate with this augmented vector (Eq. 3.5)

$$\mathbf{x}_{aug}^T = [\mathbf{x}^{*T} \hat{\mathbf{x}}^{\#T}] \quad (3.4)$$

$$\hat{\boldsymbol{\tau}}_k^T = \mathbf{x}_{aug}^T \mathbf{P} \quad (3.5)$$

This approach is taken by Meng *et al.* (2003), and Lennox *et al.* (2000) to compute the score estimates and the forecast for the unknown part of the trajectory in their respective studies. However, this is only the first step of the more complete *iterative imputation* (II) method proposed by Arteaga and Ferrer (2002), in this method, one iterates between equations (3.2) and (3.5) until convergence. This II algorithm is a very illustrative way of understanding how the missing data elements of the trajectories are being replaced by the PCA forecast.

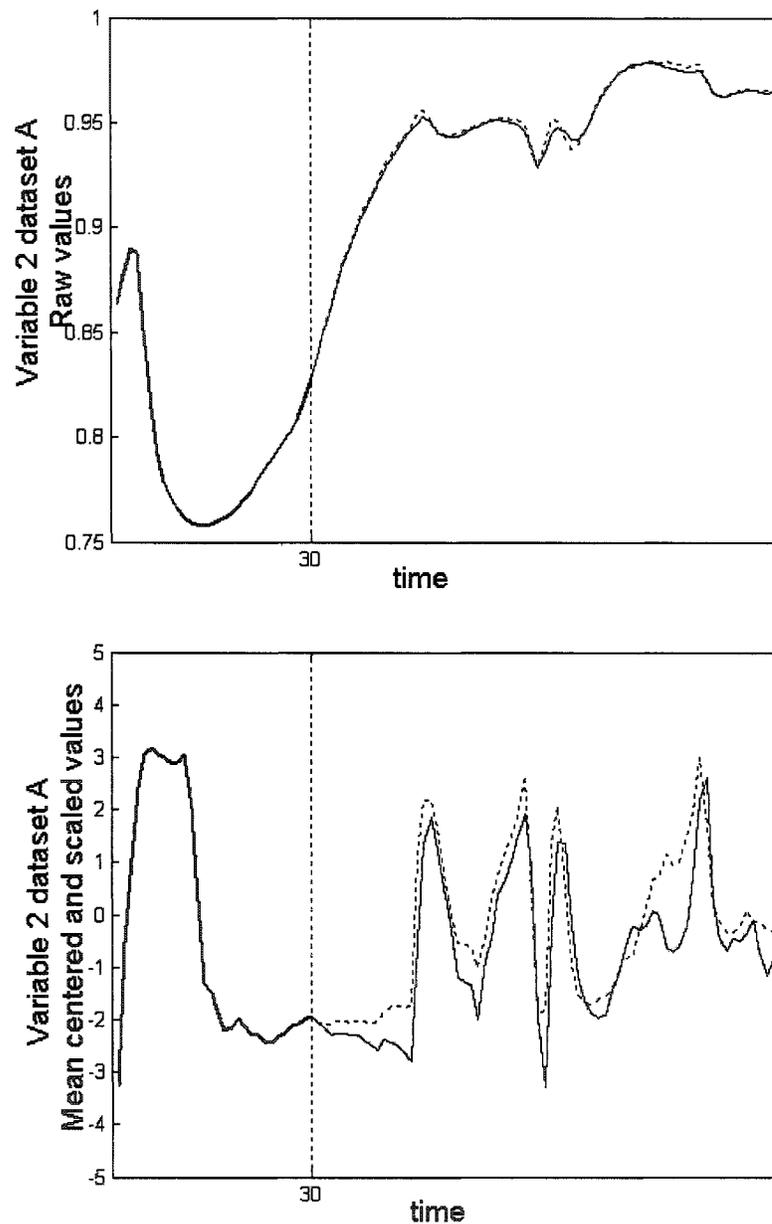
SCP computes each of the scores independently of each other, and Nelson *et al.* (1996) showed that superior results can be obtained by calculating all of the scores at once by projecting onto the hyper plane formed by the  $\mathbf{p}^*$  vectors. In the limit, when there are no missing measurements this will result in scores identical to SCP since the loadings vectors will be orthogonal. It is this loss of orthogonality as well as small values of the  $\mathbf{p}_i^{*T} \mathbf{p}^*$  (at the start of the batch) that causes the difficulties associated with estimating scores with missing data. Projection to the plane (PP) methods involve regressing the known  $\mathbf{x}^*$  vector onto the matrix defined by  $\mathbf{P}^*$ . Using OLS one gets  $\text{PP}_{\text{OLS}}$  (Eq. 3.6).

$$\hat{\boldsymbol{\tau}}^T = \mathbf{x}^{*T} \mathbf{P}^* (\mathbf{P}^{*T} \mathbf{P}^*)^{-1} \quad (3.6)$$

However, with certain combination of missing measurements (early in the batch) some of the columns of  $\mathbf{P}^*$  may become highly correlated and hence the  $(\mathbf{P}^{*T} \mathbf{P}^*)$  may be ill-conditioned. Therefore Nelson *et al.* (1996) recommend that the projection be done using PLS, PCR or regularized least squares regression ( $\text{PP}_{\text{PLS}}$ ,  $\text{PP}_{\text{PCR}}$ , etc). In the early

work by Nomikos and MacGregor (1995b), the missing data problem was treated with  $PP_{OLS}$ , which has been proven to be analytically equivalent to the iterative imputation method (Arteaga and Ferrer, 2002). Finally, other methods are also available to overcome ill-conditioning problems that may arise in the use of the PP method. *Conditional mean replacement* (CMR) (Nelson, Taylor, and MacGregor, 1996) and *trimmed score regression* (TSR) (Arteaga and Ferrer, 2002) are methods that use the known  $\mathbf{T}$  matrix from the training data in addition to the loadings ( $\mathbf{P}^*$ ) and the available measurements ( $\mathbf{x}^*$ ) to estimate the score vector. The theoretical background for these methods can be found in the work by Nelson *et al.* (1996) and Arteaga and Ferrer (2002). In solving the CMR problem, a singularity problem may arise. This problem is solvable by a workaround suggested in the work by Nelson *et al.* (1996), where  $\hat{\mathbf{t}}$  is calculated with the use of a new estimating parameter ( $\beta$ ) computed by using PLS to fit the equation  $\mathbf{T} = \mathbf{X}^* \beta$  where  $\mathbf{T}$  and  $\mathbf{X}^*$  respectively represent the score matrix and those columns from the training data set corresponding to the known values. The parameter matrix  $\beta$  along with the current available data vector  $\mathbf{x}^*$  are then used to compute  $\hat{\mathbf{t}}$ . PLS can be used to estimate the  $\beta$  parameter matrix in order to overcome any ill conditioning present in the  $\mathbf{X}^*$  matrix.

Figures 3.6 illustrate how the predicted values for the unknown part of the trajectory using  $PP_{PLS}$  compare to the actual values (compare with Fig. 3.4 to contrast with the “Current Deviations” forecast). Note the detailed precision of this forecast, which is made possible by the use of the known time varying covariance structure of the process variables over the entire batch history. In the following Section, a more complete comparison will be done in order to contrast the quality of the forecast for the unknown part of the trajectory, obtained by each of the considered methods.



**Figure 3.6 PCA Forecast for the unknown samples using  $PP_{PLS}$ . Solid bold line: known measurements up to time 30; solid line after time 30: true unknown trajectory; dashed line: forecast by the PCA model based with information up to time 30**

### 3.2.4. The PCA forecast model interpreted as a time series forecast

Since it is clear that the *MD* option uses a forecast of the trajectory to “fill in” the unknown measurements, it is desirable to analyze the properties of the forecast model embedded in the PCA projection. In the following paragraphs, the embedded one-step ahead time series prediction model is uncovered from the PCA prediction expression to illustrate the nature of the prediction model. However, equivalent expressions for any  $f$ -step ahead forecast could have been used to illustrate the same point.

Previously, the vector  $\chi_k$  was defined as a column vector containing the  $J$  batch variables measured at time  $k$ , such that:  $\mathbf{x}^T = [\chi_1^T, \chi_2^T \dots \chi_k^T]$ . Using the notation defined previously, we can express the known batch samples at time  $k$  in terms of  $\chi_k$  (Eq. 3.7):

$$\mathbf{x}_k^{*T} = [\chi_1^T \chi_2^T \dots \chi_k^T]. \quad (3.7)$$

From these equations, it is clear that the one-step-ahead prediction  $\hat{\chi}_{k+1}$  corresponds to the forecast of the first  $J$  elements of the  $\hat{\mathbf{x}}^\#$  vector (Eq. 3.8):

$$\hat{\chi}_{k+1} = \hat{\mathbf{x}}_{k[1:J]}^\#, \quad (3.8)$$

Eq. 3.8 can be expressed as a function of the loadings matrix (Eq. 3.9), from Eq. (3.3), and taking only the first  $J$  rows of the  $\mathbf{P}_k^\#$  matrix:

$$\hat{\chi}_{k+1} = \hat{\mathbf{x}}_{k[1:J]}^\# = \mathbf{P}_{k[1:J]}^\# \hat{\mathbf{t}}_k \quad (3.9)$$

At time  $k$ , and assuming that projection to the model plane (PP<sub>OLS</sub>) is used to solve the missing data problem, the estimate of the score is (Eq. 3.10),

$$\hat{\mathbf{t}}_k^T = \mathbf{x}_k^{*T} \mathbf{P}_k^* (\mathbf{P}_k^{*T} \mathbf{P}_k^*)^{-1} \quad (3.10)$$

Substituting Eq. 3.7 and Eq. 3.10 into Eq. 3.9:

$$\hat{\boldsymbol{\chi}}_{k+1} = \hat{\mathbf{x}}_{[1:J]}^\# = \mathbf{P}_{k[1:J]}^\# \left[ \mathbf{P}_k^* (\mathbf{P}_k^{*T} \mathbf{P}_k^*)^{-1} \right]^T \begin{bmatrix} \boldsymbol{\chi}_1 \\ \vdots \\ \boldsymbol{\chi}_{k-1} \\ \boldsymbol{\chi}_k \end{bmatrix} \quad (3.11)$$

The term  $\mathbf{P}_{k[1:J]}^\# \left[ \mathbf{P}_k^* (\mathbf{P}_k^{*T} \mathbf{P}_k^*)^{-1} \right]^T$  of Eq. 3.11 is a matrix of  $J$  rows and  $Jk$  columns. Re-expressing each element of this matrix as  $\phi_{i,j}$  and grouping blocks of  $J$  columns as new square matrices  $\Phi_i$ , such that each  $\Phi_i$  has  $J$  rows and  $J$  columns gives rise to a multivariate AR prediction equation of order  $k$  with parameter matrices  $(\Phi_1, \Phi_2, \Phi_3, \dots, \Phi_k)$  (Eq. 3.12),

$$\begin{aligned} \hat{\boldsymbol{\chi}}_{k+1} &= \begin{bmatrix} \left[ \begin{matrix} \phi_{1,1} & \phi_{1,2} & \cdots & \phi_{1,J} \end{matrix} \right] & \left[ \begin{matrix} \phi_{1,J+1} & \phi_{1,J+2} & \cdots & \phi_{1,2J} \end{matrix} \right] & \cdots & \phi_{1,Jk} \\ \left[ \begin{matrix} \phi_{2,1} & \phi_{2,2} & \cdots & \phi_{2,J} \end{matrix} \right] & \left[ \begin{matrix} \phi_{2,J+1} & \phi_{2,J+2} & \cdots & \phi_{2,2J} \end{matrix} \right] & \cdots & \phi_{2,Jk} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \left[ \begin{matrix} \phi_{J,1} & \phi_{J,2} & \cdots & \phi_{J,J} \end{matrix} \right] & \left[ \begin{matrix} \phi_{J,J+1} & \phi_{J,J+2} & \cdots & \phi_{J,2J} \end{matrix} \right] & \cdots & \phi_{J,Jk} \end{bmatrix} \begin{bmatrix} \boldsymbol{\chi}_1 \\ \vdots \\ \boldsymbol{\chi}_{k-1} \\ \boldsymbol{\chi}_k \end{bmatrix} \\ &\quad \Phi_1 \qquad \qquad \Phi_2 \qquad \cdots \qquad \Phi_k \end{aligned} \\ &= \left[ \Phi_1 \Phi_2 \cdots \Phi_k \right] \begin{bmatrix} \boldsymbol{\chi}_1 \\ \vdots \\ \boldsymbol{\chi}_{k-1} \\ \boldsymbol{\chi}_k \end{bmatrix} = \Phi_1 \boldsymbol{\chi}_1 + \Phi_2 \boldsymbol{\chi}_2 + \cdots + \Phi_{k-1} \boldsymbol{\chi}_{k-1} + \Phi_k \boldsymbol{\chi}_k \quad (3.12) \end{aligned}$$

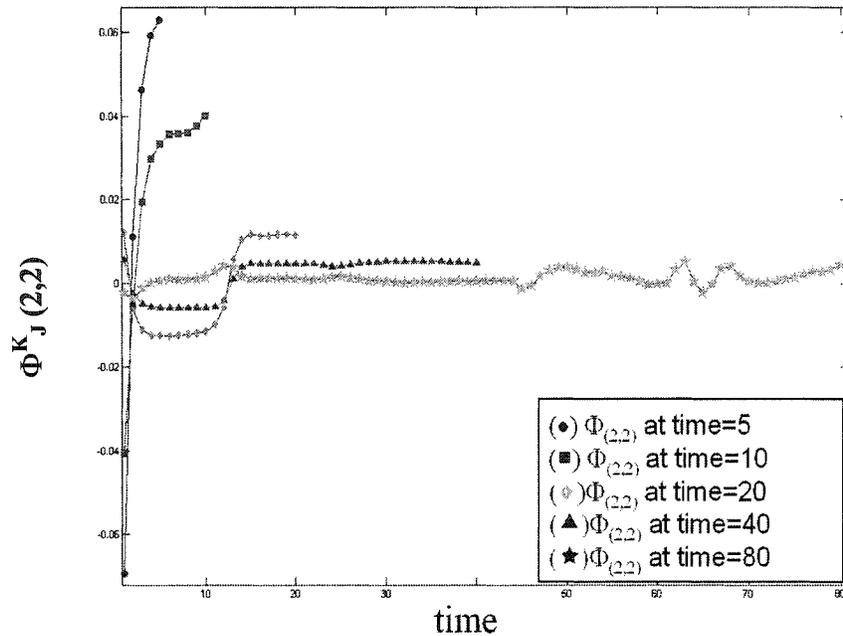
This one step ahead PCA forecast model has unique characteristics: a) the order of the PCA forecast equation (Eq 3.12) expands, as new measurements are available ( $k$  increases); b) even more importantly the parameters of this model will adapt as the batch evolves since the elements in  $\mathbf{P}^\#$  and  $\mathbf{P}^*$  change as more samples become available. This

means that all the  $\phi$  elements in Eq. 3.12 are adapted as time progresses to account for the changing auto-covariance and cross-covariance structure captured by the PCA model, this is possible because the MPCA batch model has embedded within it knowledge of the time varying covariance among all the variables over the entire time history of the batch. This contrasts with the conventional time series model (Eq. 3.1) whose coefficients do not change and will always use the same number of past measurements in order to forecast the next sample. As a result, the PCA prediction equation is actually an adaptive, nonlinear prediction and hence provides a much more powerful prediction than any fixed multivariate time series model. The same form of adaptive PCA time series prediction equation can be derived for any  $f$ -step ahead prediction with the same conclusions as above.

To further illustrate the adaptive nature of the PCA forecast model, consider the explicit forecast equation for the one step ahead prediction at different times (Eq. 3.13)

$$\begin{aligned}
 \text{At time =1} \quad \hat{\chi}_2 &= \Phi_1^1 \chi_1 \\
 \text{At time =9} \quad \hat{\chi}_{10} &= \Phi_1^9 \chi_1 + \Phi_2^9 \chi_2 + \Phi_3^9 \chi_3 \cdots + \Phi_9^9 \chi_9 \\
 \text{At time=99} \quad \hat{\chi}_{100} &= \Phi_1^{99} \chi_1 + \Phi_2^{99} \chi_2 + \Phi_3^{99} \chi_3 \cdots + \Phi_{99}^{99} \chi_{99}
 \end{aligned} \tag{3.13}$$

Figure 3.7 shows the values of the  $\Phi_j^k(2,2)$  parameter: for  $j= 5, 10, 20, 40, 80$ ; and  $k=j, j-1, j-2, \dots, 1$ ; for data set  $A$ . As seen in this Figure, the parameters adapt and expand as more measurements are available. A result of the adaptive nature of this model is that the behavior of the forecast, for the remaining unknown samples, changes with time in the batch. This adaptive behavior of the PCA predictive model is unique and results from the fact that the MPCA model built for a training set of complete batches has knowledge of how the auto and cross-covariance structure of all the variables will change over the entire duration of the batch.



**Figure 3.7** Coefficient  $\Phi_j^k(2,2)$  as it adapts and expands with time ( $k$ )

Each of the different options discussed to handle the future unknown samples will provide a different estimate of the future unknown trajectory. In the following Section, metrics are suggested to compare the accuracy of these predictions.

### 3.3. Statistics for Comparing Future Predictions

When monitoring a new batch, an estimate of the final score vector  $\hat{\tau}$  is done every time a new sample is available. Each missing data handling method will give a different estimate of the score vector and hence, a different forecast of the unknown trajectory every time a new measurement is available. In order to measure the accuracy of the forecast for the unknown part of the trajectory for each method, the errors in the prediction will be quantified and compared for all the batches in data set A and B.

For batch  $i$  at time  $k$  (given that  $k < K$ ), the forecast for each variable  $j$  generates a vector of errors in the future prediction  $\mathbf{e}_k^{ij} = [e_k^{ij}|_1 \ e_k^{ij}|_2 \ e_k^{ij}|_3 \ e_k^{ij}|_4 \ e_k^{ij}|_5 \ \dots \ e_k^{ij}|_{K-k}]$ , defined by Eq. 3.14, and illustrated in Figure 3.8. This  $(K-k)$  vector contains the individual errors in the prediction of the future unknown trajectory of variable  $j$ , for samples  $k+1$ ,  $k+2$ ,  $k+3$ , ...,  $K$ .

$$\mathbf{e}_k^{ij} = \mathbf{x}_{ijk}^\# - \hat{\mathbf{x}}_{ijk}^\# \quad (3.14)$$

This  $\mathbf{e}_k^{ij}$  vector will decrease its length as the batch evolves, because there are fewer remaining unknown measurements ( $\mathbf{x}^\#$ ).

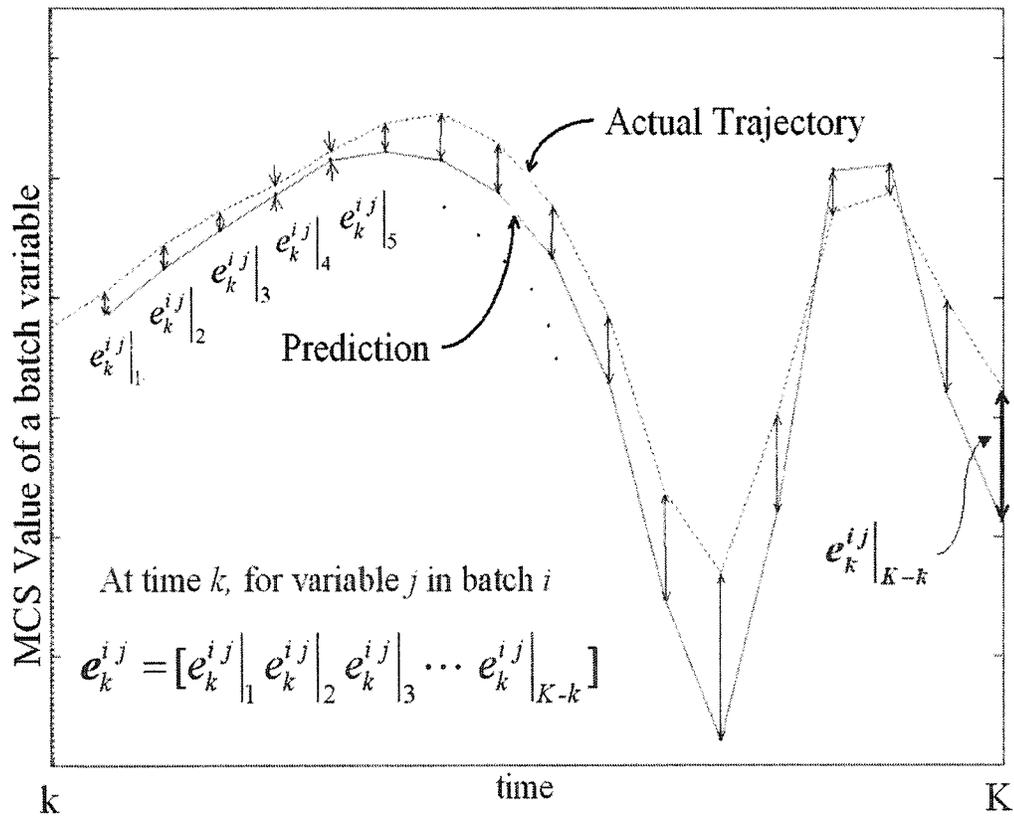


Figure 3.8 Future prediction error in batch trajectory forecasting

The original training batch data consists of  $I$  batches, each one with  $J$  variables, sampled  $K$  times throughout the batch. For each  $ijk$  sample in the original data set, we have a  $e_k^{i,j}$  vector. This set of error vectors can be computed for each option available to treat the unknown part of the trajectory.

To compare all the error vectors among options, it is desirable to summarize the  $e_k^{ij}$  vector by one summary statistic in order to quantify the future prediction error as a scalar measurement of accuracy. This accuracy will vary in time and therefore can be plotted as a trajectory for each variable in each batch. To accomplish this, two different measures are proposed:

a) Consider all individual errors ( $[e_k^{i,j}|_1, e_k^{i,j}|_2, e_k^{i,j}|_3, \dots, e_k^{i,j}|_{K-k}]$ ) equally important and simply square them, and add them to give one number, the *Future Prediction Sum of Squares* (FPRESS) defined by Eq. 3.15

$$FPRESS_k^{ij} = \sum_{l=1}^{K-k} \left( e_k^{ij}|_l \right)^2 \quad (3.15)$$

b) A second option is to weight each individual error ( $e_k^{ij}|_l$ ) by the inverse of the distance to the current time sample, in such way that the one step ahead prediction  $e_k^{ij}|_1$  becomes more important than the (k-K) step ahead prediction  $e_k^{ij}|_{K-k}$ ; the weighted *Future Prediction Mean Square Error* (FPMSE) is defined in Eq. 3.16

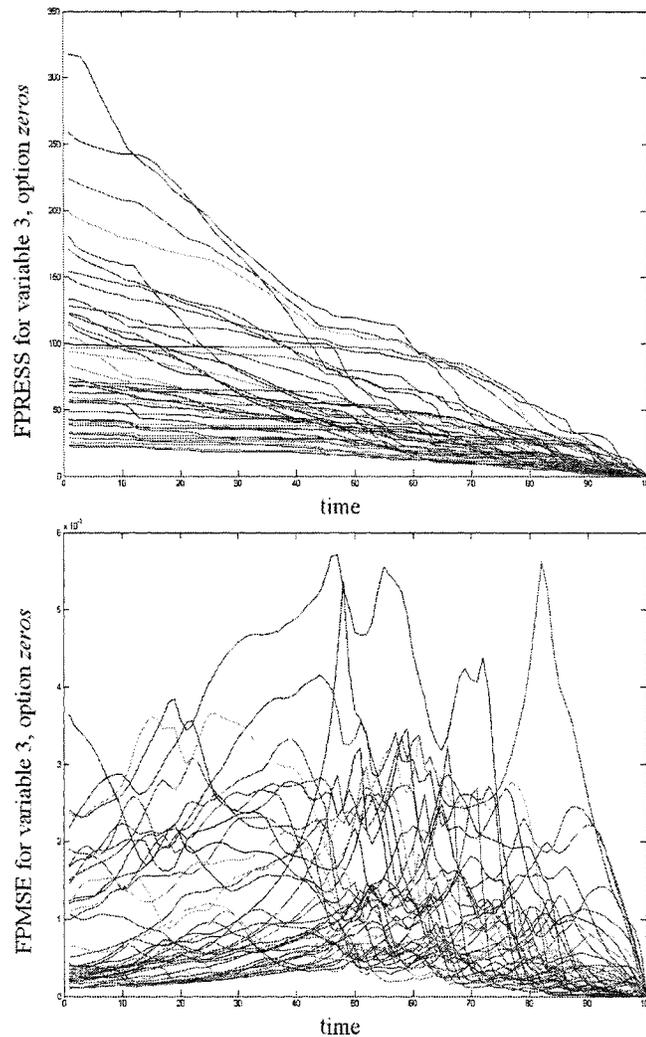
$$FPMSE_k^{ij} = \frac{\sum_{l=1}^{K-k} \frac{1}{l} \left( e_k^{ij}|_l \right)^2}{\sum_{l=1}^{K-k} \frac{1}{l}} \quad (3.16)$$

The *FPRESS* will provide a measure of global forecast accuracy, since it accounts for the error from time  $k+1$  to time  $K$ , on the other hand, the *FPMSE* provides a measure of local forecast accuracy, considering the immediate forecast more important than the long term forecast.

Figure 3.9 illustrates the shape of the *FPRESS* and *FPMSE* as they change with the evolution of the batch; the plot shows both calculations for all batches in the industrial data set A, using variable 3 and considering zeros ( $Z$ ) as the *filling in* option.

The calculation of these quantities should be done using a new set of NOC batches; in this case however, all the available observations are used to build the model and will be used to calculate the *FPRESS*, and *FPMSE*. The calculation sequence is similar to that used in computing the confidence intervals for the time varying scores (Nomikos and MacGregor, 1995b); the batches in the NOC data base are *monitored* with the model replacing at every time sample the unknown measurements with those of the considered option; then the scores are estimated; and in the case of MD, the forecast of the unknown part of the trajectory for each variable is calculated using these scores estimates. With this forecast the *FPRESS* and *FPMSE* are then calculated.

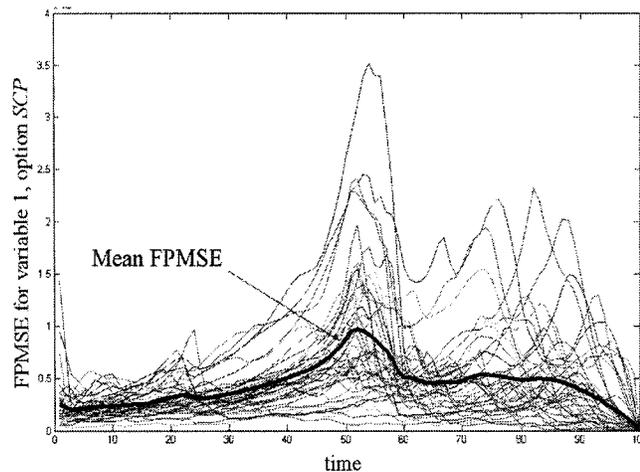
The ten options considered in handling the unknown samples of the trajectory are: zeros ( $Z$ ), mean centered and scaled current deviation ( $CD$ ), raw current deviation ( $CD_{raw}$ ), a fixed AR time series (TS) based on an average fit to the complete batch data; and different score estimation methods for the missing data option: SCP,  $PP_{ols}$ ,  $PP_{pls}$ , II, CMR and TSR (the missing data option will be referred to by the initials of the method used to estimate the score).



**Figure 3.9** FPRESS and FPMSE for variable 3 in data set A, using *zeros* option

Because *FPRESS* and *FPMSE* can be computed for all variables at all time samples, for each batch in the data sets, they can also be visualized as accuracy trajectories, and as such they can be compared. In order to make a comparison between the *FPRESS* trajectories for the 10 considered options (as well as for the *FPMSE*), we calculate an “average trajectory” for each of these new measures, for each variable over the whole data set. This will give us a better view of the behavior of each trajectory. This is illustrated in Fig. 3.10 where the plot shows all the individual *FPMSE* trajectories for

variable 1 (for all batches in dataset A) and the mean of these new trajectories using SCP option for the score estimation.



**Figure 3.10 Mean FPMSE for variable 1 in data set A, using missing data and solving with SCP**

In order to perform the comparison exercise, the *FPRESS* and *FPMSE* mean trajectories are plotted per variable, for each of the 10 different methods. A brief discussion of the results follows; only some representative results are included due to the large number of graphs necessary to include all the results.

Figure 3.11 shows the *FPMSE* and the *FPRESS* for variable 4 in data set *A*. It is very clear how, from all the methods, the MD options have the least error in the forecast at every time period over the whole trajectory, even from the very initial samples in the batch. This behavior was found to be true in general for all the variables for this data set for both of the two different future error trajectory statistics (*FPRESS* and *FPMSE*), although the CD and TS options on limited occasions showed slightly lower values at a few time intervals.

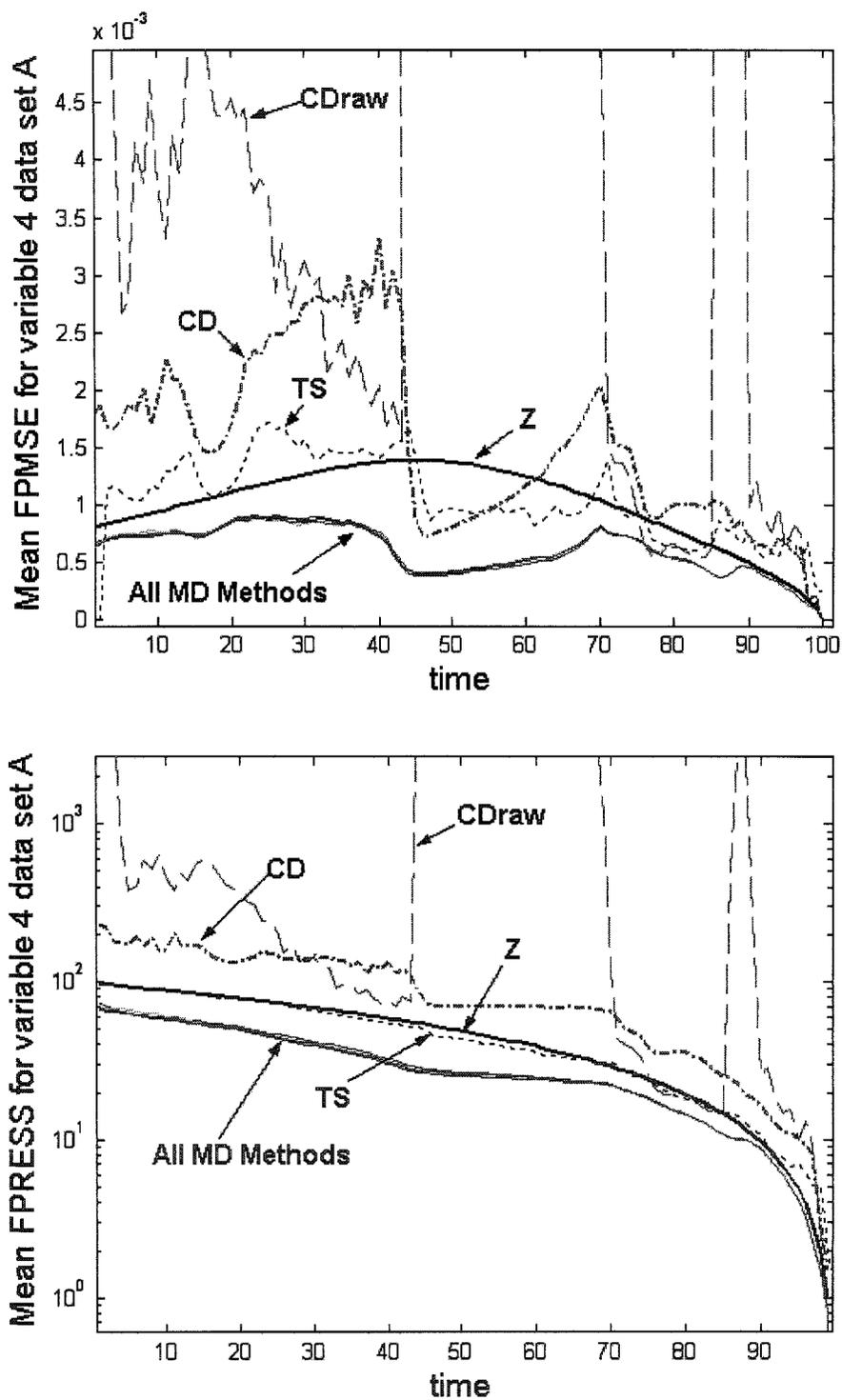


Figure 3.11 Mean FPMSE (top) and FPRESS (bottom) for variable 4 in data set A

Data set B was simulated from a theoretical model for styrene-butadiene emulsion polymerization (Nomikos, 1995). As a result, one can expect some artificial behavior. In this case it arises because deviations from the mean trajectories for the monomer feed rates were simulated as a time-invariant stationary AR time series (and hence the AR behavior is passed to other variables as well). Because of this prior knowledge about the data set, the expected result is that the prediction of the *TS* (and *Zeros*) will be the best forecasts. This is a very important benchmark to use in the comparison exercise.

Figure 3.12 show the *FPRESS* and *FPMSE* for variable 1 in dataset B. The results in these Figures are very important since it is known that the AR time series model should give the best possible forecast for this data set. However the MD options, using CMR and TSR, give predictions as good as the ones from TS and Z, showing the power of the PCA model in capturing the dynamic relationships among the variables in the dataset. Aside from some early inaccuracies, the PP methods also provide good predictions. These results are similar for the other variables in data set B. Together with the results from data set A, these findings illustrate the power of using the PCA model to predict the future unknown samples in the trajectory even from the very beginning of the batch, especially when one of the more efficient missing data inference methods is used.

In their early work, Nomikos and MacGregor (1994;1995b) used the  $PP_{OLS}$  approach which can sometimes be poor at the beginning of the batch as shown in Fig. 3.12. Another expected result from this exercise is that; for most of the variables analyzed, the CD option (based on scaled mean centered trajectories) out-performed by far the  $CD_{raw}$  option in the forecast of the future unknown samples.

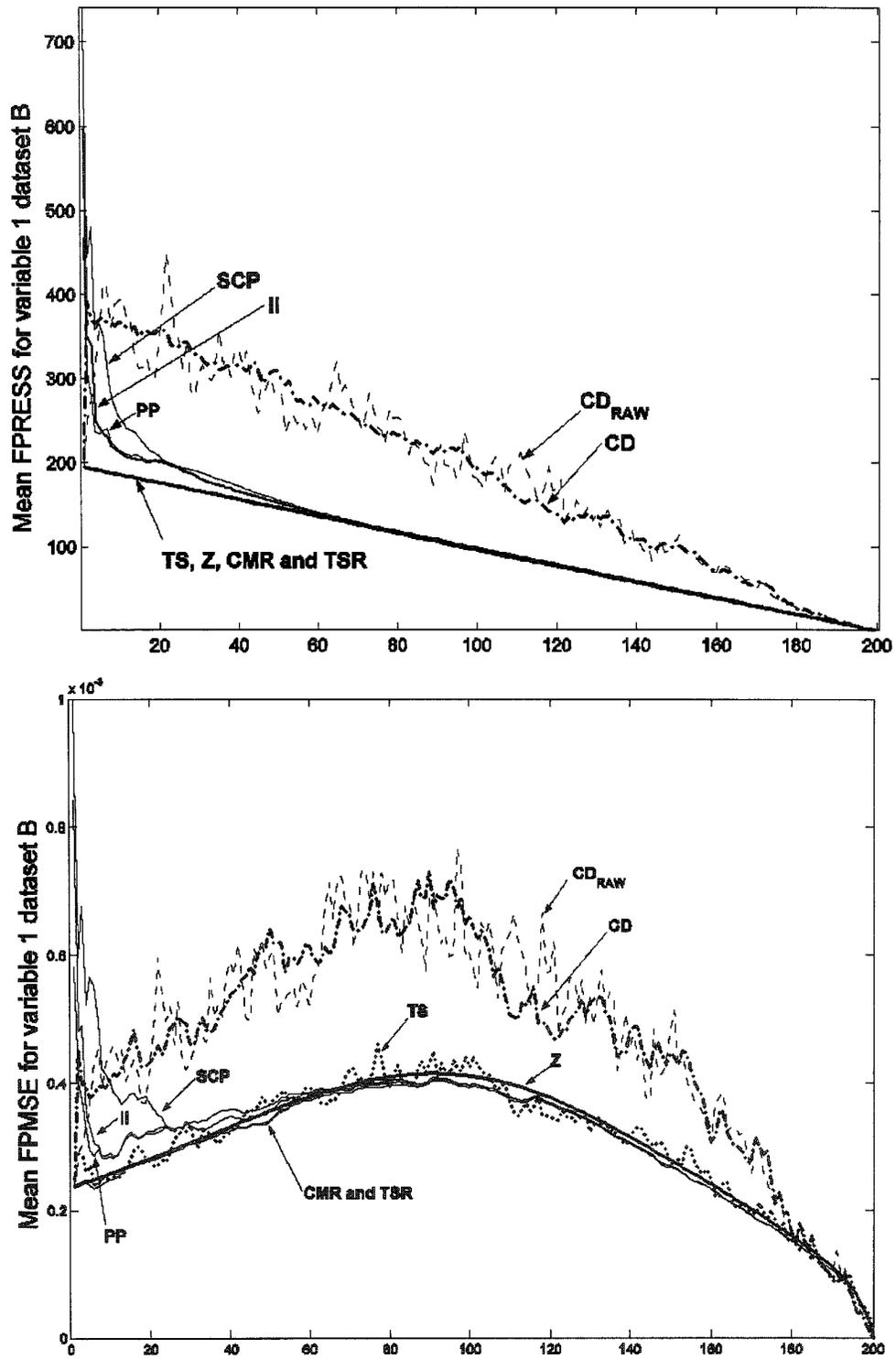
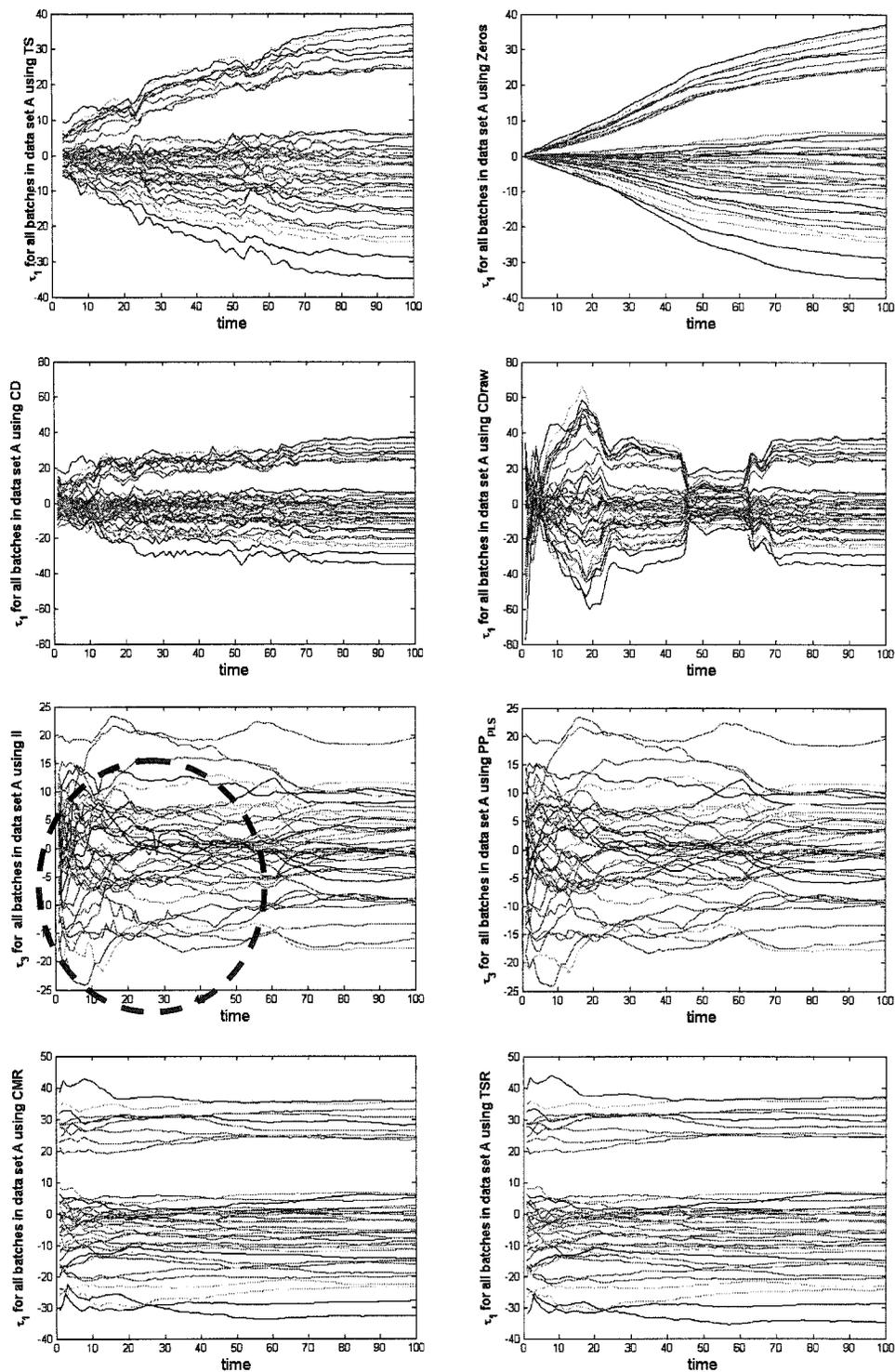


Figure 3.12 Mean FPRESS (top) and mean FPMSE (bottom) for variable 1 in dataset B

### 3.4. Quality of the score estimates

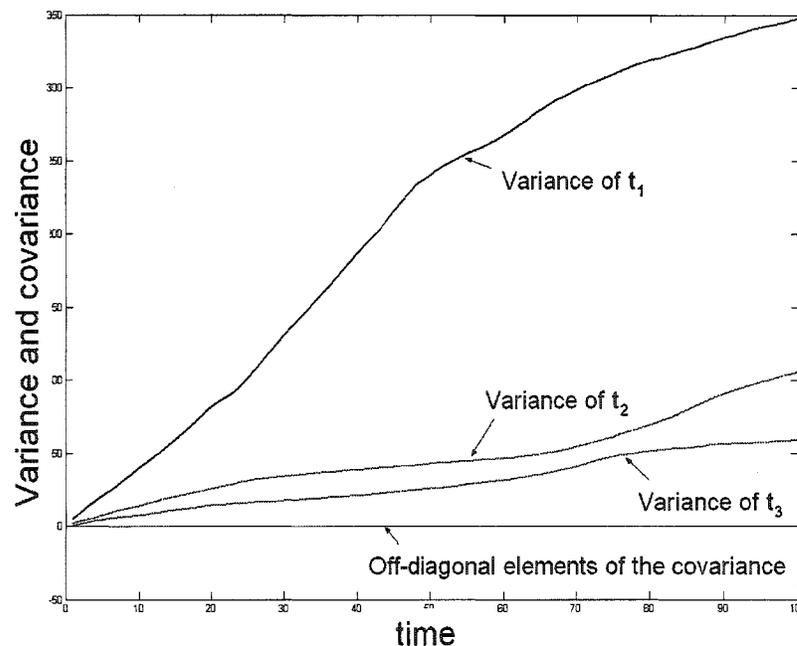
All the forecasts compared in this last Section are a function of the score estimate ( $\hat{\tau}$ ) for the end of the batch. In this Section a comparison exercise is done among the different  $\hat{\tau}$  estimates given by each of the 10 options considered (Z, CD, MD and TS) and methods for score estimation (SCP, PP<sub>PLS</sub>, PP<sub>OLS</sub>, II, CMR and TSR).

The estimates of the final batch scores, for a new batch are now contrasted for the different methods using the data set A (being of industrial nature). Nomikos and MacGregor (1995b) show how the trajectories of the estimated scores look like for the assumptions of Z, CD and MD using PP<sub>OLS</sub>. The results obtained with the additional methods considered here are not far from those previously shown by those authors. The score plots (Fig. 3.13) obtained using a fixed stationary AR time series are similar to those obtained by the Z option but starting from a different point than zero and with a much noisier evolution. It is proper to point out that applying the concept behind the Z option for any missing data problem (that is, replacing the missing values with the mean ones) is perfectly equivalent to the so-called *trimmed score method* proposed by Arteaga and Ferrer (2002). The score lines obtained using the CD option is much more stable and have a lower variance (Fig. 3.13) throughout the batch than those obtained with the CD<sub>raw</sub> option. All the score estimates using different MD methods behave in a similar way, they are all relatively horizontal lines, but with different behavior at the beginning of the batch (this will be brought up in later Section). The estimates obtained when using the II sometimes show slightly noisy score trajectories (in dashed circle in Fig. 3.13), compared to other methods like PP<sub>PLS</sub> this noisy behavior appears especially in the later components in the model (Fig. 3.13), finally, CMR score estimates were found to be very similar (almost identical) to those obtained with TSR (Fig. 3.13).



**Figure 3.13** Monitoring score plots for different “filling in” options and missing data handling methods for dataset A

In their early work, Nomikos (1995) and Nomikos and MacGregor (1995b), acknowledge that the most rigorous solution to the monitoring process is to build as many models as time samples in the data set, and use the corresponding model at each time interval to calculate the scores (and thereby completely avoid the need to *fill in*). Score plots computed by this approach show several very important features: the covariance matrix calculated from the models built at each time sample always has two characteristics throughout the batch evolution (Fig. 3.14): *i*) the scores are always orthogonal, hence the off-diagonal terms of the score covariance matrix are **zero**; this property will be referred to as *orthogonality*; and *ii*) the variance of  $t_1$  is greater than the variance of  $t_2$ , and so on for the entire batch, this property will be referred to as *coherence*. These results are exactly as expected, since each of the scores at each point in time is calculated from an independent PCA model. Hopefully, a single PCA model with a good missing data algorithm should provide accurate score estimates having orthogonality and coherence properties close to this ideal situation.

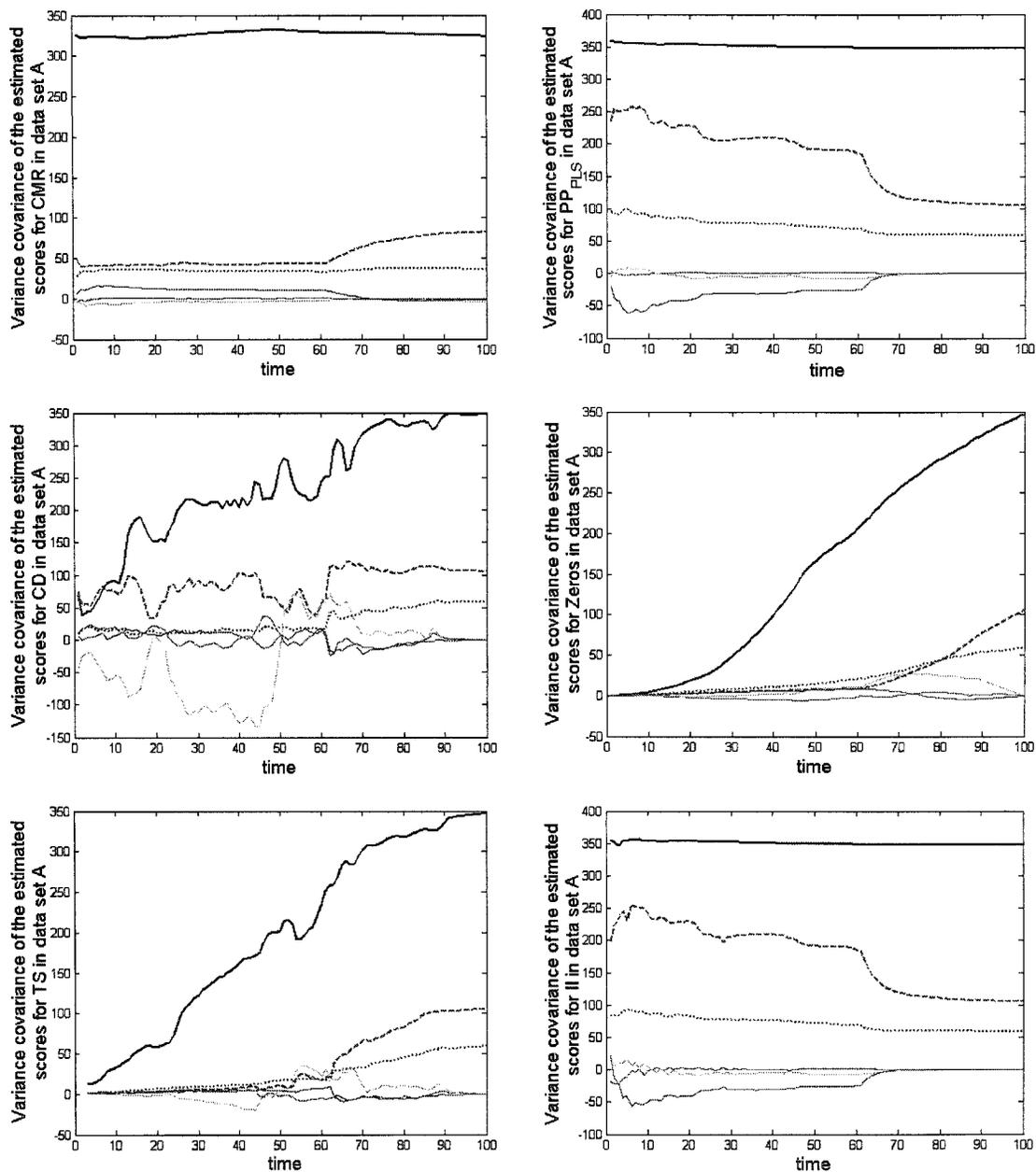


**Figure 3.14** Variance and covariance of the scores from  $K$  models throughout the batch, dataset A

The different approaches to estimating the batch scores ( $\hat{\tau}$ ) are now compared at different time periods ( $k$ ) in terms of these characteristics computed from the estimated score variance-covariance matrix. In analyzing the covariance of the scores for the 10 methods considered, for both datasets, it was observed that the only approaches that not only ensure orthogonal and coherent behavior in the score estimates, but also offer a stable and accurate value of the covariance throughout the batch time, even at the very beginning of the batch, are the missing data approaches solved with *CMR*, *TSR* and *PP<sub>PLS</sub>*. These three methods for handling missing data have already been shown to have superior performance in the estimation of the score vector (Arteaga and Ferrer, 2002; Nelson, Taylor, and MacGregor, 1996). For the *CMR*, *TSR* and *PP<sub>PLS</sub>* option the off-diagonal elements of the covariance trajectories are kept close to zero throughout the batch (*orthogonality*). This does not happen entirely when using *SCP*, where the *orthogonality* is lost during some periods of time.

The non-missing data options for the *filling in* mechanism generally fail in satisfying any of the two desired properties (*orthogonality* and *coherence*). Fig. 3.15 illustrates how the *CD* and *Z* options lose orthogonality as well as the coherence. Finally, the covariance behavior of the *CD* always showed better properties than the *CD<sub>raw</sub>*; maintaining better orthogonality and coherence. Using the *MD* option, and solving with *PP<sub>OLS</sub>* or *II* gave good properties in the covariance trajectories for data set *A*, but not as good for data set *B*. *CMR* and *TSR* appear to give the best score estimates, with the best properties for both data sets.

A final very important property is the *stability* of the score estimates during the first samples in the batch. When the *MD* approach is solved via *SCP* (and to a less extent when solving by *PP<sub>PLS</sub>*, *PP<sub>OLS</sub>*, and *II*) highly variable and un-reliable estimates of the score can sometimes be obtained during the first samples. However, if the missing data approach is solved with *CMR* and *TSR*, this problem disappears. Fig. 3.16 is a plot of the variances for  $t_1$ , for the 10 considered methods for dataset *B*. Of course, the *Z*, and the *TS*



**Figure 3.15** Time evolving variance-covariance of the scores for CMR, PP<sub>PLS</sub>, CD Zeros, TS and II for dataset A  
 var t<sub>1</sub>(—), var t<sub>2</sub>(- -), var t<sub>3</sub>(...), covariances (—)

options give the lowest variance at the beginning of the batch (because they assume all the rest of the measurements are at its mean value); however these two options have shown not to fulfill *orthogonality* and *coherence* and their estimate of the final score is unreliable. The next lowest variances at the beginning of the batch are given by the CMR and TSR methods for the MD option; these methods have shown to keep *orthogonality* and *coherence* during the batch run **and** have proven to give accurate estimates of the final score, the rest of the methods for handling missing data give a high variability of the  $t_1$  estimate at the beginning of the batch.

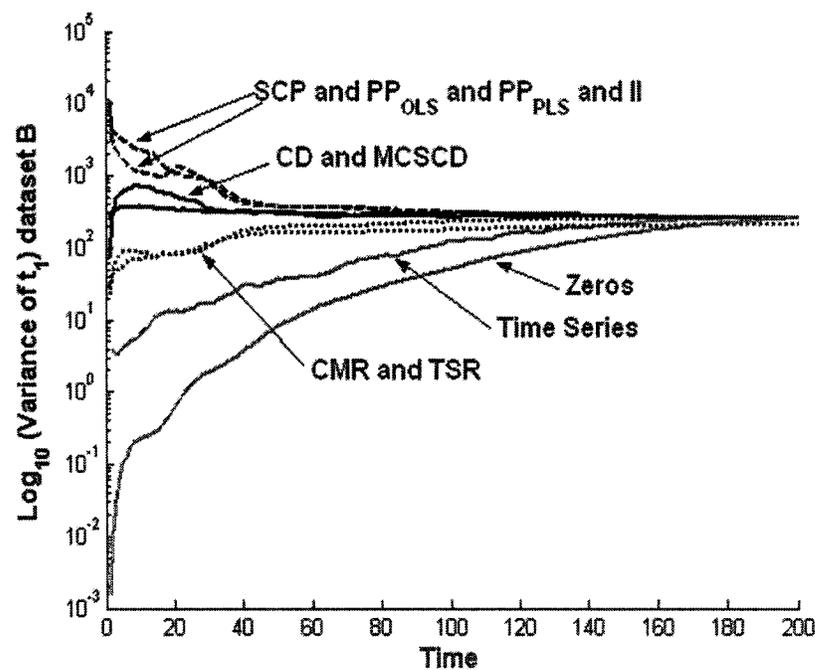


Figure 3.16 Evolving variance of  $t_1$  for dataset B, using all the methods and options

From all the above results, it can be concluded that the best score estimates, and the best forecast for the unknown samples, are the ones obtained when the PCA model is fully used (MD option) with CMR and TSR to estimate the scores, even from the beginning of the batch run.

### 3.5. Monitoring Performance

As shown in previous sections, filling in by the MD options gives much better scores and process trajectory predictions for the remainder of the batch. The improved qualities of these trajectory predictions are important in process control applications (Flores-Cerrillo and MacGregor, 2002;2003) where manipulated variables are being used to control final product quality based on these trajectory predictions. However, in process monitoring for faults, it is not obvious that better predictions will necessarily yield better monitoring. The purpose of this Section is to investigate the impact of the prediction or filing-in method on the fault detection and isolation properties of monitoring schemes.

A critical desirable feature in any monitoring scheme is to detect an abnormality in the batch evolution, and trigger the proper alarm as early as possible. The faulty batches included in the data sets A and B had several faults that were easy to detect by any of the methods. Because of this, a new dataset was created using a detailed simulation of the emulsion SBR polymerization process (Kozub and MacGregor, 1992b) (the original source of dataset B), and new faulty batches with more subtle deviations were simulated. The new dataset consists of 200 NOC batches; each batch has 9 variables (total volume, conversion, accumulated composition, styrene feed, butadiene feed, chain transfer agent (CTA) feed, reactor temperature, jacket temperature, and density) sampled 350 times throughout the batch run. Each one of the 200 NOC batches is simulated using a second order AR stochastic disturbances in the input flow of the styrene and CTA ( $\sigma_a^2=0.0005$ ,  $\phi_1= 0.98$ , and  $\phi_2= -0.03$  for styrene feed; and  $\sigma_a^2 = 1.25 \times 10^{-6}$ ,  $\phi_1= 0.8$  and  $\phi_2= 0.17$  for CTA feed) and white noise disturbances in the rest of the variables with standard deviations given in Table 3.1. This data set is divided as follows: 100 batches are used as training, and the rest as validation data. A PCA model with 5 components is built, this model captures 72.4% of the total variance for the training set.

Variable	Units	$\sigma(\text{noise})$
Temp.	Celsius	0.1
Conv.	-	0.001
Dens.	g/l	0.25
Composition	% Styrene	0.0003
Volume	L	$3.91 \times 10^{-6}$

**Table 3.1 Standard deviation of the disturbances added to the simulation results**

A new set of 170 faulty batches is created: 10 repeats for faults 1 and 2, and 50 repeats for faults 3, 4 and 5. The first fault is a thermocouple failure that occurs early in the batch (time=21) when the temperature happens to be above the set-point; the thermocouple measuring the reactor temperature breaks and keeps its zero-order-hold value (last good value recorded). The PID controller for the temperature in the reactor tries to cool the reactor and the cooling medium temperature is forced to its lower bound causing the reaction to die. Fault number two is a thermocouple failure as well, but in the thermocouple measuring the heating medium temperature and the fault occurs late in the batch evolution (time =290) just after the flow of styrene is interrupted. Fault number three is a very small bias ( $2 \times 10^{-4}$  mol/min) in the feed of butadiene from early in the batch (time=1). Fault number four is a ramp in the flow of butadiene (with a final magnitude of  $5 \times 10^{-4}$  mol/min) starting at time 151. Finally, fault number five is an increase of 20% in the noise variance for the styrene feed flowmeter (time=51).

Monitoring charts (D and SPE) were then established based on the following five filling in schemes: Z, CD, II, PP<sub>OLS</sub>, and TSR. From the previous work it was shown that TSR and CMR gave almost identical predictions, and PP<sub>OLS</sub> and II, although theoretically equivalent (Arteaga and Ferrer, 2002), were seen to behave numerically different. For each filling in method, the D (proportional to the Hotelling's  $T^2$ ) and SPE chart limits were determined by passing all the training batches through the monitoring charts and

calculating the approximate 95% and 99 % confidence limits. The correct calculation of Hotelling's statistic ( $D_{ki}$ ) and its confidence limits are presented in the Appendix 2 (since this is different from the usual approximate approach (Nomikos and MacGregor, 1995b)) while the limits on the SPE were calculated as shown in Nomikos and MacGregor(1995b).

To check the validity of the approximate control limits, and to see that they are comparable across all the approaches, the total percentage of samples falling outside the calculated 95% confidence limits for all of the training set and test set data are shown in Table 3.2. As expected the percentage of samples falling outside the 95% intervals for the training set is very close to 5% for all approaches, while for the test set the percentage falling outside the 95% limits is slightly higher (especially for the SPE). However, the important point in this Table is that for "in-control-batches" the probability of violating the control limit is essentially the same for all approaches, thereby allowing valid comparisons to be made for the detection of faulty batches as discussed below.

PERCENTAGE OF SAMPLES ABOVE 95 % CI OUT OF 35,000						Data Set
	CD	II	PPols	TSR	Z	
HT2	4.54	4.96	4.96	4.96	4.96	Training Data Set
SPE	5.17	5.22	5.18	5.13	5.24	
HT2	6.85	5.04	5.04	5.04	5.04	Validation Data Set
SPE	9.17	8.98	10.15	10.08	9.22	

**Table 3.2 Percentage of samples outside of 95% confidence limits for the validation and training data sets**

To compare the detectability of the 5 faults under different “filling in” methods, all of the faulty batches were then passed through the control charts of the various methods. In this study an *alarm* is determined to have been generated when three consecutive samples are outside the 95% confidence limit in either the D or the SPE chart.

Even though both the SPE chart (Nomikos and MacGregor, 1994;1995b) and the D chart were considered (this last statistic has been corrected to take into account the time-varying covariance of the scores, see Appendix 2) all of the faulty batches generated alarms in the SPE chart before they did on the D chart. As each faulty batch is monitored, the number of samples to detection or time to detections is recorded; this is the number of samples after the fault occurs, and until an *alarm* is generated. Table 3.3 lists the mean times to detection (MTD) and standard deviation of the mean time to detection over all the repeats of a fault for each of the considered methods; all quantities have been rounded to the closest integer.

Fault	Method	MTD	$\sigma(\text{MTD})$
FAULT 1	Z	3	0
	CD	3	0
	II	3	0
	PP <sub>OLS</sub>	3	0
	TSR	3	0
FAULT 2	Z	16	3
	CD	17	3
	II	16	3
	PP <sub>OLS</sub>	16	3
	TSR	16	3
FAULT 3	Z	120	11
	CD	120	9
	II	115	10
	PP <sub>OLS</sub>	107	7
	TSR	106	9
FAULT 4	Z	93	6
	CD	108	5
	II	102	5
	PP <sub>OLS</sub>	99	5
	TSR	94	6
FAULT 5	Z	68	1
	CD	53	7
	II	64	10
	PP <sub>OLS</sub>	69	8
	TSR	65	9

**Table 3.3 Comparison of the Mean Time To Detection (MTD) and the standard deviation of the MTD ( $\sigma(\text{MTD})$ ) of different methods**

The results in Table 3.3 and an Analysis of Variance (ANOVA) (Table 3.4) performed on all the repeats for all faults show no statistical evidence of any difference among the methods with any fault type, nor any consistently better method across fault types. A much more massive simulation study would be needed to show any small differences that might exist. The general conclusion is that, even if there are differences, these differences appear to be small, and any differences may depend on the type of faults. This may appear to be a surprising result, but upon reflection it might be reasonable. For each “filling in” method the control chart limits are determined separately to give identical “in-control” error detection rate. However, for every faulty batch, the “in-control” assumption has been broken for every control chart, regardless of the filling-in method. Furthermore, in the presence of a fault, the foundation or theory behind all the “filling-in” methods has been violated. For example, in this situation, the correlation structure in the data is broken and none of the PCA/PLS models are valid, and thereby degrading the MD estimation based on them. As a result, the difference in the time to detection among the charts using the different “filling-in” methods will depend mainly upon how rapidly the fault will cause a significant deterioration in the prediction of the scores. If there is not a great difference in this rate of deterioration relative to the different control limits for each chart, then one might expect there to be little difference among the methods in the time to detection. The time to detection of a fault clearly does not appear to be strongly related to the quality of the “filling-in” or trajectory estimation performance of the “in-control” state. Hence, for the detection of faults, all the monitoring charts make efficient use of the existing data up to the current time and appear to perform well as long as the control limits are individually determined for each “filling-in” option used.

Source	SS	df	MS	F	Prob>F
Columns	3324.9	4	831.23	0.21765	0.92864
Rows	253870	2	126930	33.237	1.52E-14
Interaction	20687	8	2585.8	0.67709	0.71201
Error	2807000	735	3819		
Total	3084900	749			

**Table 3.4 Two Way ANOVA Table for the Times to Detection**

### 3.6. Conclusions

In monitoring the progress of batch processes, the trajectories of all process variables from the current time until the end of the batch are unknown. These remaining trajectory-portions can be estimated in a number of ways: *i*) ad hoc approaches such as using the average trajectories ( $Z$ ), or maintaining for the remainder of the batch the current deviations of the scaled trajectories (CD); or *ii*) more powerful statistical estimation approaches based on using the PCA or PLS models (MD approaches). The latter missing data (MD) approaches are shown to be capable of providing excellent predictions of the reminding process variable trajectories that optimally use all the available data and knowledge of the time varying covariance structure among all the variables over the entire batch, provided by the PCA or PLS models of the batch.

For batch processes that are still in a state of control, the missing data estimation methods provide a powerful way of presenting the expected behavior of the process variable trajectories for the remainder of the batch, and of estimating the final product quality (Nomikos and MacGregor, 1995a). These are both extremely useful tools for monitoring. For the on-line control of final product quality these efficient PLS estimation methods have proven to be critical to the success of batch control (Flores-Cerrillo and MacGregor, 2002;2003;Yabuki and MacGregor, 1997;Yabuki, Nagasawa, and MacGregor, 2000). However, for the purpose of on-line detection of process faults (in process monitoring) the differences among the trajectory estimation methods appears to

be much less critical since the control charts used in each case are tailored to the filling in mechanism employed. All the approaches appear to provide powerful charting methods for monitoring the progress of batch processes.

*Acronyms*

AIC	Akaike Information Criterion
AR	Auto regressive
CD	Current deviation option to fill in the unknown samples in the trajectory
CTA	Chain Transfer Agent
CMR	Conditional Mean Replacement
DMC	Dynamic Matrix Controller
FPRESS	Future Prediction Sum of Squares
FPRMSE	Future Prediction weighted Mean Square Error
II	Iterative Imputation
MCS	Mean centered and auto-scaled
MD	Missing data option to fill in the unknown samples in the trajectory
NOC	Normal Operating Conditions
PCA	Principal Component Analysis
PLS	Projection to Latent Structures
PP	Projection to the model Plane
PP <sub>PLS</sub>	Projection to the model Plane solved by PLS
PP <sub>OLS</sub>	Projection to the model Plane solved by OLS
SCP	Single Component Projection
STD	Samples to detection after fault occurrence and until alarm is generated
TS	Time Series
TSR	Trimmed Score Regression
Z	Zeros option to fill in the unknown samples in the trajectory

# Chapter 4

## Product Design for Batch Processes using Latent Variable Model Inversion via Constrained Optimization

This chapter presents a novel and simple methodology to design optimal operating conditions for batch processes using latent variable regression methods. This new method is illustrated for the design of minimal-duration trajectories for an emulsion polymerization process for a given set of specifications in the final product.

### 4.1. Introduction

Batch processes play an important role in today's industry; specialty chemicals and most of the pharmaceutical industry use batch systems. One of the reasons to use batch processes is their flexibility to produce a wide range of products; this capability is due to the large range of process conditions in which a batch unit can be operated. Typically any batch system will have a determined set of *operating conditions* for each one of the products. For a continuous system, these operating conditions correspond to the desired values of the process variables at steady state, and it is left to the control system to maintain these operating conditions in the presence of disturbances and eventually, to change such operating conditions (e.g. transitioning from one grade to another). As batch systems are dynamic (time varying) systems, their operating conditions are time varying as well. Therefore it is not enough to define the starting point (e.g. initial temperature) for the process variables, but also the path to follow (e.g. heating

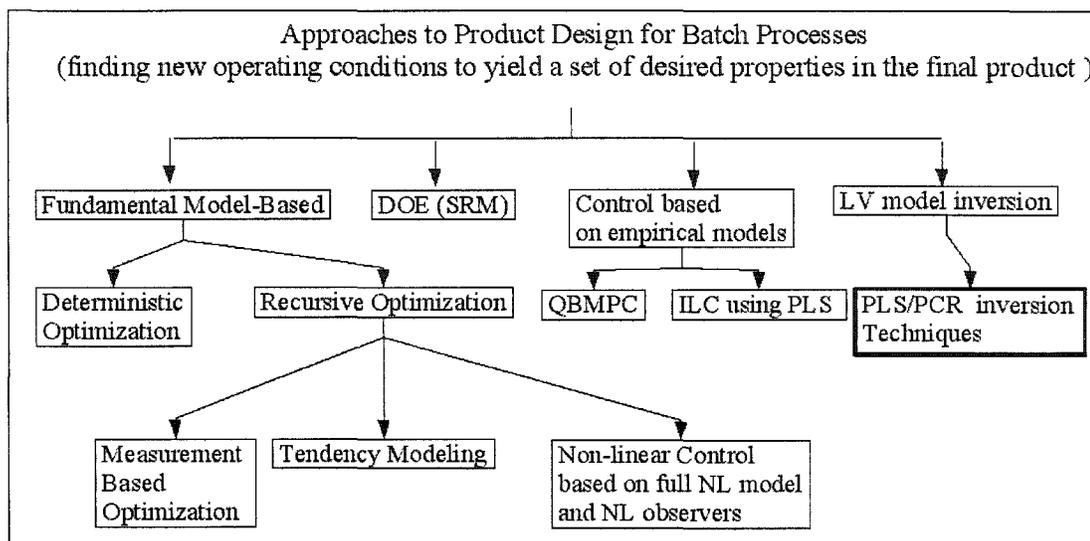
profile vs. time) and final point (e.g. final temperature). All these conditions will have great effects on the physical and chemical properties of the final product.

Determining these *operating conditions* for batch systems is not an easy task due to the fact that the process variables are highly auto and cross correlated (Kourti, 2003), and to the need to estimate the design with additional constraints (e.g. due to equipment capacity).

For example, the estimated operating conditions should achieve a certain set of specifications in the final product (e.g. molecular weight, tensile strength, PSD) while consuming the minimal resources for its operation (e.g. catalyst, raw material, time). Even more difficult is to satisfy these requirements (final product properties and minimal operational cost) when they are coupled with the batch scheduling problem. The solution to this problem is difficult not only because of the complexity of the phenomena involved in the batch operation (or the supply chain) but also because of the different sources of uncertainty.

#### **4.1.1 Background**

The design of batch process operation has been widely studied by researchers. The different methodologies used can be grouped and classified as (Scheme 1): Fundamental model-based; surface response methods; Control-based on empirical methods; and latent variable inversion methods.



**Scheme 1. Classification of the different approaches to batch operation design**

The fundamental model-based optimization for the design of batch process operation involves the use of a highly non-linear differential algebraic equation system. Several approaches have been proposed to solve the optimization of these dynamic systems. The first approach involves a one-time-solving of the complete equation system. This deterministic optimization will give as a result the trajectories for the manipulated variables so that they can be implemented into the system (Biegler, Cervantes, and Wachter, 2002; Vassiliadis, Sargent, and Pantelides, 1994a; and 1994b).

A second approach within the fundamental model-based solutions is to use some simplifications to the full deterministic models and a recursive optimization to “move” the operation in real-time, updating the solution each time a new batch is available. Examples of these proposals are those from Srinivasan *et al.* (2003) who proposed to use a set of “arcs” which are determined from previous knowledge and are continuously updated in a batch to batch mode; the use of tendency models where the kinetics of a reaction system are simplified (Filippi *et al.*, 1986; Fotopoulos, Georgakis, and Stenger, 1998) and the use of non-linear observers with a full non-linear model which can be linearized at each time sample (Clarke-Pringle and MacGregor, 1997; 1998; Kozub and

MacGregor, 1992a; 1992b). The reader is referred to the review by Srinivasan *et al.* (2003) which covers most of the deterministic solutions available, and offers a comparison among them.

Data-based models have also been proposed to solve the batch process operation design. Surface response methods (DOE- SRM) have been widely used in the design of laboratory conditions (Gorret *et al.*, 2004; Rao *et al.*, 2002) for the production of specialty chemicals since these conditions are easily modified at a low cost. These approaches rarely involve the trajectory of a variable, and mostly involve a recipe for process conditions to be kept constant throughout the batch.

A mixture of model base control techniques and iterative learning control are used to develop the Quality based Batch Model Predictive Control (QBMPC) method (Chin, Lee, and Lee, 2000). Such solution assumes that the batch system can be characterized by a single linear input-output model which involves the full trajectory. A more powerful approach is proposed by Flores-Cerrillo and MacGregor (2004) where a latent variable model is used within a recursive optimization using batch to batch information following the same mechanism as in iterative learning control (ILC).

Finally, latent variable regression (LVR) models have also been proposed to estimate from historical data, the set of operating conditions that would yield certain characteristics in the final product (Jaeckle and MacGregor, 1998; and 2000a). In their work, the authors propose the use of *model inversion* to estimate the process conditions from a given set of properties (estimate  $\mathbf{x}$  from  $\mathbf{y}$ ). They consider a continuous Linear low Density Polyethylene (LLDPE) process, and two industrial semi batch reactor examples, with a discrete characterization of the time profiles for the process variables. This is also a one-time solution which will give as a result the trajectories of the manipulated variables (MV's) to be implemented into the process. Although the authors recognize that

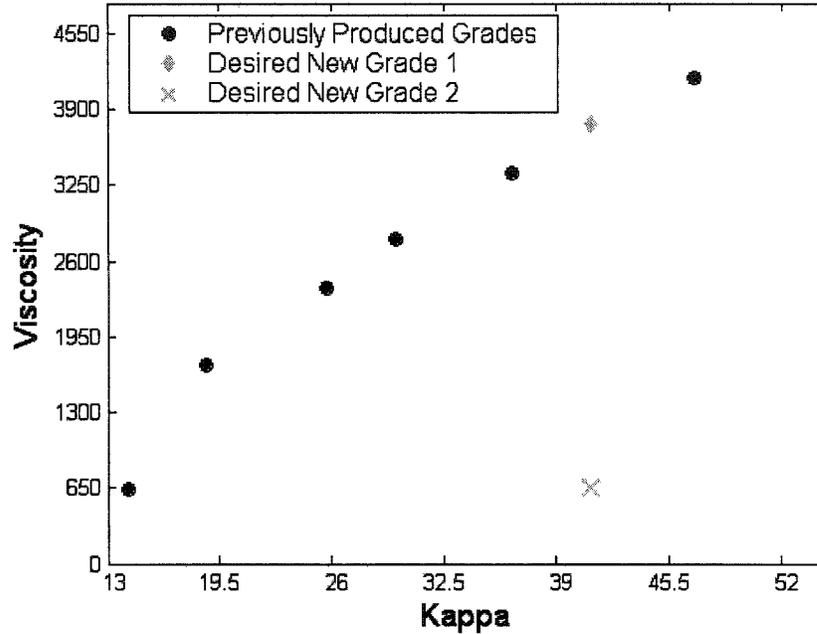
their work is usable to estimate complete batch trajectories, there is no such application reported in the literature. Their proposed approach is summarized next.

#### 4.1.2 Design of operating conditions by LVR model inversion

LVR models will not only describe an X-Y relationship, but will also include the information about the multivariate correlations in each of the spaces (X-X relationships and Y-Y relationships), distinguishing these correlation-structures from the noise present in the data (Burnham, MacGregor, and Viveros, 1999). Jaeckle and MacGregor (1998) propose a solution to the design problem with the use of LVR methods. Their approach is summarized in three steps, after proper mean centering and scaling of all matrices:

*i)* Ensure the new desired set of properties ( $\mathbf{y}_{des}$ ) has the same correlation structure as the rest of the  $\mathbf{Y}$  matrix. Consider the properties plotted in Figure 4.1. In this example from the pulp and paper industry, the kappa (which indicates the relative lignin content of the pulp) and the viscosity are strongly positively correlated in the data from past operation. The desired new grade 2 (Fig. 4.1) has a completely different *balance* of these properties. And since there is no evidence that the process has ever yielded product with this balance of kappa and viscosity, the design exercise cannot be done with this grade. However, the desired new grade 1 does *fall* within the line of grades previously produced, and there is evidence that the process can produce grades with this new desired balance of properties, and the operating conditions can be estimated for it. The square prediction error (SPE) for the desired grade after projecting it onto a PCA model of  $\mathbf{Y}$  is used to measure how close the desired grade is to the existing correlation structure of  $\mathbf{Y}$ . In this step of pre-processing, a variable selection exercise is done in the  $\mathbf{Y}$  matrix, and only one variable representative of each dimension of the  $\mathbf{Y}$  space (or the  $\mathbf{A}$  LV themselves) is kept. After this variable reduction exercise, the retained  $\mathbf{Y}$  matrix is full rank. In the case of the properties shown in Figure 4.1, the effective rank is probably one given the noise in the data. In that case, either of the variables could be used (but only

one) since they are correlated, and the true dimension of  $Y$  is close to one; by doing this, it is assumed that the deviations from perfect correlation are due to measurement error.



**Figure 4.1 Kappa and Viscosity for a Product Family of Pulp**

*ii)* Use the inverse of the model to compute the scores that should yield  $\mathbf{y}_{des}$ .

A PLS model (Eq. 4.1 – 4.5) describes the relation between  $X$  and  $Y$  through their latent variables or scores ( $\mathbf{T}$ ). The scores predict either one of the spaces (Eq. 4.4 and 4.5) by the use of their corresponding loadings. For this particular step, Eq. 4.5 is inverted (using the Moore-Penrose generalized inverse).  $\hat{\mathbf{Y}}$  is the best realization the model has about  $Y$ ; using the pseudo-inverse of  $\mathbf{Q}$  it is possible to compute the scores ( $\boldsymbol{\tau}_{new}$ ) that should yield  $\mathbf{y}_{des}$  (Eq. 4.6). Doing this assumes that  $\mathbf{y}_{des} = \hat{\mathbf{y}}_{des}$ , hence the importance of step  $i$  in verifying that  $\mathbf{y}_{des}$  is close to the plane (low SPE) and then this assumption can be held. (In this notation,  $\boldsymbol{\tau}_i^T$  is the row vector corresponding to the scores for the  $i^{\text{th}}$  observation, for a model with  $I$  observations:  $\mathbf{T}^T = [\boldsymbol{\tau}_1, \boldsymbol{\tau}_2, \boldsymbol{\tau}_3, \dots, \boldsymbol{\tau}_I]$ , refer to Figure 3.3.).

$$\mathbf{X} = \mathbf{TP}^T + \mathbf{E}_x \quad (4.1)$$

$$\mathbf{Y} = \mathbf{TQ}^T + \mathbf{E}_y \quad (4.2)$$

$$\mathbf{T} = \mathbf{XW}^* \quad (4.3)$$

$$\hat{\mathbf{X}} = \mathbf{TP}^T \quad (4.4)$$

$$\hat{\mathbf{Y}} = \mathbf{TQ}^T \quad (4.5)$$

$$\hat{\mathbf{y}}_{\text{des}} = \mathbf{Q}\boldsymbol{\tau}_{\text{new}} \quad (4.6)$$

*iii)* Use the estimated  $\boldsymbol{\tau}_{\text{new}}$  and the model for the correlation structure in  $\mathbf{X}$  to estimate the new operating conditions. Once the new score is computed, then Eq. 4.4 is used to produce the estimate for the new process conditions  $\mathbf{x}_{\text{new}}$ . This assures that the new operating conditions will have the same covariance structure as the ones already experienced at the plant with the assumption of  $\mathbf{x}_{\text{new}} = \hat{\mathbf{x}}_{\text{new}}$  and hence a SPE of zero. The estimated  $\mathbf{x}_{\text{new}}$  is then re-scaled and the mean is added (the inverse exercise as in the pre-processing of the data).

In the framework proposed by Jaeckle and MacGregor (1998) steps *ii* and *iii* can be combined by expressing the new score vector as a function of the new process conditions. The solution ( $\mathbf{x}_{\text{new}}$ ) given by the method is unique except for the scenario where the rank of  $\mathbf{Y}$  is smaller than the number of latent dimensions in  $\mathbf{X}$ . The *extra* dimensions in  $\mathbf{X}$  (which have no impact on  $\mathbf{Y}$ ) define a *null* space, this *null* space represents an operational region (multiple solutions to the problem) where  $\mathbf{x}_{\text{new}}$  can “move”, and still yield the same  $\hat{\mathbf{y}}_{\text{des}}$ . The final operating conditions along this null space can be defined by specifying additional economical or safety criteria.

### 4.1.3 Extensions to the model inversion approach

To introduce physical constraints, Yacoub and MacGregor (2004), suggest the inversion of the model, by imposing constraints on the Hotelling’s  $T^2$  statistic for the new

set of scores. Step *ii* is then computed by minimizing Eq. 4.7 instead of using the inversion framework proposed by Jaeckle and MacGregor (1998).

$$\begin{aligned} \min_{\boldsymbol{\tau}_{new}} \{ & (\mathbf{y}_{des} - \mathbf{Q} \boldsymbol{\tau}_{new})^T \mathbf{G}_1 (\mathbf{y}_{des} - \mathbf{Q} \boldsymbol{\tau}_{new}) \} \\ \text{st.} & \\ T^2 = \sum_{a=1}^A \frac{\boldsymbol{\tau}_{new,a}^2}{s_a^2} & \leq \text{const} \end{aligned} \quad (4.7)$$

In Eq. 4.7,  $s_a^2$  represents the variance of each of the columns of  $\mathbf{T}$  in the PLS model used, *const* represents the magnitude of the constraint (e.g. the 99% confidence limit of the Hotelling's  $T^2$  for the training data set), and  $\mathbf{G}_1$  is a weighting matrix; other constraints can be added to this formulation.

This model inversion approach has been successfully applied to industrial problems (Yacoub and MacGregor, 2004). In the absence of the Hotelling's  $T^2$  constraint, the result obtained by minimizing Eq. 4.7 is identical to the model inversion one.

#### 4.1.4 Contributions of this work

In this chapter, the operating condition design technique proposed by Jaeckle and MacGregor (1998) is applied to the design of batch process operation. Special analysis is proposed to constrain the estimated variable trajectories in the case where multiple solutions exist to the design problem.

A new parallel methodology is also proposed using a multi-way PLS model augmented with the derivative of the process variables with respect to its evolution index. This augmented model is used within an optimization formulation to estimate the trajectories which will yield a certain set of desired specifications in the final product. The solution is not limited to the time varying profiles as it may include initial conditions.

The augmentation of the model with the derivatives of the process variables with respect to its evolution index, and the way it is formulated into the objective function appears to give a stable and realistic solution.

Furthermore, the equation to reconstruct the variable trajectories is reformulated as an optimization problem making possible the consideration of operational constraints. The proposed optimal design methodology is illustrated with an industrial emulsion polymerization process case.

To the best of the author's knowledge, this is the first work considering the design of full trajectories for a batch process, and the inclusion of constraints using latent variable regression (LVR) models.

#### **4.2. First Industrial Example Description**

A relatively simple process is chosen as the illustration example: a pilot scale batch pulp digester. The equipment is under very tight control and disturbances are minimal (laboratory environment).

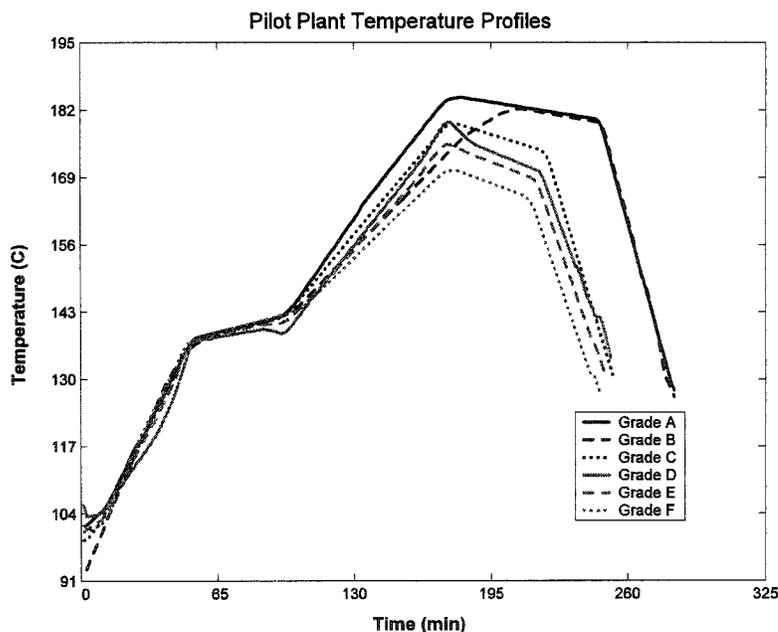
The only process variable considered is the temperature trajectory in the reactor (the initial acid content is not considered in the design exercise since it is difficult to manipulate this variable in the process and there were no available measurements for this variable). The final product is characterized by two of its properties: the kappa number and the viscosity. To remove common-cause variation in the data, each observation in the data set is constructed by averaging three repeats per grade.

In general it can be said that: the higher the temperature and the longer the cook time (batch length), the lower the viscosity and the kappa number. The data set was provided by Tembec Inc. and it consists of 6 grades of pulp, the final properties of each

of the grades is given in Table 4.1, and the temperature profiles to achieve each of these grades in the pilot reactor are shown in Figure 4.2. The quantities in Table 4.1 and Figure 4.2 have been modified for confidentiality reasons; however, the data is still usable for the intended purpose of this thesis.

Grade	Kappa	Viscosity
A	14.24	639.93
B	18.72	1706.9
C	25.71	2372.37
D	29.76	2796.3
E	36.4	3347.63
F	47	4175.21

**Table 4.1 Grades of Pulp produced in the Pilot Digester**

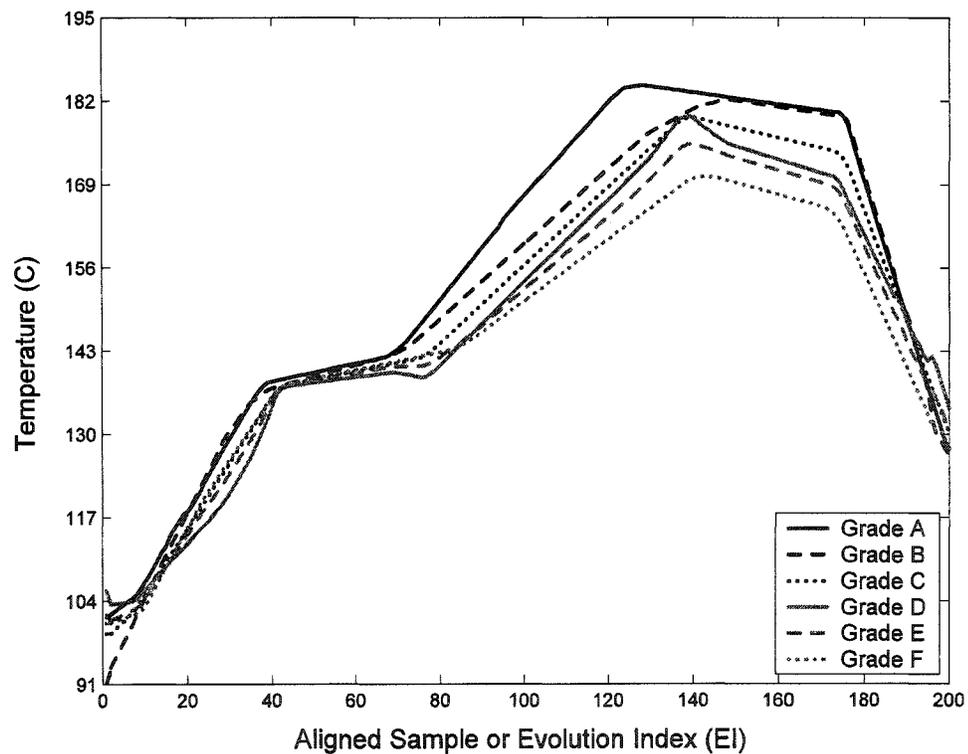


**Figure 4.2 Temperature Profiles for Different Grades of Pulp**

Because this data set is going to be fitted with a multi-way PLS model, it is necessary to perform alignment (see Section 2.5.1) on the temperature profiles because each batch run has a different length. To synchronize this data set, a simple re-sampling

is done to achieve 200 samples for all batches (interpolating linearly when needed) because the profiles are not monotonically increasing or decreasing. Other ways of alignment are possible by segmenting the batch run in different stages (as was done in Chapter 2), however for illustration purposes, this simple linear-alignment is enough.

The temperature profiles after the alignment (Fig. 4.3) are referred to as  $\mathbf{X}_{\text{PBD}}$  ( $6 \times 200$  matrix), the total time for each batch is stored in a separate matrix referred to as  $\mathbf{z}_{\text{PBD}}$  ( $6 \times 1$  column vector), and the final properties for each grade are referred to as  $\mathbf{Y}_{\text{PBD}}$  ( $6 \times 2$  matrix) – the PBD subscript stands for *pilot batch digester*. In general, the trajectories of a batch system will be referred to as  $\mathbf{X}$ , the initial conditions as  $\mathbf{Z}$ , and the set of properties of the final product as  $\mathbf{Y}$ .

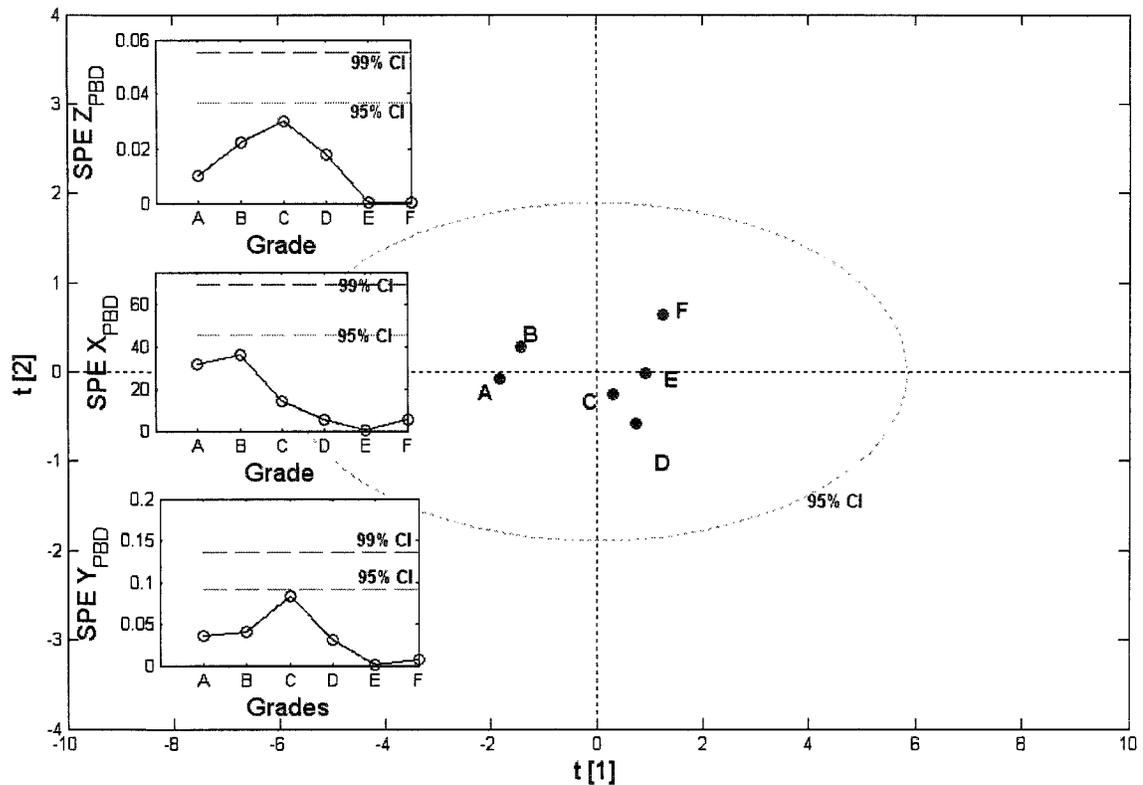


**Figure 4.3 Aligned Temperature Profiles for Different Grades of Pulp**

These three matrices  $Z_{PBD}$ ,  $X_{PBD}$  and  $Y_{PBD}$  are used to illustrate the optimal trajectory design method to be developed in the next Sections. As mentioned before, the proposed methodology makes use of a PLS model. In each of the “design exercises” throughout the next Sections, one grade at a time is left out from the *known* data set, and the final properties of the *left out* grade are specified as the desired characteristics (or desired quality). The design is computed and the estimated temperature profile is compared with the *left out* temperature profile which represents at least one known solution to the design problem.

Because it might be possible to yield the same quality in more than one way (multiple solutions to the problem) this last comparison becomes a *sub-optimal* way to benchmark the estimates. The best way to validate any estimate would be to run the process with the estimated conditions and then measure the properties of the final product; however this was not possible in this study.

As a reference for future Sections, Figure 4.4 presents a summary of a Multi block PLS (MBPLS (Westerhuis, Kourti, and MacGregor, 1998)) model between the operating conditions  $Z_{PBD}$ ,  $X_{PBD}$  and  $Y_{PBD}$ . The model uses two components and the percentage captured variance per component are 97.82% and 0.58% for  $Z_{PBD}$ , 73.52% and 17.34% for  $X_{PBD}$  and 85.43% and 12.61% for the  $Y_{PBD}$  (total R2 is 98% for  $Z_{PBD}$ , 91% for  $X_{PBD}$  and 98% for  $Y_{PBD}$ ). The residual SPE plots for each these matrices are shown in Fig. 4.4 along with the super-score plot (the block weights are set to one for all blocks in this MBPLS model).



**Figure 4.4 Super Score plot ( $t_1$ - $t_2$ ) and square prediction error (SPE) for a MBPLS model with  $Z_{PBD}$ ,  $X_{PBD}$ , and  $Y_{PBD}$**

### 4.3. Batch Trajectory Estimation using latent variable models via optimization with multivariate constraints

In this Section, the complete time profiles for each of the process variables are considered, and the model inversion (or analytical) solution is presented with a thorough analysis of the null space containing the multiple solutions to the problem. The process described in Section 4.2 is used throughout this Section as an illustrative industrial example.

The general methodology to solve the design problem is now segmented in two steps: *i*) determine which point or region in the latent space, represented by a score vector ( $\boldsymbol{\tau}_{\text{new}}$ ) corresponds to the desired set of properties ( $\mathbf{y}_{\text{des}}$ ) and *ii*) use this score vector to reconstruct the operating conditions using the correlation structure captured by the LV model of the  $\mathbf{z}_{\text{PBD}}$  and  $\mathbf{X}_{\text{PBD}}$  spaces.

#### 4.3.1. Solving for $\hat{\boldsymbol{\tau}}_{\text{new}}$ that will yield $\mathbf{y}_{\text{des}}$

Jaeckle and MacGregor (1998) proposed the direct inversion of the latent variable model, once the  $\mathbf{Y}$  is reduced to full rank ( $\boldsymbol{\tau}_{\text{new}} = (\mathbf{Q}^T \mathbf{Q})^{-1} \mathbf{Q}^T \mathbf{y}_{\text{des}}$ ).

However, in some circumstances the new desired quality may not be entirely defined as equality constraints (where all values of  $\mathbf{y}_{\text{des}}$  must have a determined value) but may include inequality constraints. In such cases, the corresponding desired score vector is not easily found by exploration or inversion. To find the desired score vector in such scenario, the following optimization is proposed.

$$\begin{aligned} \min_{\boldsymbol{\tau}_{\text{new}}} & \left\{ (\mathbf{y}_{\text{des}} - \mathbf{Q} \boldsymbol{\tau}_{\text{new}})^T \mathbf{G}_1 (\mathbf{y}_{\text{des}} - \mathbf{Q} \boldsymbol{\tau}_{\text{new}}) + \rho \left( \sum_{a=1}^A \frac{\tau_{\text{new},a}^2}{s_a^2} \right) \right\} \\ \text{s.t.} & \\ \mathbf{A} \mathbf{Q} \boldsymbol{\tau}_{\text{new}} & < \mathbf{b} \end{aligned} \quad (4.7b)$$

The elements of the diagonal matrix of  $\mathbf{G}_1$  may be zeros for those elements of  $\mathbf{y}_{\text{des}}$  which are not subject to equality constraints. These elements will have a corresponding weight in the  $\mathbf{A}$  matrix and a corresponding constraint given in  $\mathbf{b}$ . Also in this equation, the Hotelling's distance has been included as a soft constraint with a given weight  $\rho$ . This soft constraint is particularly useful for selecting a reasonable solution when there are multiple solutions to the problem (i.e. a null space in  $\mathbf{X}$ ).

This objective function is flexible enough to be used instead of the model inversion using the pseudo-inverse of  $\mathbf{Q}$  and will give the same solution in the unconstrained case.

$\mathbf{G}_1$  is usually a diagonal matrix and the diagonal elements will determine how much weight is given to each quality variable  $\mathbf{y}_{des}$  in the solution. This weighting matrix can impose more weight on those properties which are of more interest to the customer. If all the variables of  $\mathbf{y}_{des}$  are equally important to the customer, then the weighting can be assign to be proportional to the fractions of explained variance from the PLS model (Eqs. 4.8 to 4.10). This is the approach taken in this work. This will down-weight a variable which is poorly predicted by the PLS model. In other words,  $\mathbf{G}_1$  in Eq. 4.7b is an indication of how well predicted each quality property is. This is important to consider in Eq. 4.7b since not all columns of  $\mathbf{Y}$  have the same predictability.

$$\mathbf{G}_{1,i} = 1 - \frac{RSSY_i}{TSSY_i} \quad (4.8)$$

$$TSSY_i = \sum_{j=1}^J (\mathbf{Y}_{j,i}^2) \quad (\text{Where } \mathbf{Y} \text{ has } i \text{ columns and } j \text{ rows}) \quad (4.9)$$

$$RSSY_i = \sum_{j=1}^J (\mathbf{Y}_{j,i} - \hat{\mathbf{Y}}_{j,i})^2 \quad (4.10)$$

In equations 4.8 to 4.10, the  $i$  index runs along the columns of  $\mathbf{Y}$  (variables), and  $j$  index along the rows of  $\mathbf{Y}$  (observations),  $TSSY_i$  is the total sum of squares per variable ( $\mathbf{Y}$  is mean-centered),  $RSSY_i$  is the residual sum of squares per variable (sum of squares of the residuals after deflating the  $\mathbf{Y}$  matrix with the PLS model,  $\mathbf{Res}_y = \mathbf{Y} - \mathbf{TQ}^T$ ).

### 4.3.2. Re-constructing the operating conditions $\hat{\mathbf{x}}_{new}$ from $\boldsymbol{\tau}_{new}$

The use of latent variable techniques in modeling the trajectories of the variables in a batch process and their effect on the final quality is well documented (Nomikos and MacGregor, 1995a). And, as shown in Chapter 3, the reconstruction of the variable trajectories using a score estimate and the corresponding loadings from the PCA model is an accurate approach. These modeling features of a multiway latent variable model are now applied in solving the design problem.

The use of complete trajectories in the design exercise is proposed originally in (Jaeckle and MacGregor, 2000a), however this is the first work where the concept is applied and investigated in detail.

Once the new score vector ( $\boldsymbol{\tau}_{new}$ ) is calculated (Eq. 4.7 or 4.7b), the process conditions are estimated (Eq. 4.11). It is assumed that these conditions will follow exactly the correlation structure of the data used as a training set for the PLS model, and therefore the value of SPE will be equal to zero ( $\mathbf{x}_{new} = \hat{\mathbf{x}}_{new}$ ).

$$\hat{\mathbf{x}}_{new} = \mathbf{P}\boldsymbol{\tau}_{new} \quad (4.11)$$

The problem is now extended to consider a set of new operating conditions  $\mathbf{x}_{new}$ , which will:

- a) Project to the desired point of the latent space (given by  $\boldsymbol{\tau}_{new}$ ), while
- b) Tolerating a SPE different than zero but within control limits.

These requirements could be built into Eq. 4.7b since the relationship between  $\boldsymbol{\tau}_{new}$  and  $\mathbf{x}_{new}$  is known (given by the  $\mathbf{W}^*$  loading matrix, Eq. 4.13). Building such an

equation will result in a single optimization problem as in Yacoub and MacGregor (2004).

However, in batch systems, the size of  $\mathbf{x}_{\text{new}}$  is easily in the order of hundreds of elements. Solving the optimization problem which will estimate the  $\mathbf{x}_{\text{new}}$  vector that best predicts  $\mathbf{y}_{\text{des}}$  subject to the PLS model (Eq. 4.1 to 4.5); and subject to the constraints in the Hotelling's  $T^2$  becomes difficult to solve when the SPE is free to be different than zero.

As soon as the SPE is free to be different than zero, then all the solutions lying in the residual space (orthogonal to the plane defined by the PLS) that project in the same point of the latent space – given by  $\boldsymbol{\tau}_{\text{new}}$  – will become a solution to the problem. This situation coupled with the hard constraint in Eq. 4.3 makes the problem difficult to solve in one single objective function.

Because of this, the problem was divided in two optimization problems:

a) Estimate the best set of scores ( $\boldsymbol{\tau}_{\text{new}}$ ) that better describe  $\mathbf{y}_{\text{des}}$  subject to the constraint in the Hotelling's  $T^2$  (Eq. 4.7 or 4.7b), and

b) Estimate the best set of operating conditions ( $\mathbf{x}_{\text{new}}$ ) that will project in the latent point specified by  $\boldsymbol{\tau}_{\text{new}}$  (1<sup>st</sup> term of Eq. 4.12), while being at a constrained distance from the plane (second term of Eq. 4.12)

$$\min_{\mathbf{x}_{\text{new}}} \left\{ \left( \boldsymbol{\tau}_{\text{xnew}} - \boldsymbol{\tau}_{\text{new}} \right)^T \mathbf{G}_2 \left( \boldsymbol{\tau}_{\text{xnew}} - \boldsymbol{\tau}_{\text{new}} \right) + \left( \mathbf{x}_{\text{new}} - \hat{\mathbf{x}}_{\text{new}} \right)^T \boldsymbol{\Lambda} \left( \mathbf{x}_{\text{new}} - \hat{\mathbf{x}}_{\text{new}} \right) \right\} \quad (4.12)$$

In equation 4.12,  $\mathbf{G}_2$  and  $\Lambda$  are weighting matrices (taken to be diagonal in this thesis); the hard constraint given by Eq. 4.3 is converted into a soft constraint in the first term of Eq. 4.12 where  $\boldsymbol{\tau}_{\text{new}}$  is the score vector corresponding to the  $\mathbf{x}_{\text{new}}$  column vector (given by Eq. 4.3 and written explicitly in Eq. 4.13). The second term of Eq. 4.12 provides for a small SPE. In this term,  $\hat{\mathbf{x}}_{\text{new}}$  is the prediction of  $\mathbf{x}_{\text{new}}$  given by the model (Eq. 4.14). Incorporating equations 4.13 and 4.14 into Eq. 4.2 gives Eq. 4.15 (where  $\hat{\boldsymbol{\tau}}_{\text{new}}$  is the solution to Eq. 4.7b)

$$\boldsymbol{\tau}_{\text{new}} = \mathbf{W}^{*\text{T}} \mathbf{x}_{\text{new}} \quad (4.13)$$

$$\hat{\mathbf{x}}_{\text{new}} = \mathbf{P} \boldsymbol{\tau}_{\text{new}} \quad (4.14)$$

$$\min_{\mathbf{x}_{\text{new}}} \left\{ \left( \mathbf{W}^{*\text{T}} \mathbf{x}_{\text{new}} - \hat{\boldsymbol{\tau}}_{\text{new}} \right)^{\text{T}} \mathbf{G}_2 \left( \mathbf{W}^{*\text{T}} \mathbf{x}_{\text{new}} - \hat{\boldsymbol{\tau}}_{\text{new}} \right) + \left( \mathbf{x}_{\text{new}} - \mathbf{P} \mathbf{W}^{*\text{T}} \mathbf{x}_{\text{new}} \right)^{\text{T}} \Lambda \left( \mathbf{x}_{\text{new}} - \mathbf{P} \mathbf{W}^{*\text{T}} \mathbf{x}_{\text{new}} \right) \right\} \quad (4.15)$$

Equation 4.15 can further be manipulated to the form of a standard quadratic programming (QP) problem (Eq. 4.16):

$$\min_{\mathbf{x}_{\text{new}}} \left\{ \frac{1}{2} \mathbf{x}_{\text{new}}^{\text{T}} \mathbf{H} \mathbf{x}_{\text{new}} + f^{\text{T}} \mathbf{x}_{\text{new}} \right\} \quad (4.16)$$

where

$$\mathbf{H} = \mathbf{W}^* \mathbf{G}_2 \mathbf{W}^{*\text{T}} + \Lambda - \Lambda \mathbf{P} \mathbf{W}^{*\text{T}} - \mathbf{W}^* \mathbf{P}^{\text{T}} \Lambda + \mathbf{W}^* \mathbf{P}^{\text{T}} \Lambda \mathbf{P} \mathbf{W}^{*\text{T}}$$

$$f^{\text{T}} = -\hat{\boldsymbol{\tau}}_{\text{new}}^{\text{T}} \mathbf{G}_2 \mathbf{W}^{*\text{T}}$$

The weighting matrix  $\mathbf{G}_2$  is taken as a diagonal matrix with ones in the main diagonal (since it is desirable to have all the elements in  $\boldsymbol{\tau}_{\text{new}}$  equal to those in  $\hat{\boldsymbol{\tau}}_{\text{new}}$ ); this

matrix has as many columns and rows as components in the PLS model. Finally, the weighting matrix  $\Lambda$  is established (in a similar way as  $\mathbf{G}_1$  was) using the fractional captured variance, in this case for the columns in  $\mathbf{X}$  from the PLS model (Eqs. 4.17 to 4.19).

Because of the unfolding performed on the 3D batch data converting it into a 2 way matrix (Fig. 3.1) the final  $\mathbf{X}$  matrix contains the complete time profiles for each of the process variables. Therefore the PLS model will explain for each variable, those periods of evolution where there is structure and information and disregard periods of noise or random behavior. By weighting the second term of Eq. 4.16 with the time varying captured variance per variable; the formulation will accomplish better prediction for those periods of the batch which are well explained by the PLS model, down weighting those periods of the batch which are noise and with poor structure in each variable.

Defining  $\mathbf{G}_2$  and  $\Lambda$  in this way will enable the objective function to emphasize the accuracy of the solution to those variables of  $\mathbf{Y}$  and  $\mathbf{X}$  respectively, which are better explained by the model and hence making the solution robust to the presence of uncertainty which can be *filtered* through the PLS model.

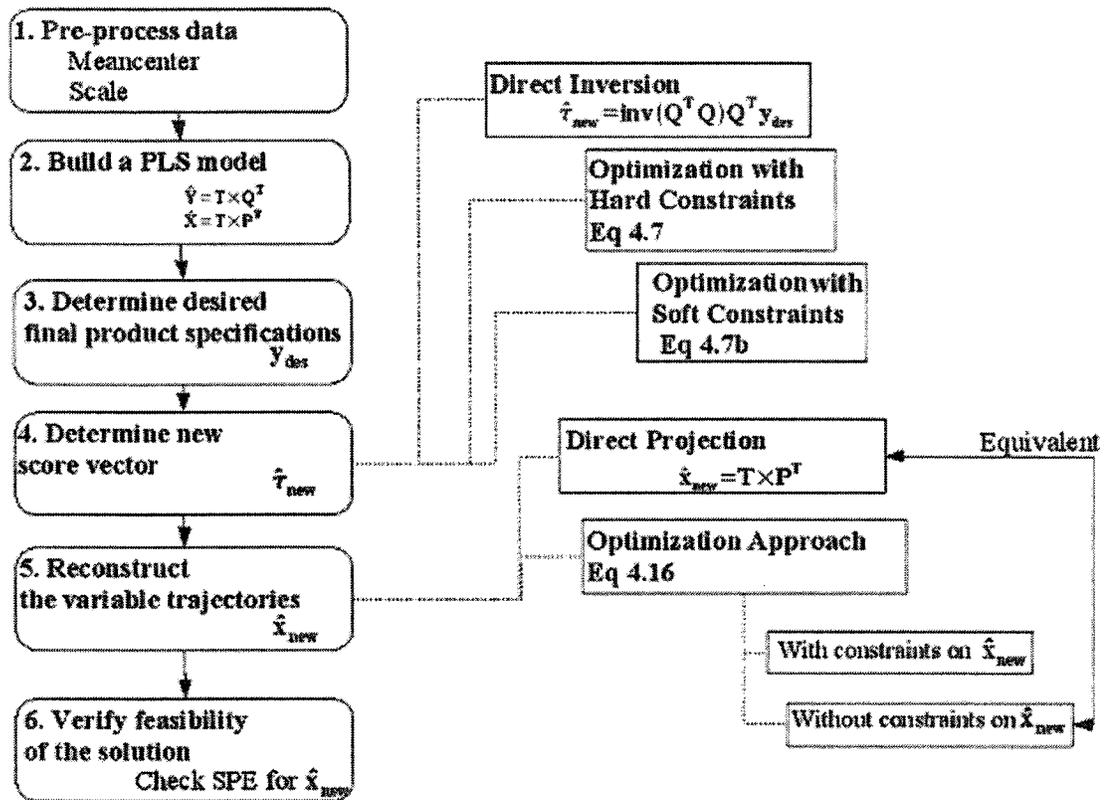
$$\Lambda_i = 1 - \frac{RSSX_i}{TSSX_i} \quad (4.17)$$

$$TSSX_i = \sum_{j=1}^J (\mathbf{X}_{j,i}^2) \quad (\text{where } \mathbf{X} \text{ has } i \text{ columns and } j \text{ rows}) \quad (4.18)$$

$$RSSX_i = \sum_{j=1}^J (\mathbf{X}_{j,i} - \hat{\mathbf{X}}_{j,i})^2 \quad (4.19)$$

The design methodology as proposed up to this point and the possible options are illustrated in Scheme 2. When direct inversion is considered in step 4 and, either direct

projection or optimization with no constraints in  $\hat{\mathbf{x}}_{\text{new}}$  is considered in step 5, the solution is identical to the one obtained by model inversion technique.

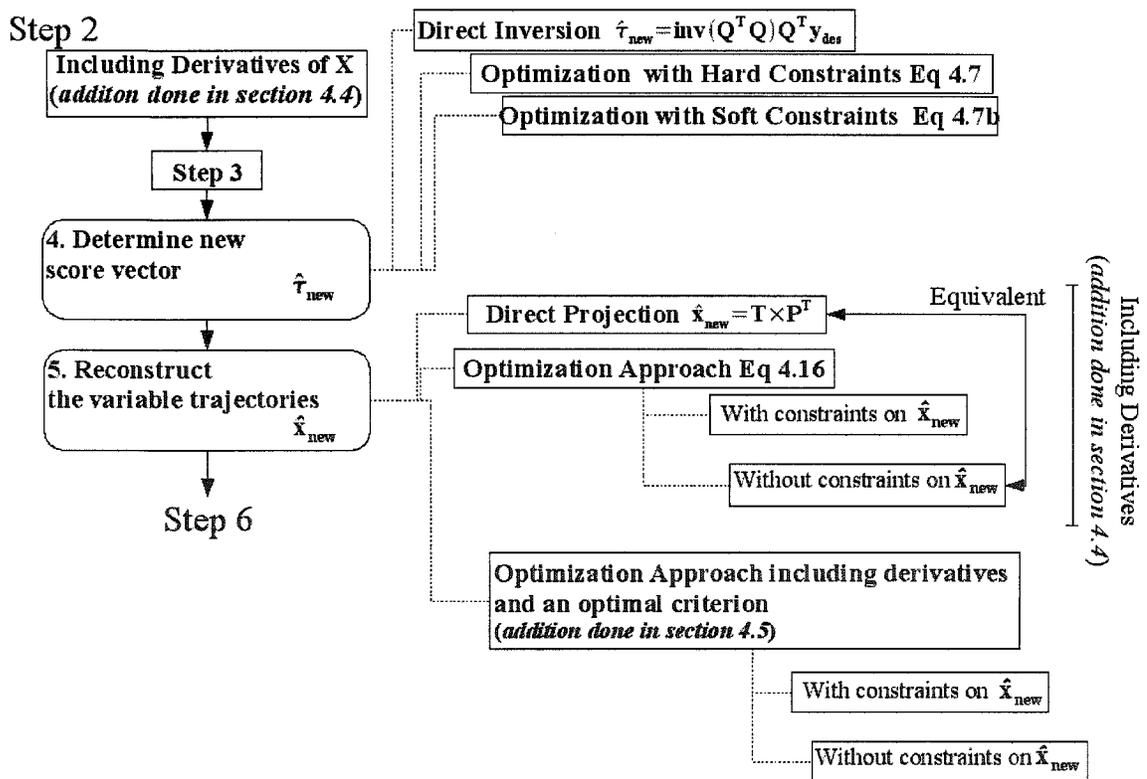


**Scheme 2. Steps to follow and computational options in the design of batch process operating conditions**

In Section 4.3.3, this design method illustrated in Scheme 2 is applied to the industrial process described in Section 4.2. In the first part of Section 4.3.3 (unconstrained solution) the design is done via direct inversion for step 4 and direct projection for step 5. In the second part of Section 4.3.3 (constrained solution), the multiple solutions to the problem, as solved in the unconstrained Section, are explored. Some of these multiple solutions are computed with the use of hard and soft constraints in step 4, and their corresponding trajectories are estimated using Eq. 4.16 with no

constraints on  $\hat{\mathbf{x}}_{\text{new}}$ . In the later part of Section 4.3.3, direct inversion is used in step 4, and bounds are imposed on  $\hat{\mathbf{x}}_{\text{new}}$  in step 5 to contrast the estimated trajectories with the ones obtained when constraints are considered in step 4 of the methodology.

In Section 4.4, the whole technique is extended to include the derivatives of the variable trajectories. This extension affects the model building in step 2, throughout the reconstruction step 5. Inclusion of constraints on  $\mathbf{x}_{\text{new}}$  is still optional. Further modification to the technique using derivatives is considered in Section 4.5 to include an optimal criterion, and is illustrated in Section 4.6 where the optimal trajectories are estimated for an industrial emulsion polymerization process. The flow of the following Sections and the additions to the design technique are illustrated in Scheme 3.



Scheme 3. Additions to the design technique done in Sections 4.4 and 4.5

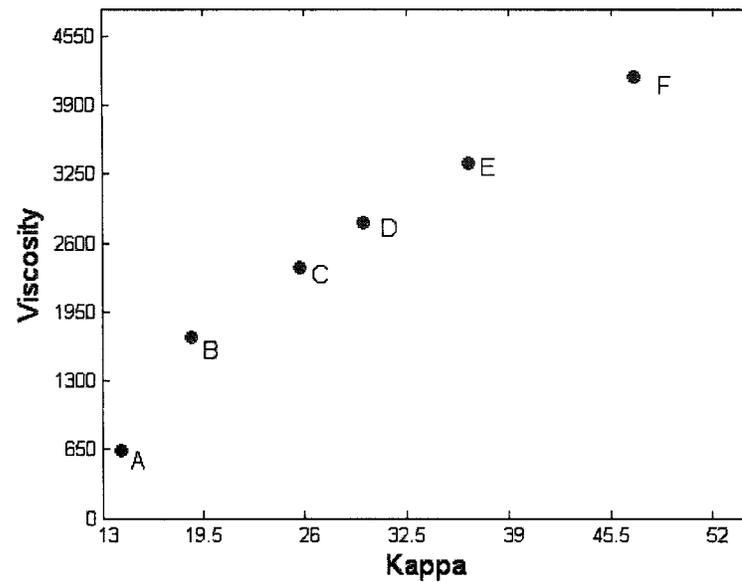
The final design methodology is flexible enough to consider constraints in the desired specifications for the final product; and to estimate the operating conditions in an optimal way with the possible inclusion of operational constraints.

### 4.3.3. Application to an Industrial Pulp Digester

#### *a) Unconstrained solution*

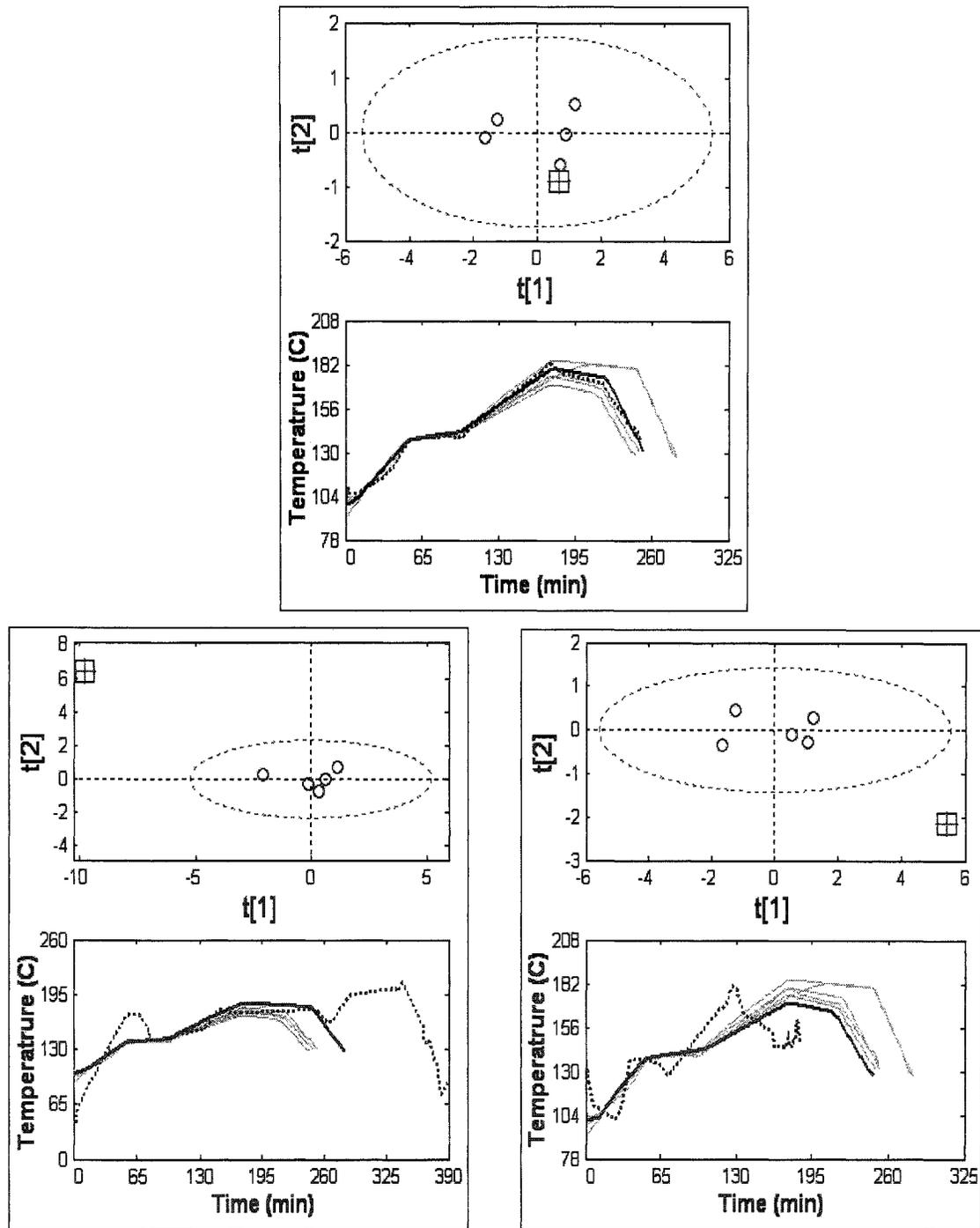
The product quality variables for the grades being considered (listed in Table 4.1) are plotted in Fig. 4.4. The design exercise is first performed (leaving out one grade at a time) with no restrictions to the Hotelling's  $T^2$  in Eq. 4.7b, using both quality variables (kappa and viscosity) and the  $[\mathbf{z}_{new}\mathbf{x}_{new}]$  solution is computed as  $\mathbf{P}\boldsymbol{\tau}_{new}$  (Eq. 4.11) unless it is indicated otherwise. The final results are presented in their raw values (re-applying the time scale to the trajectories).

The MBPLS model for each case is built with two principal components. From Fig. 4.5, it is clear that the design exercise for grades A and F represent an extrapolation to the model, the design for grades B, C, D and E require an interpolation. Figure 4.6 illustrates the solution for grade C (which is representative of all other intermediate grades), and for grades A and F. In this Figure, each of the subplots contain the temperature trajectories and the corresponding scores for the new design grades and the existing grades used in the model; these plots clearly show how the design for grades A and F represents an extrapolation in the super-score spaces.



**Figure 4.5 Grades of pulp being produced by the pilot batch digester**

In Figure 4.6, the dashed ellipses correspond to the 90% confidence limit around the origin (Nomikos and MacGregor, 1995b). This unconstrained solution is found to be identical to the one obtained by model inversion (Jaeckle and MacGregor, 1998). In the case of the design of an intermediate grade (Fig. 4.6 top); the solution appears to be very accurate; this does not happen when the desired grade scores are outside the normal region of operation (Fig. 4.6 bottom subplots).



**Figure 4.6** Score Plots ( $\circ$  model,  $\square$   $\tau_{new}$ ,  $+$   $\tau_{xnew}$ ) and Design Estimates (dotted line), known solution (solid black) and model building grades (gray lines), for grade C (top), A (bottom left) and F (bottom right)

In the following sub-Sections, these “unique” solutions are analyzed and contrasted with the multiple possible solutions to the problem which are obtained when only one quality variable is considered.

*b) Constrained Solution*

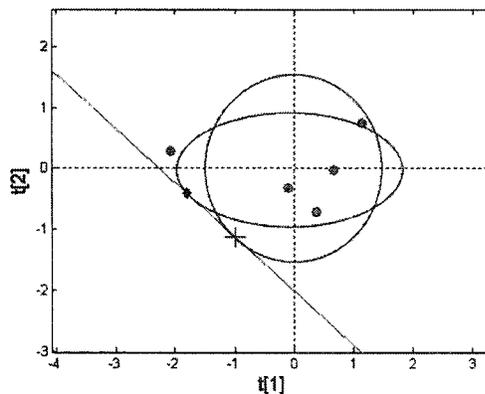
Jaeckle and MacGregor (1998) recognized the existence of a unique solution if the effective rank of  $\mathbf{Y}$  is equal to the number of latent variables in  $\mathbf{X}$ . The solution obtained in the past Sections was unique because it was assumed that the  $\mathbf{Y}$  had a rank of two. Table 4.2 lists the percentage captured variance for a PCA model fitted for  $\mathbf{X}_{\text{PBD}}$  and  $\mathbf{Y}_{\text{PBD}}$  separately. From this Table it is clear that  $\mathbf{X}$  is well explained with 2 latent variables and  $\mathbf{Y}$  with one.

Component	PCA R2 ( $\mathbf{X}_{\text{PBD}}$ )	PCA R2CUM ( $\mathbf{X}_{\text{PBD}}$ )	PCA R2 ( $\mathbf{Y}_{\text{PBD}}$ )	PCA R2CUM ( $\mathbf{Y}_{\text{PBD}}$ )
1	75.68%	75.68%	99.00%	99.00%
2	17.54%	93.22%	1.00%	100.00%
3	6.78%	100.00%	.	.

**Table 4.2 Percentage captured variance of X and Y in individual PCA models**

Performing a variable selection exercise in the  $\mathbf{Y}_{\text{PBD}}$  space (keeping only kappa) reduces the rank of  $\mathbf{Y}_{\text{PBD}}$  to be one. The solution to the design exercise from Section 4.3.2, after performing this variable selection, includes a *null space* since the effective rank of  $\mathbf{Y}$  is smaller than the rank of  $\mathbf{X}$ . Therefore the inversion is projecting from a lower dimensional space to a higher dimensional space (Jaeckle and MacGregor, 1998; 2000a). Since  $\mathbf{X}_{\text{PBD}}$  is only one dimension bigger than  $\mathbf{Y}_{\text{PBD}}$ , then the null space is a one dimensional space (a line). This is illustrated in Figure 4.7 for the design of grade A from grades B to F. The solid line in this plot represents the null space contained in the score space of the  $\mathbf{T}$ ; any  $\boldsymbol{\tau}_{\text{new}}$  along this line satisfies  $\mathbf{y}_{\text{des}} = \hat{\mathbf{y}}_{\text{des}} = \mathbf{Q}\boldsymbol{\tau}_{\text{new}}$ .

When a null space exists (infinite solutions), the solution obtained by the optimization of Eq. 4.16 with no constraints will depend on the initial guess; while the solution given by the pseudo-inverse of  $\mathbf{Q}$  gives the one with minimal Euclidean norm (Jaeckle and MacGregor, 1998). This latter solution is not the best possible solution since Euclidean distances in the score spaces are not meaningful. A minimal Hotelling's  $T^2$ , or Mahalanobis distance should be used. To find this point along the null space, equation 4.7b is used instead of equation 4.7 (using a soft constraint instead of a hard constraint in the Hotelling's  $T^2$ ). This formulation is easy to solve since the Mahalanobis distance is a quadratic function as well. The two solutions along the null space are illustrated in Figure 4.7. The minimum Euclidean distance is found with a circle of minimum radius which is tangent to the null space, the minimum Mahalanobis distance is found with an elliptic function (given by the variances of the scores) which is tangent to the null space.



**Figure 4.7 Null Space and Minimum Euclidian Distance (+) and Mahalanobis Distance (♦) Solution along the Null Space in the Design of Grade A in the Pilot Batch Digester**

The calculation of the null space is given by (Jaeckle and MacGregor, 2000a), for a PCR model; however, it is easily modified for a PLS model. The null space can also be found and understood in a more intuitive way as shown below.

One condition where a null space will exist is when the effective rank of  $\mathbf{Y}$  (referred to as  $k$  in the work by Jaeckle and MacGregor, 2000a) is smaller than the effective rank of  $\mathbf{X}$  (referred to as  $A$ ). The dimension of the null space is given by  $A-k$ , in such case; there are  $(A-k)$  dimensions in which  $\boldsymbol{\tau}_{\text{new}}$  can move without changing the predicted value of  $\mathbf{y}_{\text{des}}$ . This can be expressed as,

$$\hat{\mathbf{y}}_{\text{des}} = \mathbf{Q}(\boldsymbol{\tau}_{\text{new}} + \Delta\boldsymbol{\tau}_{\text{new}}) \quad (4.20)$$

In order for the movement ( $\Delta\boldsymbol{\tau}_{\text{new}}$ ) not to affect the prediction of  $\mathbf{y}_{\text{des}}$ , and hence become a null space, it is necessary that,

$$\mathbf{Q}\Delta\boldsymbol{\tau}_{\text{new}} = 0 \quad (4.21)$$

the  $\mathbf{Q}$  matrix has  $k$  rows and  $A$  columns, so Eq. 4.21 is a linear equation system with  $k$  equations and  $A$  variables,

$$\begin{aligned} q_{1,1}\Delta\tau_1 + q_{1,2}\Delta\tau_2 + \cdots + q_{1,A}\Delta\tau_A &= 0 \\ q_{2,1}\Delta\tau_1 + q_{2,2}\Delta\tau_2 + \cdots + q_{2,A}\Delta\tau_A &= 0 \\ \vdots & \\ q_{k,1}\Delta\tau_1 + q_{k,2}\Delta\tau_2 + \cdots + q_{k,A}\Delta\tau_A &= 0 \end{aligned} \quad (4.22)$$

If the variables of  $\mathbf{Y}$  are all independent of each other (as it is in the methodology proposed by Jaeckle and MacGregor), then for  $k < A$  the system in Eq. 4.22 has more variables than equations and there is an infinite number of solutions along the space described by the linear system given by Eq. 4.22 (which depends on the values of  $\mathbf{Q}$ ).

For example in the case of the pilot batch digester, if only kappa is considered ( $k$  is equal to one and  $A$  is equal to two) the directions of movement in the  $t_1$ - $t_2$  plane

( $\Delta\tau_{\text{new}}$ ) that will not affect the predicted  $\mathbf{y}_{\text{des}}$  are given by:  $q_{1,1}\Delta\tau_1 + q_{1,2}\Delta\tau_2 = 0$  (the null line in Fig. 4.7 is calculated in this way). This mathematical path to compute the null-space gives the same solution as the one proposed by Jaeckle and MacGregor (2000) (which performs SVD on the  $\mathbf{Q}$  matrix) if the columns of  $\mathbf{Y}$  are orthogonal.

If, in the same design exercise, both properties are considered in the  $\mathbf{Y}$  space (kappa and viscosity): The effective rank of  $\mathbf{Y}$  is being set to two and the problem has a unique solution. However, due to the ill-conditioning of the  $\mathbf{Y}$  space, the presence of noise, and the fact that this desired grade is outside the range of grades previously produced at this site; the desired scores for this unique analytical solution will appear well outside the 99% confidence region for the Hotelling  $T^2$ . This situation may arise in any case where the selection of orthogonal columns of  $\mathbf{Y}$  is not easy nor clear, and correlated columns of  $\mathbf{Y}$  are included in the design problem.

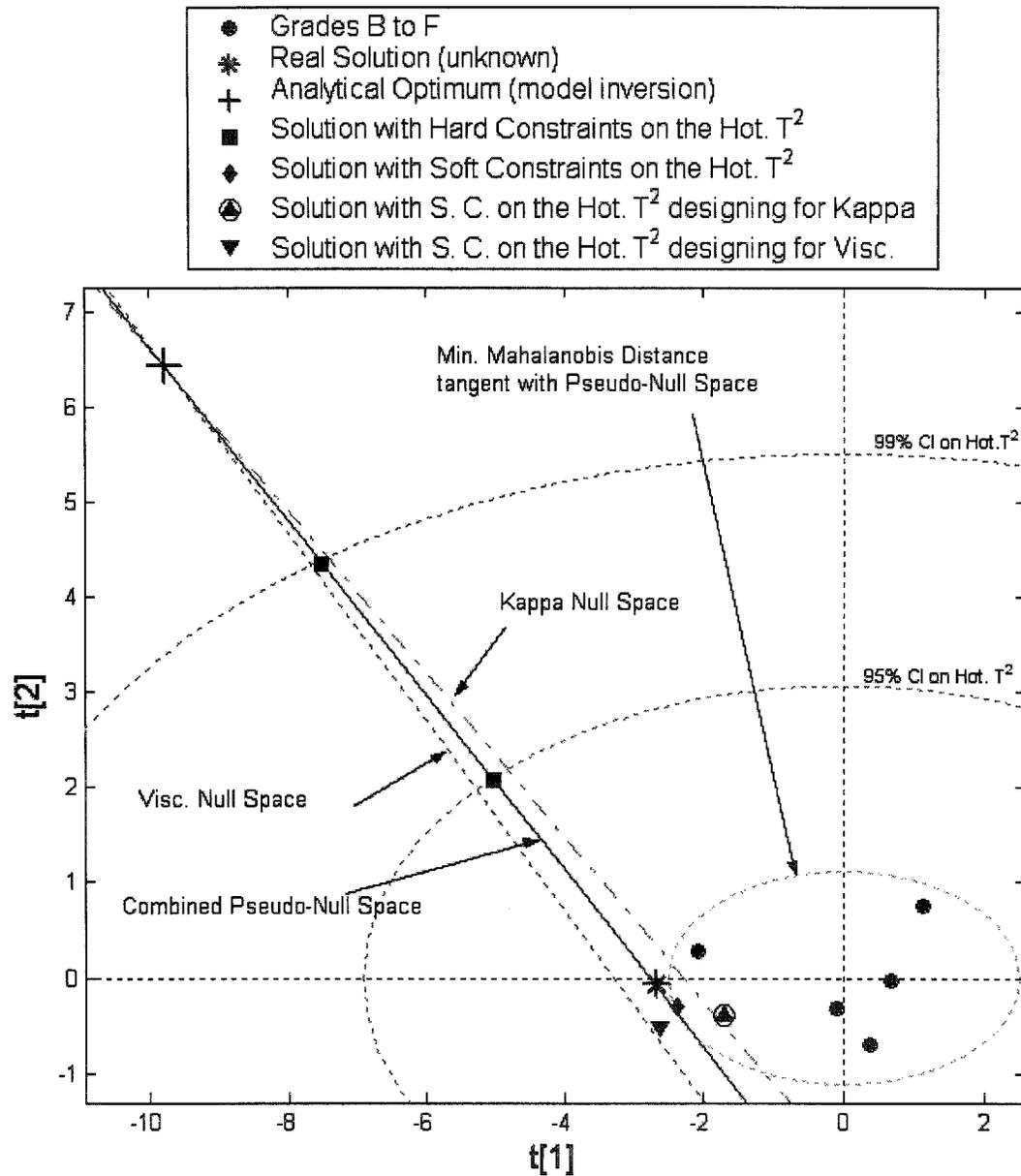
In a situation like this, there are two paths to follow:

- a) Consider an alternative solution (with minimal Mahalanobis distance) along any of the univariate null-spaces. These univariate null-spaces are given by each of the equations in the linear system described by equation 4.22. If the solution is moved along the null space for variable  $i$  in  $\mathbf{y}_{\text{des}}$  ( $\mathbf{y}_{\text{des},i}$ ), the prediction for this particular variable will not change ( $\hat{\mathbf{y}}_{\text{des},i} = \mathbf{y}_{\text{des},i}$ ). However, the prediction for the rest of the variables of  $\mathbf{y}_{\text{des}}$  will change more or less depending on how correlated these elements are with  $\mathbf{y}_{\text{des},i}$  ( $\hat{\mathbf{y}}_{\text{des},j} \neq \mathbf{y}_{\text{des},j}$  for any element  $j \neq i$ ). Because of this, it is clear that the SPE for  $\mathbf{y}_{\text{des}}$  ( $\text{SPE} = (\hat{\mathbf{y}}_{\text{des}} - \mathbf{y}_{\text{des}})^T \times (\hat{\mathbf{y}}_{\text{des}} - \mathbf{y}_{\text{des}})$ ) given by any alternative solution along the univariate null-spaces will be different from zero.

- b) Re-consider the rank of  $\mathbf{Y}$ , compute a *pseudo-null space* and find an *alternative solution along this space*. From a separate PCA model on  $\mathbf{Y}$ , it is clear that the effective rank of  $\mathbf{Y}$  is one (Table 4.2). With this in mind, a pseudo-null space can be computed using the same mathematical steps proposed by Jaeckle and MacGregor (2000a), which involve a SVD on the  $\mathbf{Q}$  loadings matrix. This space is referred to as a *pseudo-null space* because the problem has a unique analytical solution and any solution along this pseudo-null space will have an SPE ( $\text{SPE} = (\hat{\mathbf{y}}_{\text{des}} - \mathbf{y}_{\text{des}})^T \times (\hat{\mathbf{y}}_{\text{des}} - \mathbf{y}_{\text{des}})$ ) different than zero (any solution along a true multivariate null space has an  $\text{SPE} = 0$ ). However, this pseudo null-space exhibits two important characteristics. First, the pseudo-null space (computed by SVD on  $\mathbf{Q}^T$  as shown in (Jaeckle and MacGregor, 2000)) falls in between the existing univariate null spaces, and second, it represents the path of minimal increment in the SPE for  $\mathbf{y}_{\text{des}}$ .

The first of these concepts is illustrated in Figure 4.8. The two univariate null spaces (computed using the equation system defined by Eq. 4.22) are very close to each other (because of the high correlation between kappa and viscosity) and the pseudo-null space (computed by SVD on  $\mathbf{Q}^T$  as shown in (Jaeckle and MacGregor, 2000a)) is between the two univariate null spaces.

Because the  $\mathbf{Y}$  is considered to be full rank, then this problem has a unique analytical solution ( $\dagger$ ) since the considered rank of  $\mathbf{Y}$  is two, and  $\mathbf{X}$  has 2 latent variables as well.

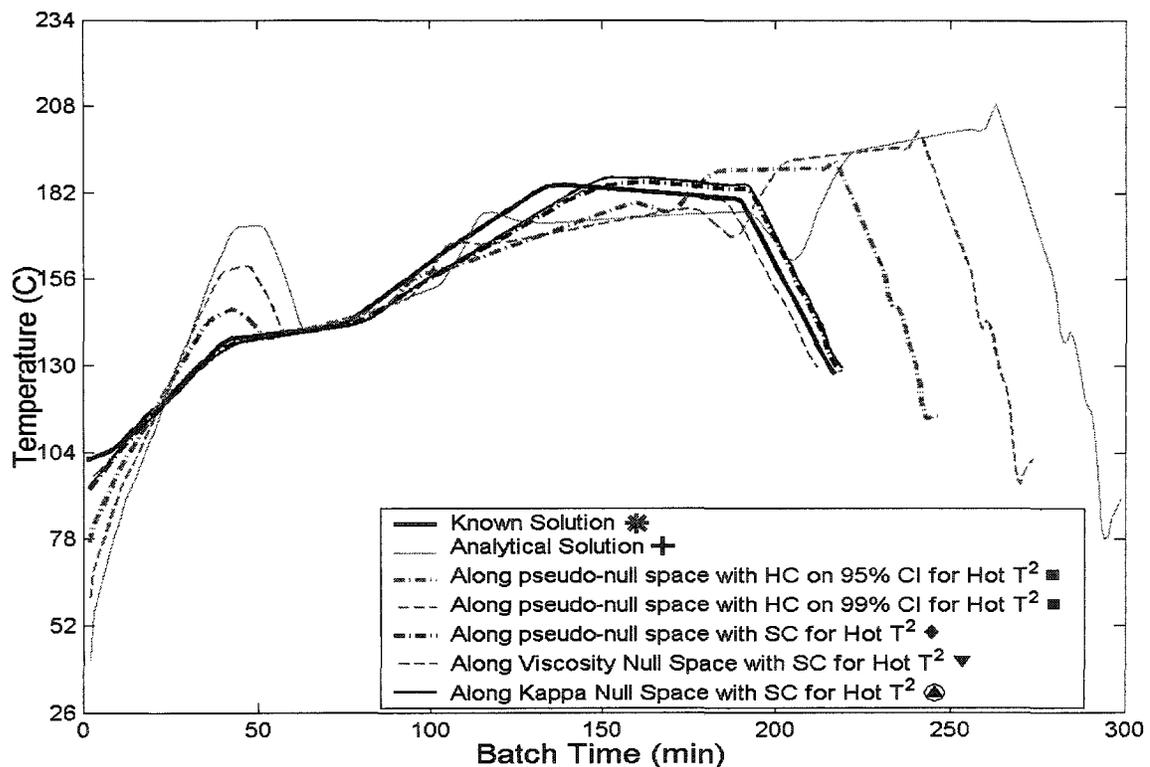


**Figure 4.8 Univariate null spaces, pseudo-null space and several solutions for the design exercise in Section 4.3.2**

Solving the problem by minimizing Eq. 4.7 and 4.7b yields other solutions (Fig. 4.8), these solutions appear along the pseudo-null space; hard constraint solutions (■ marker in Fig. 4.8) appear at the intersect of the ellipse corresponding to the constraint

and the pseudo-null space, the solution with soft-constraints ( $\blacklozenge$  marker in Fig. 4.8) is at the point along the pseudo-null space, tangent with the smallest possible ellipse. Other solutions are illustrated as well (Fig. 4.8), such as the ones obtained if any of the univariate null-spaces are followed with soft constraints on the Hotelling's  $T^2$  ( $\blacklozenge$  and  $\blacktriangledown$  markers in Fig. 4.8).

It is interesting to notice that the unknown true solution (\*), which is the score vector obtained when the known trajectory for Grade A is projected onto the PLS latent space, projects very close to the solutions obtained with soft-constraints ( $\blacklozenge$ ,  $\blacklozenge$  and  $\blacktriangledown$ ). The temperature trajectories corresponding to these solutions (Fig. 4.9) are all close to the known trajectory for grade A.



**Figure 4.9** Temperature Trajectories for the several solutions on the null and pseudo-null space for the design exercise of grade A in the pilot batch digester.

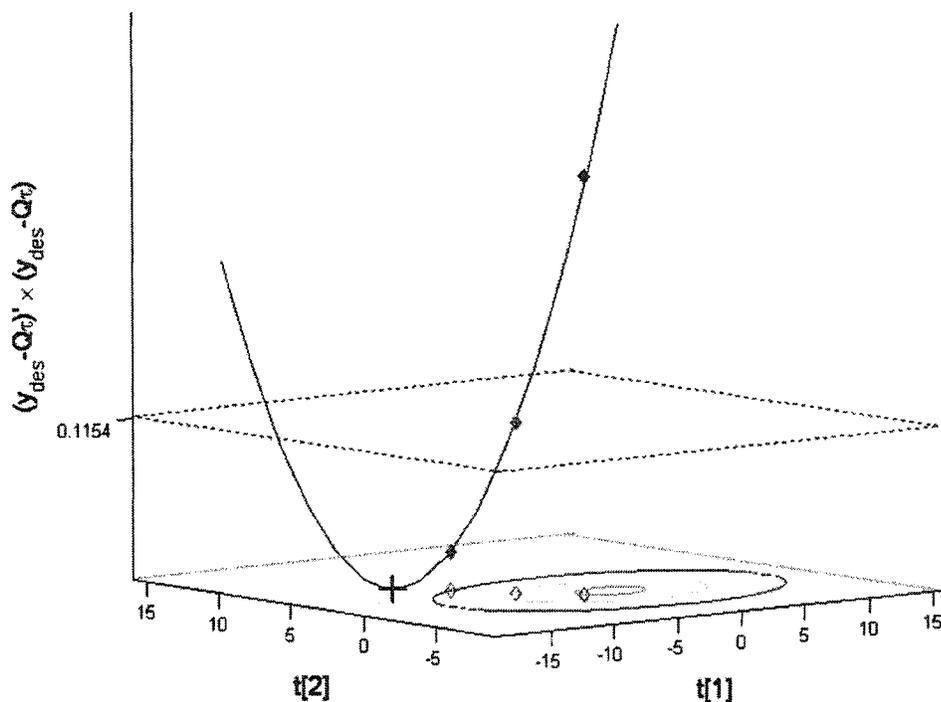
Symbols in legend correspond to the points in Figure 4.8

From an operational point of view, there is no doubt that the solution obtained with soft-constraints along the pseudo-null space is the best solution of all. It falls between the solutions obtained by following either of the univariate null-spaces, and the difference among these (see Fig. 4.9) is very small.

It has been established that either of the univariate null-spaces will keep the prediction of its corresponding  $\mathbf{y}$  unchanged. If these univariate null-spaces do not overlap into a multivariate null-space, the resultant pseudo-null space is the direction of minimum change in the multivariate prediction of  $\mathbf{Y}$ . This implies that: as the chosen solution moves along this pseudo-null space and away from the analytical one, the SPE between  $\hat{\mathbf{y}}_{des}$  and  $\mathbf{y}_{des}$  will still increase. This SPE might become considerable compared with the SPE of the observations used to build the PLS model. The 95% confidence limit on the SPE for the  $\mathbf{Y}$  space from the PLS model is used in Figure 4.10 to contrast the increasing SPE of the solution obtained as this solution moves along the null space. This comparison is not entirely valid since the  $\mathbf{Y}$  used to build the PLS model includes measurement uncertainty, and  $\mathbf{y}_{des}$  does not. The marked points along the parabolic curve are the solutions along the pseudo-null space illustrated in Figure 4.8.

As illustrated in Figure 4.10, an alternative solution along the pseudo-null space will give a SPE for  $\mathbf{y}_{des}$  which is different from zero and above the SPE observed in the training data set. This will introduce doubts about the feasibility of the alternative solution (it might not yield  $\mathbf{y}_{des}$  in the final product).

In this situation there is a tradeoff between the alternative solution along the pseudo null-space (which is close to operational practices already experienced but with  $SPE \neq 0$ ), and the analytical solution (which is far from known operational practices and with  $SPE=0$ ).

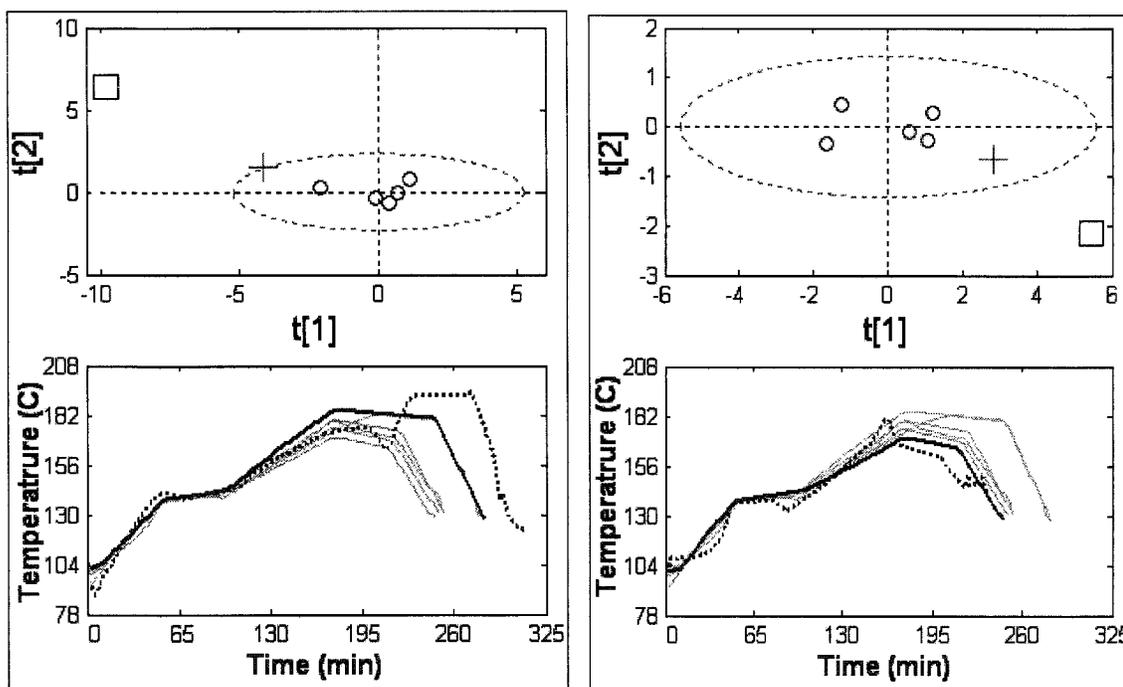


**Figure 4.10 Combined SPE for solutions along the pseudo-null space in the design of grade A for the pilot digester, dashed line is 95% confidence limit from the model (symbols as in Fig. 4.8)**

In practice, the final decision will come from the analysis of the estimated solutions and how far is  $\hat{y}_{des}$  from  $y_{des}$ . If the difference between  $\hat{y}_{des}$  and  $y_{des}$  is within tolerance, then the solution along the pseudo null-space with minimal Mahalanobis distance is the best.

It is important to point out that using univariate constraints (or bounds) on the  $\mathbf{X}$  space (which might be another way to anchor the solution) will not solve the problem, and the estimates will not be better (Fig. 4.11) since these constraints will break the correlations in the data.

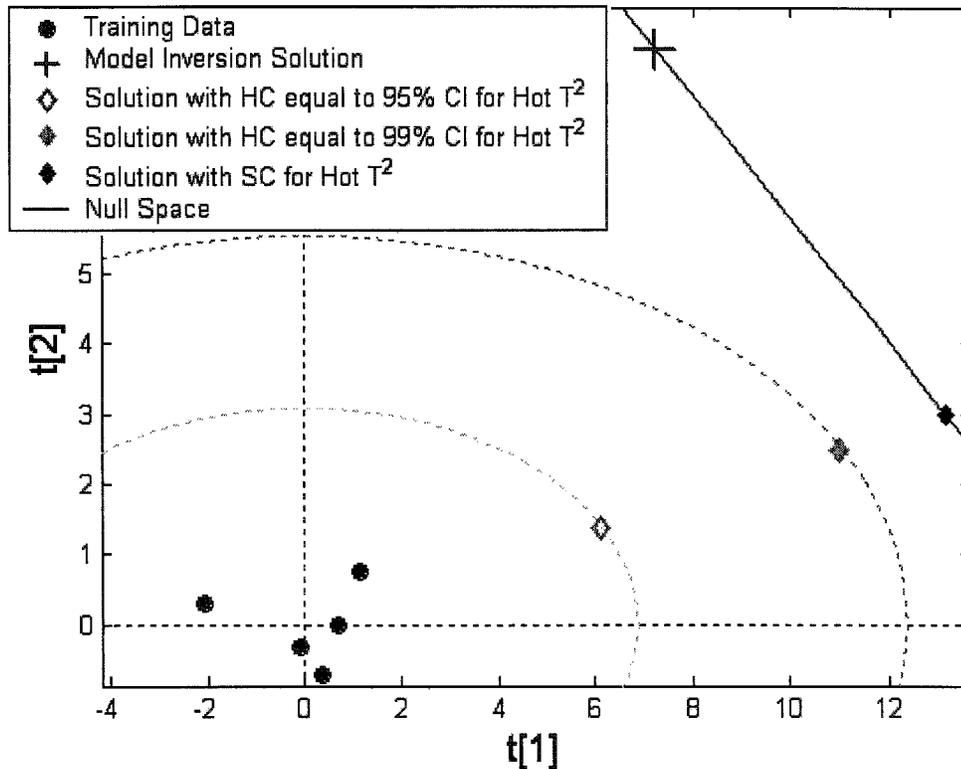
The estimates in Figure 4.11 are obtained when the initial temperature is bounded within the range in the training data (since this is ambient temperature) and the aligned trajectories are bounded to be within 2.5 standard deviations from the mean trajectories (total range of 5 standard deviations) throughout the rest of the batch run.



**Figure 4.11** Score Plots ( $\circ$  model,  $\square$   $\tau_{new}$ ,  $+$   $\tau_{xnew}$ ) and Design Estimates (dotted line), known solution (solid black) and model building grades (gray lines) for grades A (left) and F (right) in the pilot batch digester, imposing bounds on the solution

In the previous exercise, the null space was oriented in such way that alternative solutions, closer to the origin, could be found (Fig. 4.8). However this may not be possible if the null-space (or pseudo-null space) happens not to be oriented towards the origin of the score space (Fig. 4.12).

In such case the solutions with soft constraints on the Hotelling's  $T^2$  will be far from the origin as well. Furthermore, any solution with hard constraints will have a large SPE (since it is moving away from the direction of minimal change for  $\hat{y}_{des}$ ).



**Figure 4.12 Null space oriented away from the origin**

For batch processes particularly, this could be problematic since it was shown that when the score fell outside confidence intervals (Fig. 4.6), the predicted trajectories included very unusual structural characteristics; in spite of the prediction being completely on the plane (SPE equal to 0). This behavior of the predictions suggests that there is certain structure about the trends in the trajectories that is not being entirely captured by the multi-way model, a solution to this problem is suggested in the following Section.

#### 4.4. Model Augmentation with Derivatives

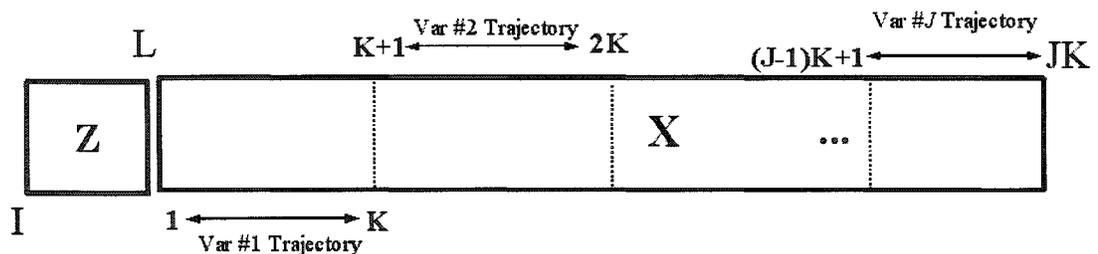
In Section 4.3, it was shown that some unconstrained design estimates exhibited undesired structural characteristics, such as drastic changes in curvature and “jumps”. These undesirable solutions occurred when the solution in the score space lay well outside the 95% confidence region of the training data. The inclusion of multivariate constraints (in the Hotelling’s  $T^2$  of the estimate) was seen to correct the problem and is much more effective than the inclusion of univariate constraints in the process trajectories (bounds) in pulling the solution towards the origin.

From the above discussion one might conclude that the multi-way PLS model for the batch system lacked some structural information about the trend of the trajectories in order for it to be incorporated into the design estimates. It might be desirable then, to augment the model to include trend structure, such as the derivatives of the trajectories. Such additional information in the PLS model on the normal behavior of the derivatives might lead to design solutions with acceptable trend and curvature behavior even without introducing constraints. The reason for this is that by inverting the augmented PLS model, the derivatives of the solution would also be forced to be consistent with past operation performance (i.e. constrained to lie on the model plane)

Information about the trends is contained in the derivative of the trajectories of the process variables with respect to its evolution index (EI). Mean centering the batch data  $\mathbf{X}$  matrix (Fig. 2.1) as suggested by Nomikos and MacGregor (1994) removes the major non-linear component of the batch trajectories. If the derivatives are calculated with the raw batch data, then part of the trend information contained in the derivatives is related to this nonlinear component that is being removed from the trajectories. Mean-centering and auto-scaling these raw-trajectory-based derivatives will further distort the trend information. Because of this, the derivatives of the mean-centered and scaled trajectories are the ones to be incorporated into the model.

As shown in Chapter 2, when a set of batch trajectories is synchronized (or aligned) the time itself becomes an important process variable to be included in the analysis (see Fig. 2.11, Section 2.5.2). The indicator variable chosen to align the rest of the variables against becomes a variable in the system which will be known a priori and will not change for all batches. For example, if the trajectories are aligned against conversion (conversion becomes the evolution index), then the conversion trajectory is a known vector for all batches. The fact that this evolution index (EI) is a known vector allows the derivatives of the trajectories (with respect to this EI) to be expressed as a linear function of these.

For a data set consisting of  $I$  batches, with  $L$  initial conditions, and  $J$  variables being measured  $K$  times during the batch, the unfolded X matrix can be re-ordered as shown in Figure 4.13,



**Figure 4.13 Data Structure and re-ordered X to be used in the augmented model**

The derivatives of the trajectories ( $X$ ) with respect to their evolution index, can then be expressed as (Eq. 4.23), where  $D$  is a  $JK \times JK$  matrix; modifying  $D$  by adding  $L$  columns of zeros ( $D$  modified is referred to as  $D_A$ ) allows the derivative to be a function of the augmented  $[Z X]$  matrix (Eq. 4.24),  $D_A$  is a  $JK \times (L+JK)$  matrix (Appendix 3).

$$\frac{d(X)}{d ei} = \mathbf{X}D^T \tag{4.23}$$

$$\frac{d(\mathbf{X})}{d \text{ ei}} = [\mathbf{Z} \ \mathbf{X}] \mathbf{D}_A^T \quad (4.24)$$

The reordering of  $\mathbf{X}$  is not essential, but simplifies the construction of  $\mathbf{D}$  and  $\mathbf{D}_A$ . For this particular application, these matrices are built using numerical derivatives as shown in Appendix 3.

The steps to be taken now are: to build a PLS model using  $\mathbf{Z}$ ,  $\mathbf{X}$ ,  $[\mathbf{Z} \ \mathbf{X}] \mathbf{D}_A^T$  and  $\mathbf{Y}$ , and then re-formulate the optimization problem to use this augmented PLS model; estimate the new operating conditions  $[\mathbf{z}_{\text{new}}^T \ \mathbf{x}_{\text{new}}^T]$ , and illustrate how the estimates compare with the ones obtained in past Sections.

#### 4.4.1 Building the augmented multi-way PLS Model

The multi-way PLS model to be built has to incorporate three different types of data into the  $\mathbf{X}$  matrix: the initial conditions, the trajectories and the derivatives of the trajectories. The best way to balance these three types of data is to scale each block of data to unit variance, as in a multi-block PLS model (Westerhuis, Kourti, and MacGregor, 1998). This implies equal importance of each block in the analysis. However, this scaling is left to the discretion of the user.

To achieve this, first the  $\mathbf{Z}$  and  $\mathbf{X}$  matrices are mean centered and block-scaled to unit variance as shown in (Westerhuis, Kourti, and MacGregor, 1998), then the matrix of derivatives with respect to the EI is computed (with Eq. 4.24 and referred to as  $\mathbf{dX}$ ) and scaled by the factor in Eq. 4.25 for  $\mathbf{dX}$  to have block-unit variance as well.

$$\sqrt{\frac{1}{\sum_{j=1}^{JK} s_{dX,j}^2}} \quad \text{where } s_{dX,j}^2 \text{ is the variance of the column } j \text{ of } \mathbf{dX} \quad (4.25)$$

Further, the weight of each block can be manipulated if necessary to improve the predictability of  $\mathbf{Y}$ . Now, the PLS model between the augmented matrix  $[\mathbf{Z} \ \mathbf{X} \ \mathbf{dX}]$  and  $\mathbf{Y}$  can be fitted.

This augmented model will capture the information about the structure of the trajectories and the structure in the derivative of the trajectories. Using this multi-way augmented PLS model with the product design formulations in Eqs. 4.7 and 4.13 is a powerful way to simultaneously incorporate multivariate constraints on the trajectories and the curvature structure of the trajectories of the estimated design ( $\mathbf{x}_{\text{new}}$ ) to yield product with a certain desired quality ( $\mathbf{y}_{\text{des}}$ ).

#### 4.4.2 Re-Formulating the Objective Function

Once the  $\boldsymbol{\tau}_{\text{new}}$  is estimated (as described in Section 4.3.1), the reconstruction of the initial conditions and the trajectories is re-formulated to include the derivatives of the trajectories. Consider the augmented column vector  $\mathbf{xa}_{\text{new}}$  (Eq. 4.26), the derivative of the trajectories ( $\mathbf{x}_{\text{new}}$ ) with respect to EI can be expressed as a function of  $\mathbf{xa}_{\text{new}}$  and  $\mathbf{D}_A$  (Eq. 4.27). If the augmented PLS model is used to build  $\mathbf{H}$  and  $\mathbf{f}$  in Eq. 4.13, and  $\mathbf{x}_{\text{new}}$  in this equation is replaced by a new column vector containing  $\mathbf{xa}_{\text{new}}$  and the derivative of  $\mathbf{xa}_{\text{new}}$  with respect to EI, Eq. 4.13 becomes Eq. 4.28.

$$\mathbf{xa}_{\text{new}} = \begin{bmatrix} \mathbf{z}_{\text{new}} \\ \mathbf{x}_{\text{new}} \end{bmatrix} \quad (4.26)$$

$$\frac{d \mathbf{xa}_{\text{new}}}{d ei} = \mathbf{D}_A \mathbf{xa}_{\text{new}} \quad (4.27)$$

$$\min_{\mathbf{xa}_{new}} \left\{ \frac{1}{2} \left[ \mathbf{xa}_{new}^T \left[ \mathbf{D}_A \mathbf{xa}_{new} \right]^T \right] \mathbf{H}_a \begin{bmatrix} \mathbf{xa}_{new} \\ \mathbf{D}_A \mathbf{xa}_{new} \end{bmatrix} + \mathbf{f}_a^T \begin{bmatrix} \mathbf{xa}_{new} \\ \mathbf{D}_A \mathbf{xa}_{new} \end{bmatrix} \right\} \quad (4.28)$$

In Eq. 4.28,  $\mathbf{H}_a$  and  $\mathbf{f}_a$  are constructed as in Eq. 4.16 but using the loadings from the augmented multi-way PLS model;  $\mathbf{H}_a$  is a  $(2JK+L) \times (2JK+L)$  matrix, and  $\mathbf{f}_a$  has  $2JK+L$  elements. The weighting matrix  $\Lambda$  is constructed in the same way (Eqs. 4.17 to 4.19) but using the augmented matrix  $[\mathbf{Z} \ \mathbf{X} \ \mathbf{dX}]$ .  $\mathbf{G}_2$  continues to be a diagonal matrix with ones, and as many elements in the diagonal as components in the augmented PLS model. This equation (4.28) can further be manipulated to be reduced to a standard QP form.  $\mathbf{H}_a$  can be segmented into 4 sub-matrices  $\mathbf{H}_1$ ,  $\mathbf{H}_2$ ,  $\mathbf{H}_{12}$ , and  $\mathbf{H}_{21}$  as shown in Figure. 4.14, and  $\mathbf{f}_a$  can be segmented also into  $\mathbf{f}_1$  and  $\mathbf{f}_2$ , with  $\mathbf{f}_1$  being the first  $JK+L$  elements of  $\mathbf{f}_a$ , and  $\mathbf{f}_2$  being the rest  $JK$  elements of  $\mathbf{f}_a$ .

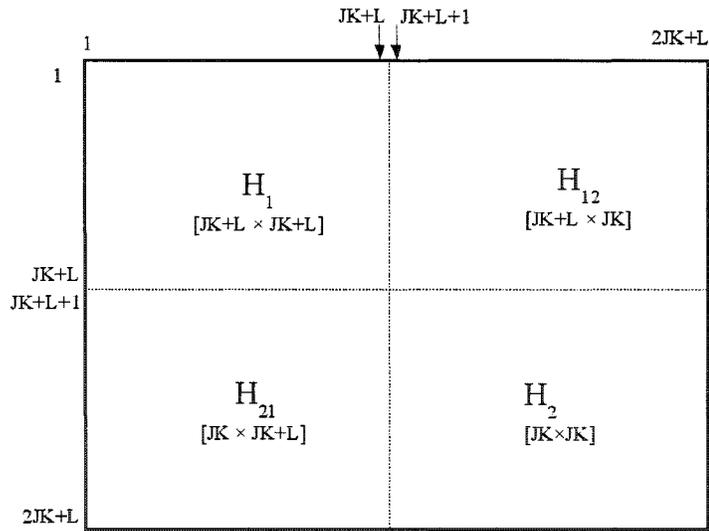


Figure 4.14 Segmentation of  $\mathbf{H}_a$

By doing these segmentations, Eq. 4.28 can be manipulated algebraically to become Eq. 4.29, which has a standard quadratic programming form.

$$\min_{\mathbf{xa}_{\text{new}}} \left\{ \frac{1}{2} \mathbf{xa}_{\text{new}}^T \Omega \mathbf{xa}_{\text{new}} + \phi^T \mathbf{xa}_{\text{new}} \right\}$$

where

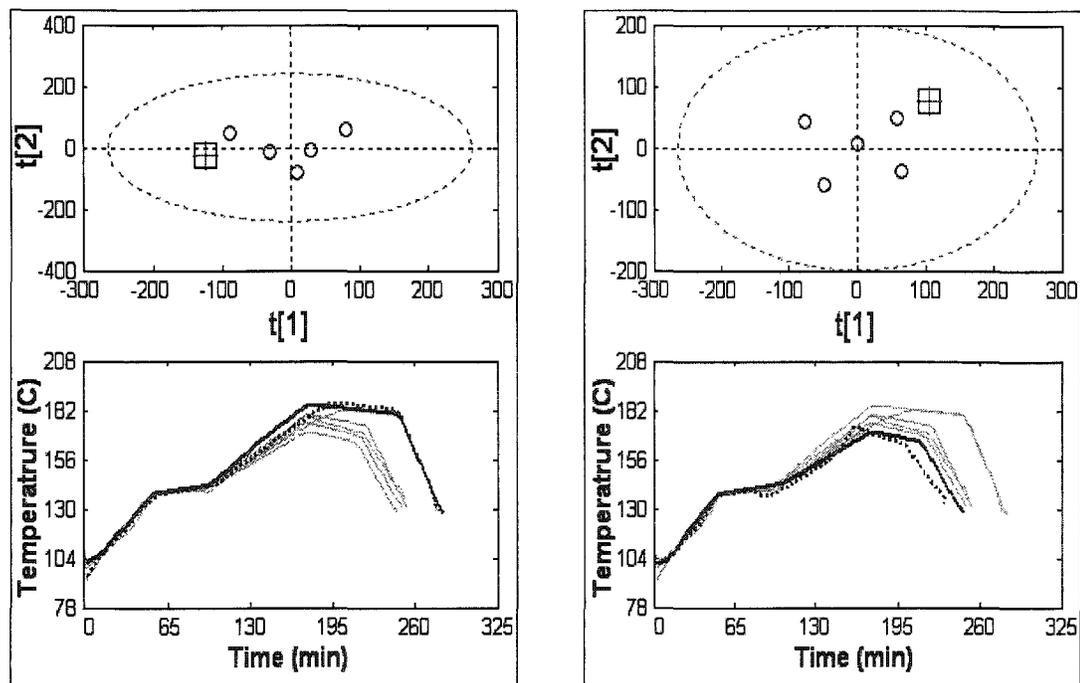
$$\Omega = \mathbf{H}_1 + \mathbf{D}_A^T \mathbf{H}_{21} + \mathbf{H}_{12} \mathbf{D}_A + \mathbf{D}_A^T \mathbf{H}_2 \mathbf{D}_A$$

$$\phi^T = \mathbf{f}_1^T + \mathbf{f}_2^T \mathbf{D}_A$$

(4.29)

#### 4.4.3 Application to industrial Pulp Digester

The results for the design of the temperature trajectories for grades A and F for the pilot batch digester; obtained by building an augmented multi-way PLS model and solving the optimization problem in Eqs. 4.7 and 4.29 (with no constraints) are illustrated in Figure 4.15.



**Figure 4.15** Score Plots ( $\circ$  model,  $\square$   $\tau_{\text{new}}$ ,  $+\tau_{\text{xnew}}$ ) and Design Estimates (dotted line), known solution (solid black) and model building grades (gray lines) for grades A (left) and F (right) in the pilot batch digester using augmented PLS model

In Figure 4.15 notice *a*) the lack of drastic changes in curvature in the trajectories (compare these with Figs 4.6 bottom and 4.11), and *b*) the scores are no longer outside the boundaries of the normal operating region, and thus, the Hotelling's  $T^2$  constraint is no longer required for these grades.

The inclusion of the derivatives with respect to the EI into the multi-way PLS model improves the solution without the need of bounds or extra constraints because the trend structure of the trajectories is being modeled. And since the solution is close to the normal operation region it is no longer essential to explore the null-space to find a realistic solution. The optimization formulation allows for the SPE to be different than zero; the SPE for both cases is practically zero because no constraints were imposed on the  $\mathbf{X}$  space.

In the next Section, the objective function used to estimate the batch trajectories (Eq. 4.29) is further expanded to include an optimization criterion into the design. The extended formulation is illustrated by estimating the trajectories with minimal batch duration.

#### **4.5. Optimal Trajectory Design - Minimize Batch Length**

In the previous examples, the designs obtained with the multi-way augmented PLS models show SPE's virtually equal to zero. The general idea is now to tolerate a certain amount of model-mismatch and moderate extrapolations in the score space to allow some exploratory designs. These new exploratory designs will contain distortions in the trajectories to achieve some optimal criteria, as long as the SPE is kept within a conservative limit. Using the augmented multi-way PLS model approach should yield acceptable distortions to the trajectories since the curvature and trajectory structure from past historic data is still being passed to the design estimate.

Designing batches with shorter durations is always a welcome improvement to any batch process; the economical benefits of such an improvement are obvious.

The effectiveness of any data-based optimization approach depends on the information content of the data, and the power of the method to extract such information from the data (i.e. poor methods like OLS require very well conditioned data; and powerful methods like PLS can handle ill-conditioned data), but at the end, the information must be somehow contained in the data. To optimize the design of a batch trajectory with respect to time requires a data set with batches having different lengths. These batches should be aligned (or synchronized), and at the end, the *time usage* trajectory must be added to the data set (as shown in Section 2.5.2), since this is precisely the trajectory which will be minimized in the new design.

The objective function (Eq. 4.9) can be expanded (Eq. 4.30) to consider a weight imposed to the time usage trajectory, such that the new design must: *a*) have scores ( $\boldsymbol{\tau}_{\text{xnew}}$ ) close to those desired ( $\boldsymbol{\tau}_{\text{new}}$ ) in the quadratic space (term 1), *b*) be at a minimal orthogonal distance from the model plane (term 2), and *c*) have as low values as possible for certain trajectories (e.g. time usage) (term 3). This formulation is nothing but the transformation of the hard constraints in the reconstruction of  $\mathbf{x}_{\text{new}}$  given by equations 4.13 and 4.14, into soft constraints.

$$\min_{\mathbf{x}_{\text{new}}} \left\{ (\boldsymbol{\tau}_{\text{xnew}} - \boldsymbol{\tau}_{\text{new}})^T \mathbf{G}_2 (\boldsymbol{\tau}_{\text{xnew}} - \boldsymbol{\tau}_{\text{new}}) + (\mathbf{x}_{\text{new}} - \hat{\mathbf{x}}_{\text{new}})^T \boldsymbol{\Lambda} (\mathbf{x}_{\text{new}} - \hat{\mathbf{x}}_{\text{new}}) + \boldsymbol{\eta} \mathbf{x}_{\text{new}} \right\} \quad (4.30)$$

This objective function can further be manipulated to include initial conditions (Eq. 4.26) and have a standard QP form. The loadings to build  $\mathbf{H}$  and  $\mathbf{f}$  should be those of a PLS model where the initial conditions and the variable trajectory blocks are considered in the  $\mathbf{X}$  space (e.g. by scaling each block to unit variance and using this block scaled  $[\mathbf{Z} \ \mathbf{X}]$  as  $\mathbf{X}$  in the PLS model).

$$\min_{\mathbf{xa}_{\text{new}}} \left\{ \frac{1}{2} \mathbf{xa}_{\text{new}}^T \mathbf{H}_0 \mathbf{xa}_{\text{new}} + \mathbf{fo}^T \mathbf{xa}_{\text{new}} \right\}$$

where

$$\mathbf{H}_0 = \mathbf{W}^* \mathbf{G}_2 \mathbf{W}^{*T} + \Lambda - \Lambda \mathbf{P} \mathbf{W}^{*T} - \mathbf{W}^* \mathbf{P}^T \Lambda + \mathbf{W}^* \mathbf{P}^T \Lambda \mathbf{P} \mathbf{W}^{*T}$$

$$\mathbf{fo}^T = \boldsymbol{\eta} - \boldsymbol{\tau}_{\text{new}}^T \mathbf{G}_2 \mathbf{W}^{*T}$$

(4.31)

Finally, the derivatives can be integrated into the formulation by using  $\mathbf{D}_\Lambda$  (Appendix 3),  $\mathbf{H}_0$  and  $\mathbf{fo}$  from Eq. 4.31 and segmenting these two last matrices, as  $\mathbf{H}_a$  and  $\mathbf{f}_a$  were, to compute  $\boldsymbol{\Omega}$  and  $\phi$  to be used in Eq. 4.29.

If  $\boldsymbol{\eta}$  contains positive values in those elements corresponding to the time usage trajectory, then the solution to this final formulation should provide a design that will yield the desired quality, with a low usage of time, while keeping a conservative Mahalanobis distance from the existing training data, and a conservative orthogonal distance to the model plane. This approach is illustrated in the following Section with the use of an industrial data set from a polymerization process.

#### 4.6. Industrial Example: Optimal Trajectory Estimation for an Emulsion Polymerization Process

The process consists of a semi-batch reactor, where an emulsion polymerization is executed. The data set, provided by Air Products and Chemicals Inc. consists of 55 batches; for each batch there are 11 initial conditions ( $z_1, \dots, z_{11}$ ), 9 variable trajectories (conversion,  $x_1, \dots, x_8$ ), and 13 quality characteristics ( $y_1, \dots, y_{13}$ ). The trajectories for such process are illustrated in Figure 4.16. Prior to the alignment, a low-pass median filter is applied to smooth out some noisy peaks in the data. The trajectories are then aligned with respect to the conversion (sampling every 0.1% for the first percent, and every 0.5% for

the rest of the batch, total 208 samples per batch); the time usage trajectory is computed as shown in Section 2.5.2, and this trajectory is added to the aligned data set (Fig. 4.17).

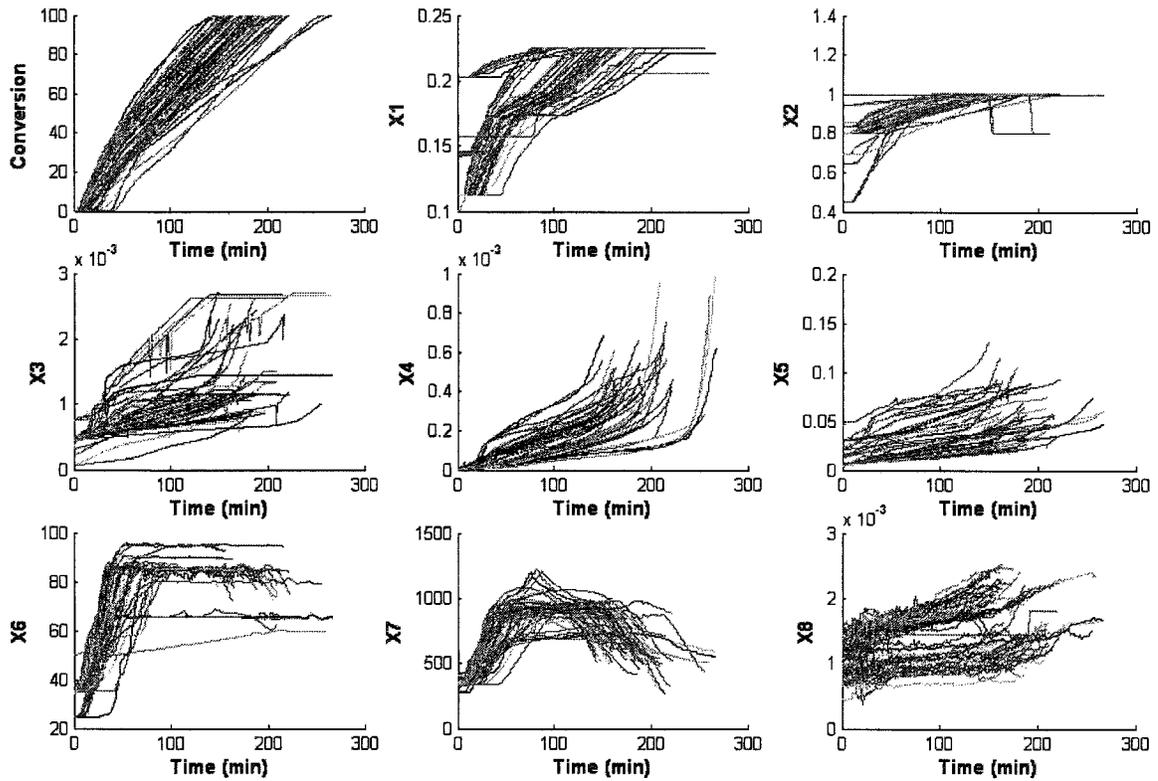
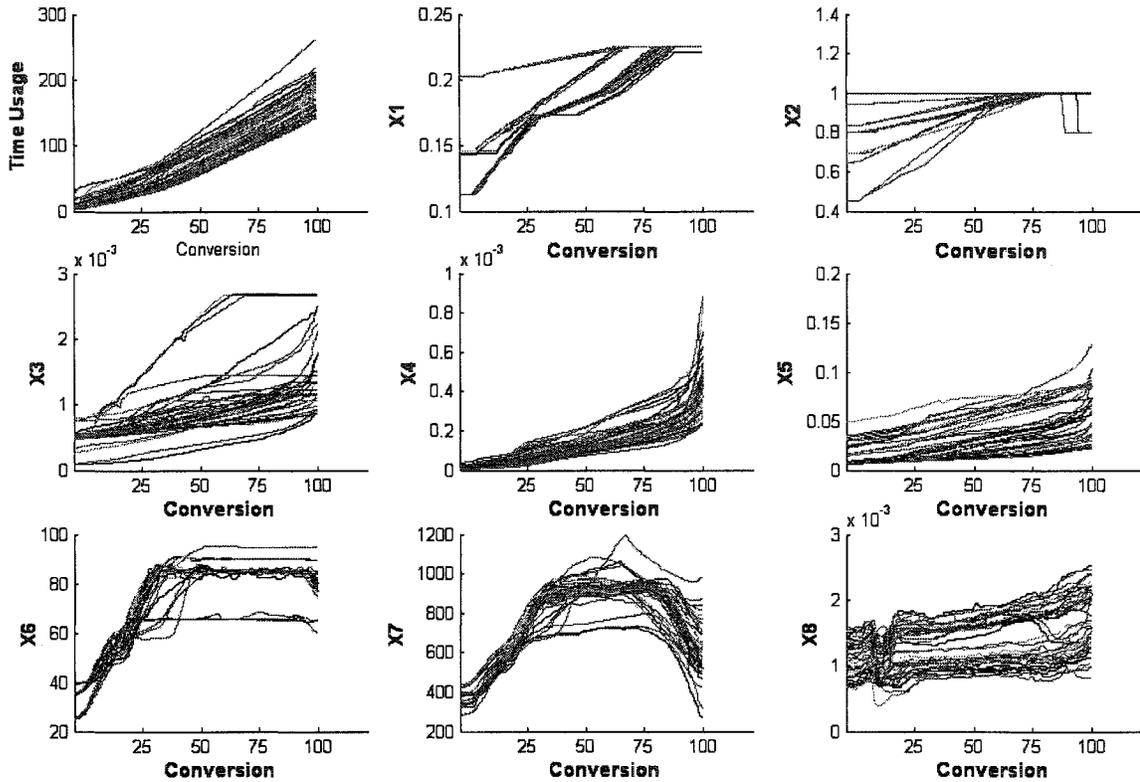


Figure 4.16 Raw Process Trajectories for the Emulsion semi-batch process



**Figure 4.17 Aligned Process Trajectories and Time Usage for the emulsion semi-batch process**

The task now is to design a new product which has to satisfy the following requirements:

$y_1$  to  $y_6$  and  $y_8$  are to be kept in normal range

$y_7 = y_{7des}$  and  $y_9 = y_{9des}$

$y_{10} < y_{10const}$ ,  $y_{11} < y_{11const}$ ,  $y_{12} < y_{12const}$  and  $y_{13} < y_{13const}$

With the minimal possible use of time

These equality and inequality constraints are included into Eq. 4.7b, which is solved to estimate  $\tau_{new}$ . Once the  $\tau_{new}$  is estimated, the trajectory design problem is solved by minimizing Eq. 4.31.

Several optimal trajectories are shown in Figures 4.18 and 4.19. Besides the usage of time, the solutions differ in their orthogonal distance to the plane (Table 4.3) which is still being kept at a very conservative value. Case 1 is the no-constraint design case, case 2 through 5 use the optimization criterion (Eq. 4.30) with increasing weight on time usage (i.e. increasing  $\eta$ ). As the time usage is minimized the orthogonal distance to the plane increases. This is expected, since the structure in the trajectories is being distorted slightly to achieve shorter times. This distortion, however, is a structured distortion which obeys as much as possible the structure captured in the PLS model. This result is very significant, since it means that the deviations from the plane are being controlled and performed in a structured manner, especially in the way the curvature structure is being kept due to the inclusion of the derivatives (since there are no unexpected jumps in the solutions). Figure 4.19 illustrates with more detail how the different variable trajectories are being adjusted, allowing less time usage while maintaining the structure from the augmented multi-way PLS model; and since the structure is being kept the orthogonal distance to the plane is kept minimal. The fact that all the trajectories are being adjusted simultaneously proves that the solution is not just focusing in lowering the time usage trajectory alone, but performing adjustments in the entire solution for it to be consistent with the multivariate correlations among the different time profiles, which are captured in the augmented multi-way PLS model.

Another important fact is that, all the five solutions project to the same point of the latent space,  $\tau_{x_{new}}$ ; and for the five cases, this point of projection in the score space corresponds to the desired one ( $\tau_{x_{new}} = \tau_{new}$ ). This is achieved by adjusting  $G_2$  along with  $\eta$  in equation 4.30, and it is very important to do so; since the primary objective of the optimal design exercise is for the final product to keep the desired quality characteristics in spite of the adjustments done to the variable trajectories (the  $G_2$  weighting matrix has to be incremented along with  $\eta$  to keep  $\tau_{x_{new}} = \tau_{new}$ ). The shape of the solutions plotted against time (and returning conversion as a process variable) is shown in Figure 4.20.

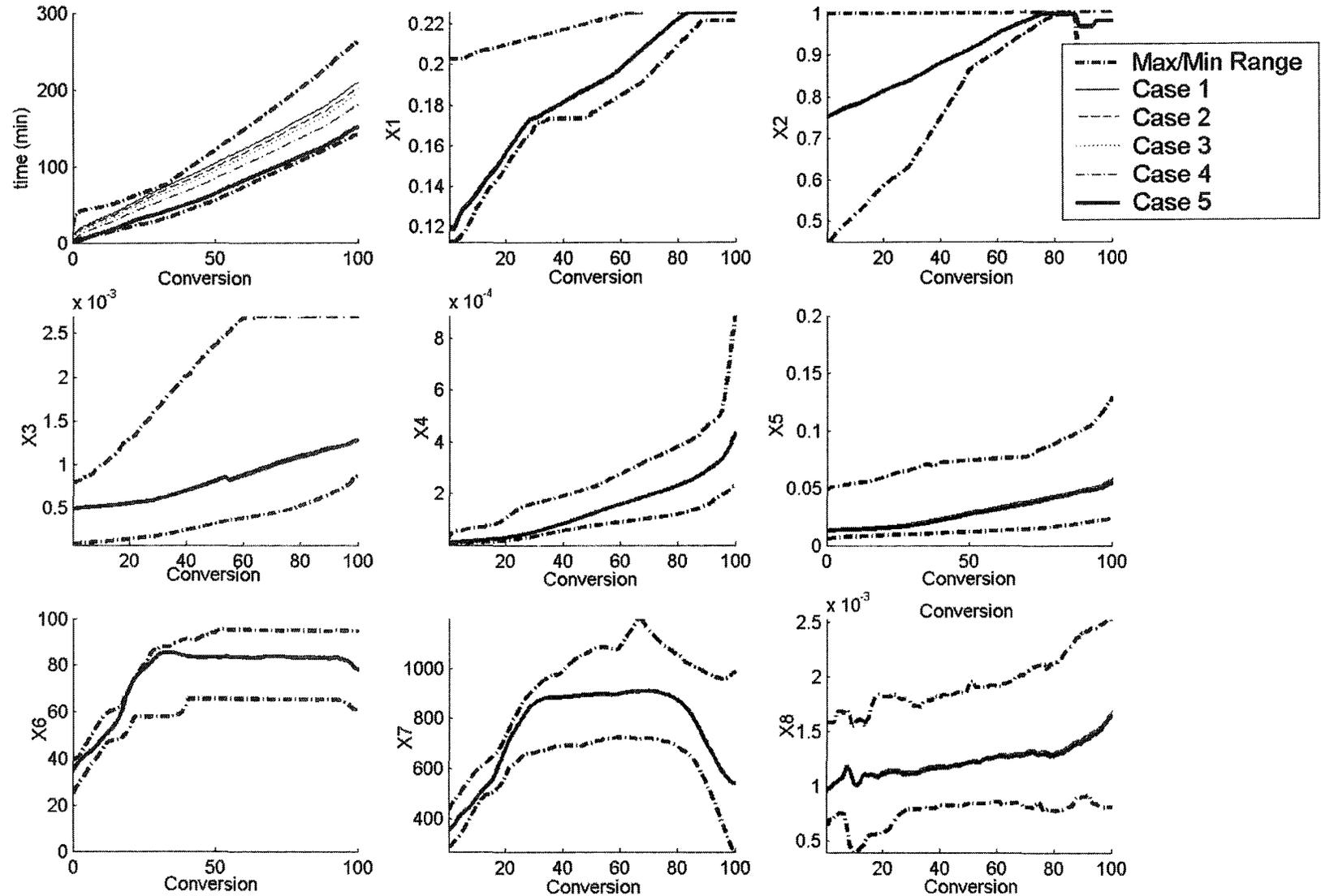


Figure 4.18 Different Optimal Solutions for Different values of  $\eta$

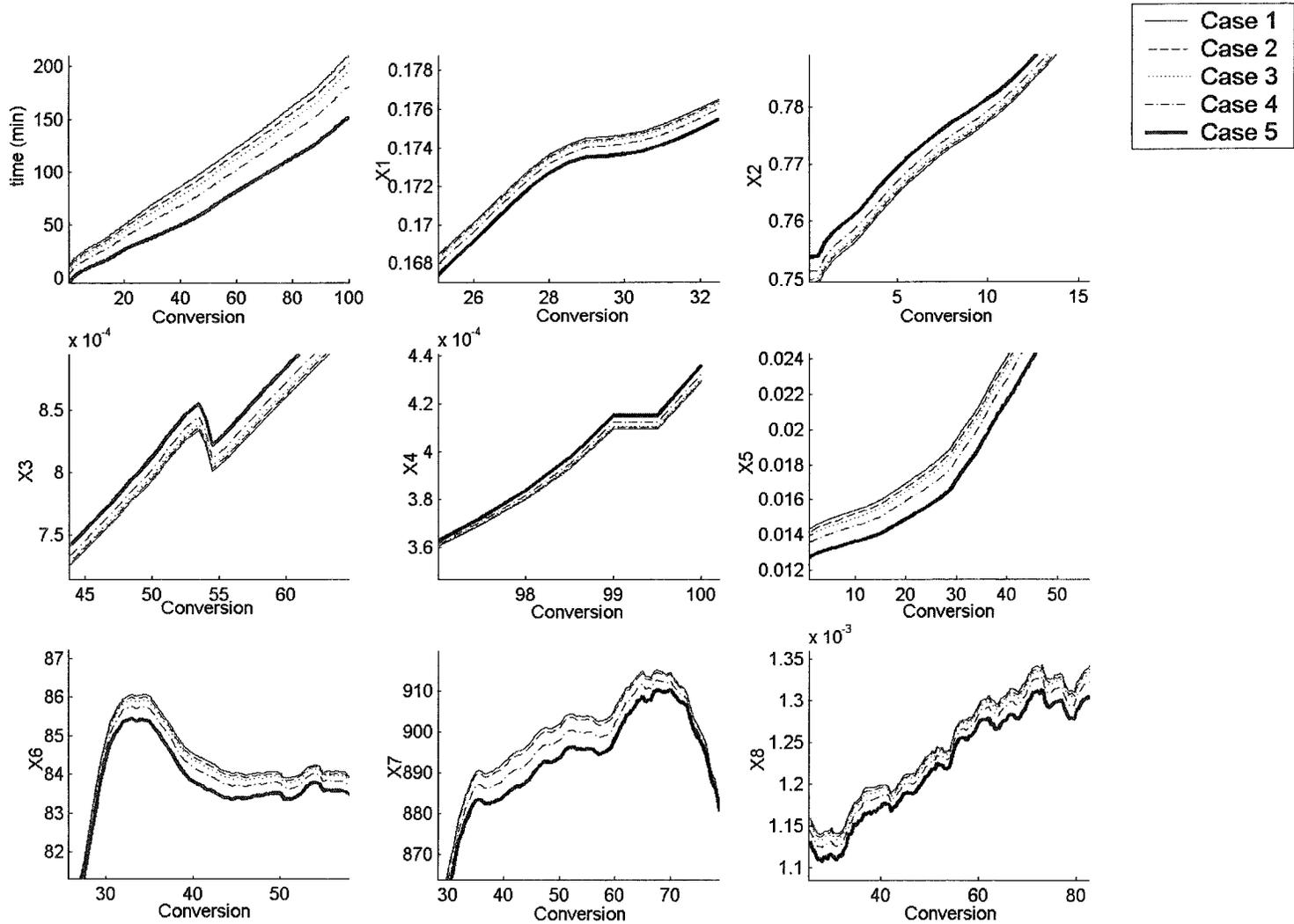


Figure 4.19 Detail of Different Optimal Solutions for Different values of  $\eta$

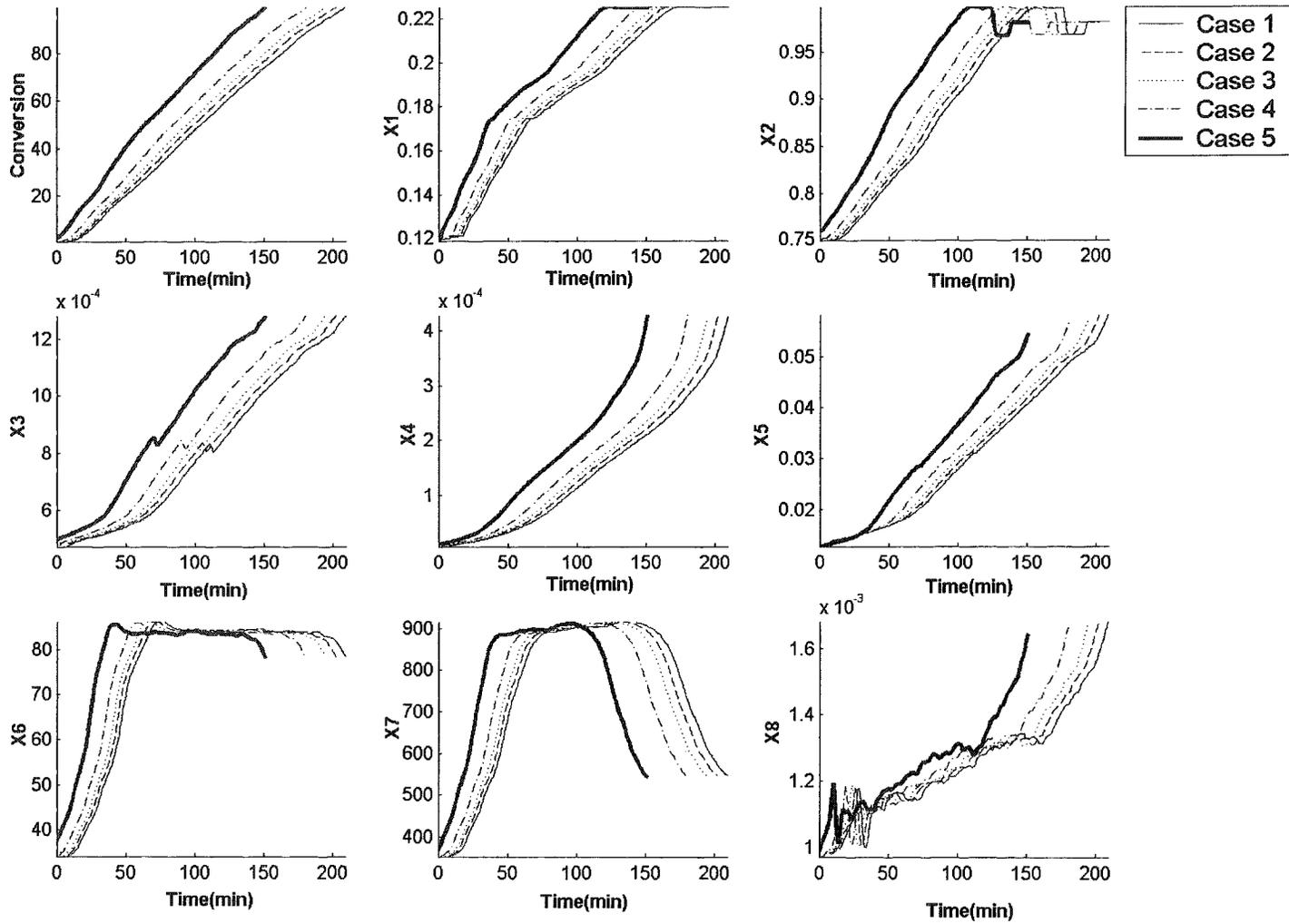


Figure 4.20 Un-aligned Different Optimal Solutions for different values of  $\eta$

Case	SPE	SPE 95% C.I.
1	4.24E-019	6.88
2	0.01	6.88
3	0.04	6.88
4	0.15	6.88
5	0.6	6.88

**Table 4.3 Square prediction error for the five different optimal solutions**

In this illustrative industrial example, the *time usage* (Section 2.5.2) is a key concept in order to find the optimal design. The alignment exercise should uncover the differences in time usage in the data set. The optimal trajectory design method proposed in this work can also satisfy other optimality criteria like a minimal consumption of a raw material; the only factor to adjust are the values given to the  $\eta$  weighting vector. The solution (if close enough to the plane) is guaranteed to have the same correlation structure as the historical data and therefore not to show any abnormal behavior that may turn the design into a non-feasible solution from a practical point of view. Even though in this example multivariate constraints were not required, they may also be imposed on the overall Hotelling's  $T^2$ , or to the Hotelling's  $T^2$  of one of the blocks considered (in which case, this summary statistic must be computed with the complete variance-covariance matrix of the block-scores because these are not necessarily orthogonal).

The selection of the final solution to be implemented may depend on physical constraints of the system or on economical constraints, or may very well be determined by the scheduling problem. The implementation of the final solution should be done with care; the uncertainty in batch systems is time varying as well, and since the estimate for some periods in the trajectories may be more uncertain than for others, it is desirable to have confidence regions for the design estimate. The development of confidence regions for the design estimates is not obvious, and requires careful study since these operational windows must be multivariate (the trajectories are not independent from each other).

In the exercise of designing batch trajectories, the uncertainty in modeling  $Y$  is as important as the uncertainty in modeling  $X$ , both sources of uncertainty should be taken into account in the development of confidence regions for design estimates, in the variable trajectories and in the initial conditions. The construction of uncertainty regions is not considered in this thesis, but is left as an open problem for future research.

#### **4.7. Batch Modeling and Troubleshooting using Augmented Models with Partial Derivatives**

In past Sections, the derivative of the variable trajectories with respect to their evolution index (EI) was included, along with the initial conditions and the trajectories into a multi-block, multi-way PLS model. The addition of the derivatives increased the information about the  $X$  space including extra information about the curvature structure of the trajectories. This became useful when the model was inverted to estimate new trajectories, since the structure of the derivatives was also included in the estimates.

Since all the analysis is already done from a multivariate perspective, then it seems sensible to include the numerical partial derivatives of the trajectories with respect to each other, and examine the effects of doing so in the predictability of  $Y$ . This analysis is important, since the inclusion of extra information in the  $X$  space might yield to overfit in the  $Y$  space. Figure 4.21 illustrates how the  $R^2Y_{cum}$  increases and  $Q^2Y_{cum}$  is kept roughly in the same value for the same number of components, as more derivatives are included into the PLS model (for the emulsion polymerization industrial data set used in Section 4.5). Even though the  $R^2Y$  increases (from 68% to 88%) not much is gained in the prediction of  $Y$ . The power of using augmented models with derivatives is (as illustrated in the past Sections) the fact that it brings more structural information about the  $X$ . This is illustrated in Fig. 4.22, where the Future Prediction Sum of Squares (Eq. 3.15) with and without the model augmentation with the derivatives is plotted for all variables in the emulsion polymerization data. In general, the FPRESS decreases with the

use of derivatives in the model, which means that the accuracy of prediction for the  $X$  improves.

The numerical partial derivative is a non-linear function of the trajectory; a very rough partial derivative can be linear if the local difference is taken into account (numerical derivative with one point), however this approximation will be very poor due to the presence of noise in the measurements. In this work, the numerical derivatives were calculated with 3, 4, and 5 (for the partial ones) points to smooth out the presence of noise in the data. This nonlinear relationship between the trajectories and their numerical partial derivatives makes it difficult to include these derivatives into the objective function to be minimized in the design exercise (as it was done in Eq. 4.28), turning the linear optimization problem into a non-linear one.

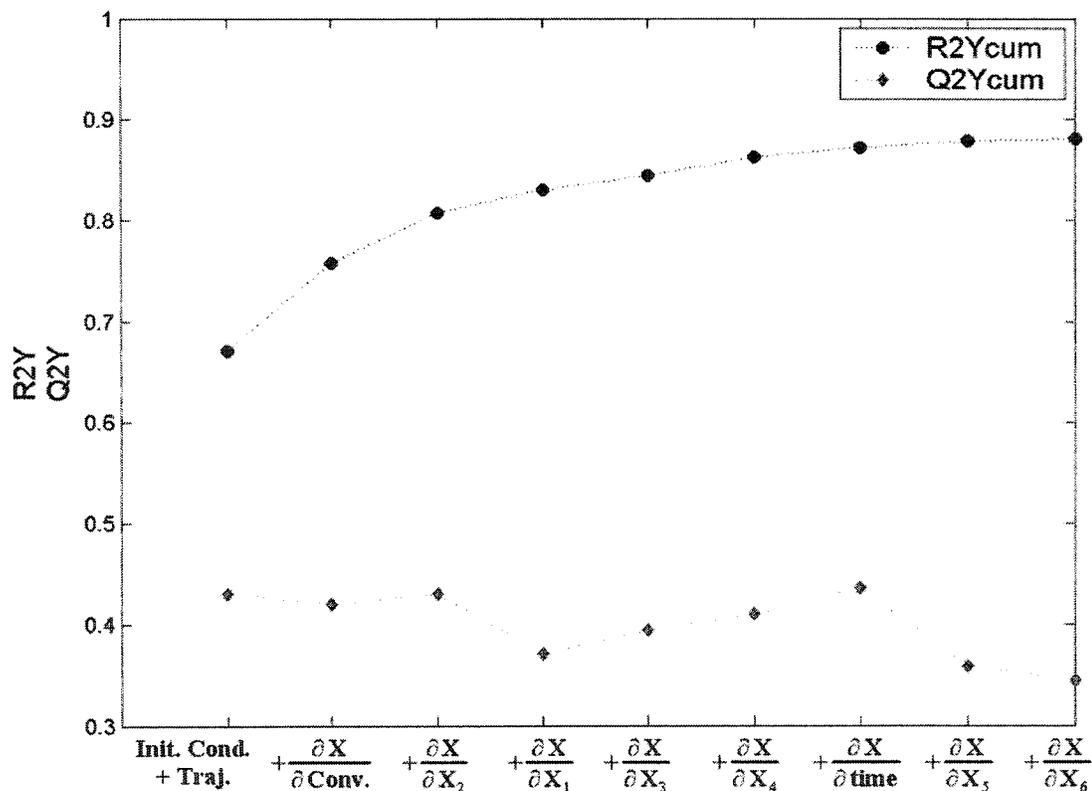


Figure 4.21 R2Y and Q2Y as partial derivatives are included in the model

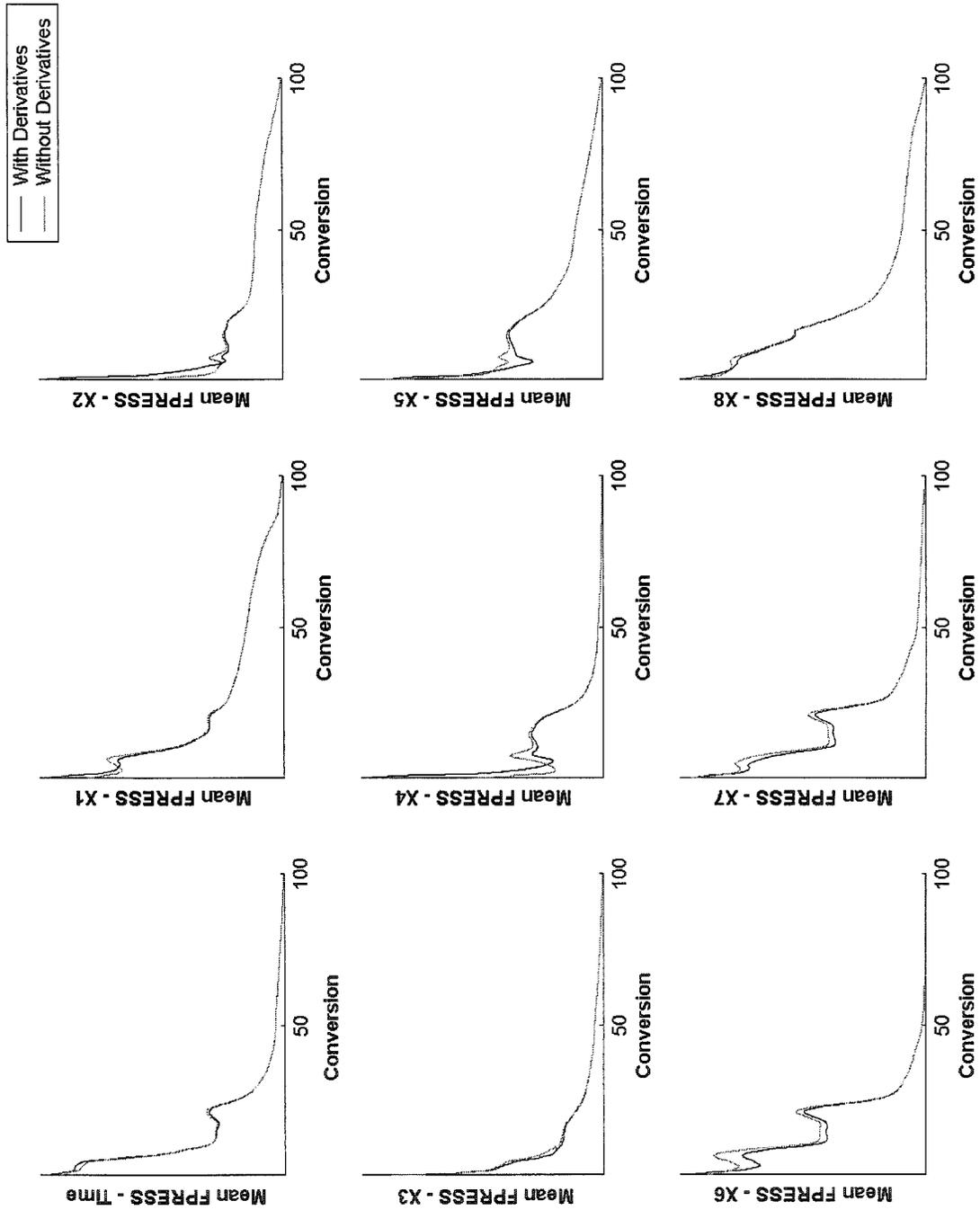


Figure 4.22 FPRESS for all variables in the emulsion polymerization data

#### 4.8. Conclusions and Future Work

The design methodology proposed by Jaeckle and MacGregor (1998) was applied in the design of complete batch trajectories along with their initial conditions. In certain cases it is possible to have more than one solution to the problem. Several ways to constraint the solution were presented and the final estimates appear to be very close to the known solution from the original data.

A parallel new design approach is also proposed to estimate the operating conditions of a batch process. The new approach uses historical data and an augmented multi-way PLS model, within an optimization framework. The use of multi-way PLS models augmented with the derivatives of the trajectories with respect to its evolution index is a powerful technique in the design of batch trajectories since it allows for the trend structure to be taken into account, and the constraints considered in the former technique might not be necessary. Derivative augmented multi-way PLS models could also bring in some benefit when the batch control problem is solved with latent variable regression models, where robustness is crucial and may require more trend structure to be considered into the model.

If required, in any of the two approaches, multivariate constraints (constraining the Hotelling's  $T^2$ ) proved to be much better than univariate constraints (bounds).

The optimization framework proposed, coupled with a good alignment technique, provides a way to compute designs that will meet the *minimal time consumption* optimal criteria. The two industrial cases presented in this work clearly show that, the proposed design and optimization technique is a feasible solution to some of the critical needs in the area, since the techniques suggested here will allow a shorter time to market for a new product, and an increase in throughput as well.

# Chapter 5

## Product Transfer Between Sites using Joint-Y PLS

This chapter presents an extension to the work by Jaeckle and MacGregor (2000b) in solving the product transfer problem. More specifically in addressing the problem of calculating the operational conditions for a plant in order to produce a product of the same quality as the one being produced at another site. The methodology proposed by Jaeckle and MacGregor (2000b) does not consider in the calculations all the data structures involved in the problem and particularly the operating conditions from the *source* plant. The Joint-Y PLS model is presented as an alternative to solve this problem using all the available data.

### 5.1. Introduction

Moving the production of a certain grade of product from one site to another is an increasing need in industry since it results in significant cost reductions in the supply chain. Finding new opportunities in the product-manufacturing network represents a significant opportunity area for cost reductions in the strategic planning exercise. The scale-up of a product from pilot plant to an existing site can also be seen as a product transfer problem. This practice is most important for corporations who strive on the manufacturing of new products such as the specialty chemicals. The successful product transfer in industry requires a considerable amount of time and expertise, and it is a very challenging task. The use of a deterministic model of the plant is a viable solution if such

a model is readily available and it is validated to accurately model the behavior of the process. However, such models are costly, lengthy in their development and they require a degree of high complexity (in order to model all the sources of uncertainty). These factors make a deterministic model prohibitive in many cases.

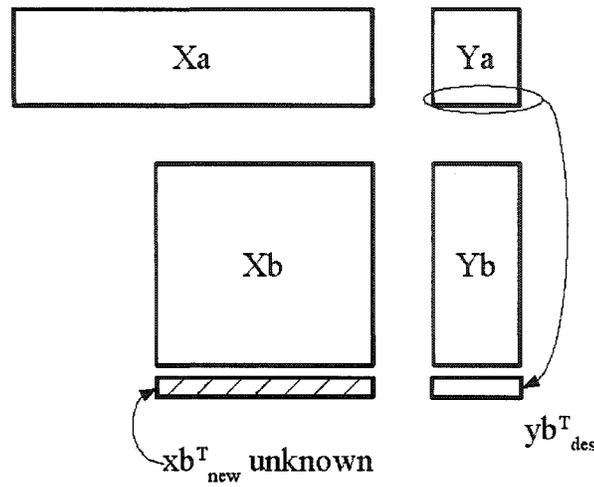
Latent variable empirical models built with historical data have proven their efficiency and robustness in modeling the region in which a process can operate (Garcia, Kourti, and MacGregor, 2003). Such models were used in Chapter 4 to design the required process conditions for a plant to yield a desired set of characteristics in the final product.

Jaeckle and MacGregor (1998; 2000a; and 2000b) presented the concept of the *null space*. In practice this represents an *operational window* which corresponds to a set of allowable process conditions resulting in the same product quality. This concept is later used in a univariate fashion in the work by Zhou *et al.* (1999) to determine the conditions for a bioprocess; in that paper the operational windows are determined by plotting pairs of process variables. However, this approach is limited since the number of pairs needed to include all the variables in an industrial site is very large making it impractical. Furthermore, the full correlation structure will not be taken into account. Chen and Wang (2000) use the same approach but with multivariate operational regions defined by the scores of a PCA model done with historical data from a Fluidized Catalytic Cracking (FCC) unit. In their work contribution plots are used to interrogate the model and complement their engineering knowledge about the unit in order to make grade changes in the process; no mapping or regression is done against the product characteristics. Lakshminarayanan *et al.* (2000) approach the product design problem using the same strategy as in the work by Jaeckle and MacGregor (1998), but with the use of a neural net to introduce non-linearity in the PCA model. Jaeckle and MacGregor (1998) used a nonlinear PLS to account for non linearities as well.

To the author's knowledge there is no other published work trying to solve the product transfer problem beyond the solution proposed by Jaeckle and MacGregor (2000b).

### 5.1.1. Product Transfer using Model Inversion

The methodology presented by Jaeckle and MacGregor (2000b) to transfer product from one site to another is an extension to the product design methodology (see section 4.1). In this problem the "desired" grade has already been produced at another (*source*) site and the operating conditions in the *target* plant are the ones to be estimated (E.g. It is desired to produce in plant B one of the grades already produced in plant A, Fig. 5.1). The scenario now includes more data structures than the ones available in the product design case (where there is only one  $\mathbf{X}$  matrix, one  $\mathbf{Y}$  matrix and the  $\mathbf{y}_{\text{des}}$  vector). Here we have a case (Fig. 5.1) where the number and type of process variables describing the operating conditions in the two plants maybe different; the number of grades produced in the two plants may differ too. Such data structures are easy to find in any scale-up scenario, where the number of variables measured in the pilot plant is not the same as the ones being measured in the full scale equipment. The number of observations differs as well since the pilot plant might have produced a wider variety of grades than the industrial scale one.



**Figure 5.1 Product transfer from plant A to plant B**

In these matrices ( $X_a$ ,  $Y_a$ ,  $Y_b$  and  $X_b$ ) each multivariate observation (row) corresponds to a grade produced in that site. Each of these rows can be obtained as the average from several normal operating conditions (NOC) taken from historical data on the same grade. This consideration is important because the correlation structure of interest in the product transfer (and also in the product design case) is the one defined by the purposeful changes in the operating conditions to achieve a difference among grades. This is different from the correlation structure modeled for monitoring which is defined by the common unwanted disturbances entering the process during production of any one grade. Taking the average conditions from several samples in the plant for a certain grade should work as a low-pass filter to remove the common cause variation from the data and leave only those wanted changes which will achieve a drift in the characteristics of the final product (from those of grade K to those of grade L). Once the data undergoes this pre-processing, the methods can be applied and to compute the new conditions for either plant.

The method proposed by Jaeckle and MacGregor (2000b) uses an extended PCR (EPCR) model of  $Y$ , and the inversion of the this model to compute the new operating

conditions. The method can be summarized in seven steps (to transfer from plant A to plant B):

- i) Mean center and scale  $\mathbf{Xb}$  with respect to its own mean and standard deviation, and mean center  $\begin{bmatrix} \mathbf{Ya} \\ \mathbf{Yb} \end{bmatrix}$  using the mean and standard deviation from  $\mathbf{Yb}$ . Since  $\mathbf{Xa}$  is not used, the scaling of this matrix is not an issue.
- ii) Analyze of the quality data  $[\mathbf{Ya} \ \mathbf{Yb}]$  to determine if  $\mathbf{Ya}$  and  $\mathbf{Yb}$  share the same correlation structure. This is done by fitting a PCA model with  $\mathbf{Yb}$  and projecting  $\mathbf{Ya}$  thru this model, the SPE for those observations of  $\mathbf{Ya}$  to be transferred should be low and within confidence limits. Eliminate any grade with very large SPE values, since there is no evidence that they can successfully be made in plant B.
- iii) Perform an SVD on  $\mathbf{Xb}$  and the extended  $\begin{bmatrix} \mathbf{Ya} \\ \mathbf{Yb} \end{bmatrix}$  such that both matrices can be modeled as:

$$\hat{\mathbf{X}}_B = \mathbf{U}_B \cdot \boldsymbol{\Sigma}_B \cdot \mathbf{V}_B^T \quad (5.1)$$

$$\begin{bmatrix} \hat{\mathbf{Y}}_a \\ \hat{\mathbf{Y}}_b \end{bmatrix} = \begin{bmatrix} \mathbf{U}_{ya} \\ \mathbf{U}_{yb} \end{bmatrix} \cdot \boldsymbol{\Sigma}_Y \cdot \mathbf{V}_Y^T \quad (5.2)$$

- iv) Estimate a regression coefficient matrix between the latent variables of  $\mathbf{Xb}$  and  $\mathbf{Yb}$

$$\hat{\mathbf{R}}_B = \mathbf{U}_B^T \cdot \mathbf{U}_{yb} \quad (5.3)$$

- v) Invert the EPCR model to estimate  $\mathbf{u}_{\text{Bnew}}$  using the scores  $\mathbf{u}_{\text{ybdes}}$  corresponding to the grade to be transferred in  $\mathbf{Y}_a$  ( $\mathbf{u}_{\text{ybdes}}^T$  is a row from  $\mathbf{U}_{\text{ya}}$ ).

$$\hat{\mathbf{u}}_{\text{Bnew}}^T = \mathbf{u}_{\text{ybdes}}^T \cdot \left( \hat{\mathbf{R}}_B^T \cdot \hat{\mathbf{R}}_B \right)^{-1} \cdot \hat{\mathbf{R}}_B^T \quad (5.4)$$

- vi) Find the null space (if any) as,

$$\hat{\mathbf{u}}_{\text{Bnull}}^T = \lambda^T \cdot \mathbf{G}_2^T \quad (5.5)$$

where  $\lambda$  is an arbitrary vector and  $\mathbf{G}_2$  is defined by SVD on  $\hat{\mathbf{R}}_B$

$$\hat{\mathbf{R}}_B = \begin{matrix} & [\mathbf{G}_1 : \mathbf{G}_2] & \cdot & \Sigma_{\hat{\mathbf{R}}_B} & \cdot & \mathbf{V}_{\hat{\mathbf{R}}_B}^T \\ \text{(B} \times \text{C)} & \text{(B} \times \text{C)} \text{(B} \times \text{(B-C))} & & \text{(B} \times \text{C)} & & \text{(C} \times \text{C)} \end{matrix} \quad (5.6)$$

where B and C are the number of significant components of  $\mathbf{Xb}$  and  $\mathbf{Yb}$  respectively. If  $C < B$  then the null space exists as a set of multiple solutions in  $\mathbf{X}$  which will give the same prediction of  $\mathbf{Y}$  (see section 4.3.3 of this thesis).

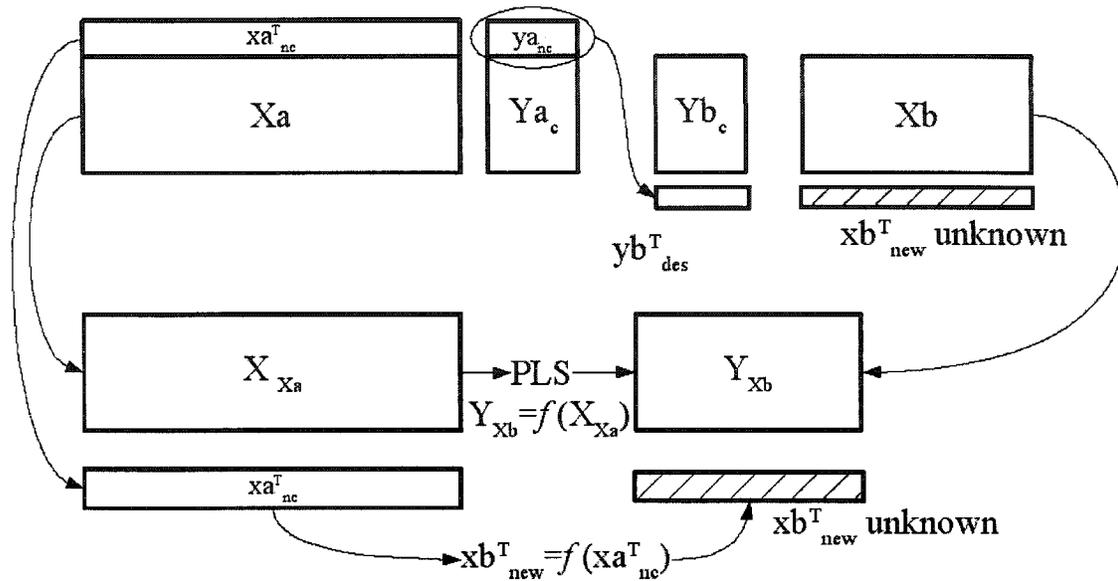
- vii) Estimate  $\hat{\mathbf{x}}_{\text{Bnew}}$  using the SVD loadings ( $\mathbf{V}_B$ ) and the new score augmented by the null space (if any).

$$\hat{\mathbf{x}}_{\text{Bnew}} = (\hat{\mathbf{u}}_{\text{Bnew}}^T + \hat{\mathbf{u}}_{\text{Bnull}}^T) \cdot \Sigma_B \cdot \mathbf{V}_B^T \quad (5.7)$$

This methodology is in essence the same as the one used in the product design exercise, except for the use of the augmented  $\begin{bmatrix} \mathbf{Y}_a \\ \mathbf{Y}_b \end{bmatrix}$  matrix. By using this augmented matrix, Jaeckle and MacGregor (2000b) recognize the importance of the *common plane* defined by the joint quality matrix, and the need to use both Y matrices to better estimate this common plane from the data which (in these scenarios) will be poor in observations since the data set includes one observation per grade. This *common plane* defined by the joint quality matrix is a key concept in the development of the JYPLS model as it will be shown further in section 5.2.

### 5.1.2. Motivation for a technique which uses all data structures

In the method suggested by Jaeckle and MacGregor (2000b), the data from the source plant ( $\mathbf{X}_a$  in Fig. 5.1) is not used at all. Before any attempt to even consider this matrix, a justification is needed to do so. Consider the scenario where plant A and plant B already produce a good number of common grades (Fig. 5.2,  $\mathbf{Y}_{a_c}$  and  $\mathbf{Y}_{b_c}$  are the same grades, and they are produced in both sites) and the one non-common grade from plant A is to be transferred to plant B (Fig. 5.2). In such case, a PLS model could be fitted (Fig. 5.2) using  $\mathbf{X}_a$  as  $\mathbf{X}$ , and  $\mathbf{X}_b$  as  $\mathbf{Y}$  because each of the rows in both sites correspond to the same grade ( $\mathbf{Y}_{a_c} = \mathbf{Y}_{b_c}$ ). Once fitted, the model ( $\mathbf{X}_b = f(\mathbf{X}_a)$ ) can be used for prediction, and the conditions corresponding to the grade to be transferred can be used to obtain the sought operating conditions in plant B ( $\mathbf{x}_{b_{new}} = f(\mathbf{x}_{a_{nc}})$ ). This approach is perfectly valid and makes no use whatsoever of the Y matrices. All the information required to do the transfer is in the  $\mathbf{X}_b$  and  $\mathbf{X}_a$  matrices.



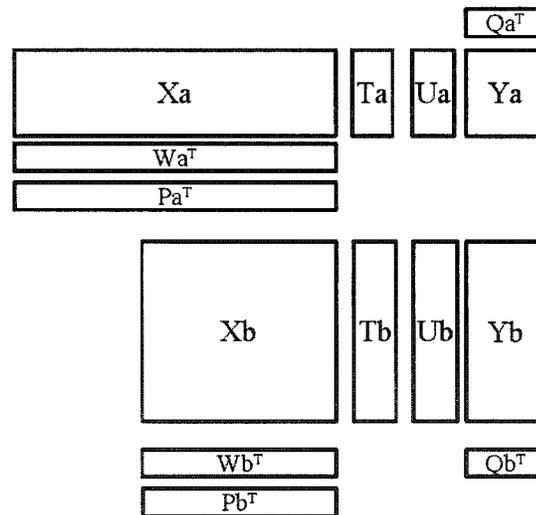
**Figure 5.2 Schematic use of PLS to directly estimate the conditions in plant B from those in plant A if enough common grades exist, with no use of the Y matrices**

In the above example, there is no doubt that  $Y_{a_c}$  and  $Y_{b_c}$  will have the same correlation structure since these are equal. If a PCA model is fitted with  $Y_{a_c}$ , the scores will fall in the same place as those of  $Y_{b_c}$  after being projected into the latent space defined by the loadings of this PCA model. If the grades produced in site A and site B are not common but share the same correlation structure, then the scores for each plant will span a region of the same latent plane. This suggests that, there should be a way to transfer information from  $X_a$  to  $X_b$  thru the latent space defined by the  $Y$ 's; and this new approach should incorporate  $X_a$  into the solution.

A novel latent variable regression method is now presented to achieve this goal: the Joint-Y PLS model. First its conceptual formulation is presented and an algorithm to estimate the parameters (loadings) for the model is presented and validated; then, the transfer problem is re-examined and solved with a JYPLS model.

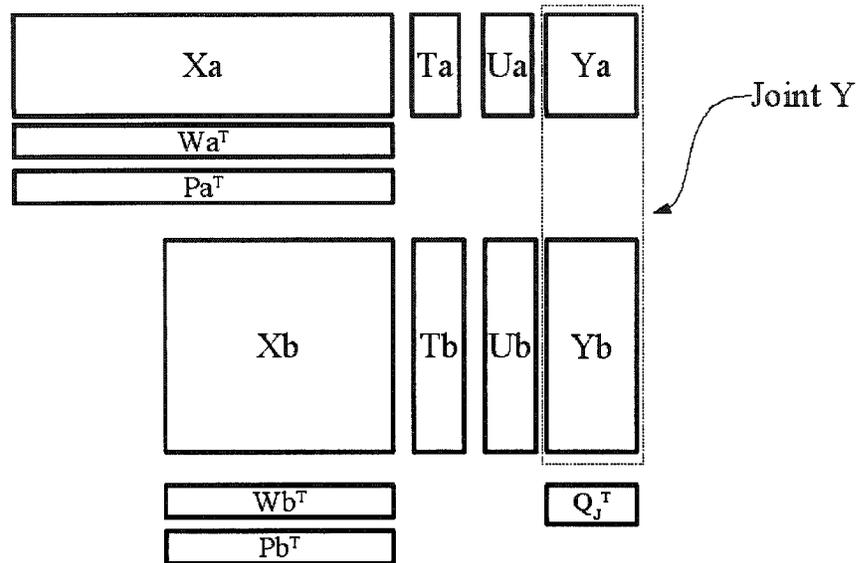
## 5.2. The Joint-Y PLS (JYPLS) Model

From a multivariate regression perspective, the most natural model to build with the four data matrices available ( $\mathbf{X}_A$ ,  $\mathbf{Y}_A$ ,  $\mathbf{X}_B$  and  $\mathbf{Y}_B$ ) are two separate PLS models that will map correlations and higher directions of variance between each set of operation conditions and grade characteristics (Fig. 5.3). And since the basic assumption is that matrices  $\mathbf{Y}_a$  and  $\mathbf{Y}_b$  should define the same plane, then the loadings  $\mathbf{Q}_a$  and  $\mathbf{Q}_b$  for the two PLS models built should be just a rotation of each other and can therefore be defined by one single loading matrix  $\mathbf{Q}_j$ , where the sub index  $j$  stands for “joint”. This  $\mathbf{Q}_j$  defines the plane described by the *Joint Y* composed of  $\mathbf{Y}_a$  and  $\mathbf{Y}_b$  (Fig. 5.4). This latent variable regression technique where the plane of the Joint-Y matrix is being modeled is referred to as the Joint-Y PLS model (JYPLS).



**Figure 5.3 Loadings and Scores for two separate PLS models**

Dimensionally, the only restriction in the JYPLS is that  $\mathbf{Y}_a$  and  $\mathbf{Y}_b$  must have the same variables defining the columns.



**Figure 5.4 Loadings and Scores for the Joint-Y PLS model**

The JYPLS model should not impose restrictions on the number of columns of  $\mathbf{X}_a$  and  $\mathbf{X}_b$ , nor on the number of observations per site. The number of sites is not a restriction as well, however for simplicity; the rest of the analysis in this chapter is done assuming two sites.

The loadings for the  $\mathbf{X}_a$  and  $\mathbf{X}_b$  matrices ( $\mathbf{P}_a$ ,  $\mathbf{W}_a$ ,  $\mathbf{P}_b$ ,  $\mathbf{W}_b$ ,) have the same interpretation as in the PLS model, but with respect to the *combined plane* mapped by the Joint Y matrix and defined by the loadings matrix  $\mathbf{Q}_J$ ; the  $\mathbf{W}_a$  and  $\mathbf{W}_b$  loadings define the direction of variation in the X space most correlated with the variation in the Y space defined by  $\mathbf{Q}_J$ . The reader is referred to the work by Burnham *et al.* (1999) for deeper insight in multivariate regression models. The JYPLS model is defined by equations 5.8 to 5.13.

$$\mathbf{Y}_J = \begin{bmatrix} \mathbf{Y}_a \\ \mathbf{Y}_b \end{bmatrix} = \begin{bmatrix} \mathbf{T}_a \\ \mathbf{T}_b \end{bmatrix} \mathbf{Q}_J^T + \mathbf{E}_{YJ} \quad (5.8)$$

$$\mathbf{X}_a = \mathbf{T}_a \mathbf{P}_a^T + \mathbf{E}_{X_a} \quad (5.9)$$

$$\mathbf{Xb} = \mathbf{Tb} \mathbf{Pb}^T + \mathbf{E}_{\mathbf{Xb}} \quad (5.10)$$

$$\mathbf{Ta} = \mathbf{Xa} \mathbf{Wa}^* \quad (5.11)$$

$$\mathbf{Tb} = \mathbf{Xb} \mathbf{Wb}^* \quad (5.12)$$

$$\mathbf{U}_J = \begin{bmatrix} \mathbf{Ua} \\ \mathbf{Ub} \end{bmatrix} = \mathbf{Y}_J \mathbf{Q}_J \quad (5.13)$$

Where  $\mathbf{Wa}^*$  and  $\mathbf{Wb}^*$  are transformations of the loadings to compute the scores directly from the original  $\mathbf{Xa}$  and  $\mathbf{Xb}$  (Eq 5.14 and 5.15), and  $\mathbf{E}_{\mathbf{YJ}}$ ,  $\mathbf{E}_{\mathbf{Xa}}$  and  $\mathbf{E}_{\mathbf{Xb}}$  represent the prediction errors. And  $\mathbf{Y}_J$  and  $\mathbf{U}_J$  are the joint matrices.

$$\mathbf{Wa}^* = \mathbf{Wa}(\mathbf{Pa}^T \mathbf{Wa})^{-1} \quad (5.14)$$

$$\mathbf{Wb}^* = \mathbf{Wb}(\mathbf{Pb}^T \mathbf{Wb})^{-1} \quad (5.15)$$

The JYPLS model makes sense from a practical point of view. If two plants are producing the same product family (e.g. film grade polyethylene), then their processes will involve similar physical and/or chemical phenomena, although these may happen in different types of units in each plant (separation columns in plant A and flash batteries in plant B) or in units with different configuration. Therefore, the two plants will have a common score space for their products ( $Y_a, Y_b$ ). Because the process variables measured in each site are also related with the common plane ( $\mathbf{T}_J$ ) defined by the grade family, then the variables between plants should be related as well. This brings the concept of *observability*: it is required for the differences between grades to be observable from the measured process variables in each site, even if the variables differ in nature. For example, plant A might be measuring the viscosity in a vessel while plant B measures the torque resistance in the agitator, these two variables are related to the resistance to motion of the mixture, and surely both will explain the same chemical characteristic in the product related to this intensive property of the mixture (e.g. Mw), and will appear in the same component of the model.

### 5.2.1. Parameter Estimation Approaches

Having defined the JYPLS model in (5.8) to (5.13), its parameters ( $\mathbf{Q}_J$ ,  $\mathbf{P}_a$ ,  $\mathbf{P}_b$ ,  $\mathbf{W}_a$  and  $\mathbf{W}_b$ ) must be estimated from the data on the two plants ( $\mathbf{X}_a$ ,  $\mathbf{X}_b$ ,  $\mathbf{Y}_a$  and  $\mathbf{Y}_b$ ). The natural path to follow in order to find a solution to the parameter estimation problem, is to position the JYPLS method within the framework established by Burnham *et al.* (1996), this requires that one define the objective function to be maximized in order to obtain the loadings matrices.

This objective function can be written (Eq. 5.16) as a variation of the objective function for the PLS model (Burnham, Viveros, and MacGregor, 1996). The objective is:

$$\begin{aligned} \max_{\mathbf{w}_a, \mathbf{w}_b} & \left\{ \begin{bmatrix} \mathbf{t}_a \\ \mathbf{t}_b \end{bmatrix}^T \begin{bmatrix} \mathbf{Y}_A \\ \mathbf{Y}_B \end{bmatrix} \begin{bmatrix} \mathbf{Y}_A \\ \mathbf{Y}_B \end{bmatrix}^T \begin{bmatrix} \mathbf{t}_a \\ \mathbf{t}_b \end{bmatrix} \right\} \\ \text{s.t.} & \\ \mathbf{t}_a &= \mathbf{X}_A \mathbf{w}_a \\ \mathbf{t}_b &= \mathbf{X}_B \mathbf{w}_b \\ \|\mathbf{w}_a\| &= \alpha \\ \|\mathbf{w}_b\| &= \beta \end{aligned} \tag{5.17}$$

In equation 5.17, the weight vectors ( $\mathbf{w}_a$  and  $\mathbf{w}_b$ ) are normalized to some values ( $\alpha$  and  $\beta$ ). However, it is not clear at this point, to which value these vectors should be normalized. The normalization of these vectors becomes clear in the analytical solution to the optimization problem in equation 5.17.

To analytically solve the problem in Eq 5.17 in a similar way as it is done for PLS, a common weight is defined (Eq 5.18). The norm constraint imposed to the  $\mathbf{w}$  loadings can be included into the objective function by the use of a lagrangian multiplier, only if this norm is equal to one (see Appendix 4). This gives a solution in the form of a eigenvector-eigenvalue problem (Eqs. 5.19) where  $\mathbf{w}_a$  and  $\mathbf{w}_b$  are segments of the  $\mathbf{w}$

vector, which is the eigenvector corresponding to the largest eigenvalue of the left hand matrix in Eq. 5.19 which involves all the data structures available.

$$\mathbf{w} = \begin{bmatrix} \mathbf{w}_a \\ \mathbf{w}_b \end{bmatrix} \quad (5.18)$$

$$\begin{bmatrix} \mathbf{X}_A^T \mathbf{Y}_A \mathbf{Y}_A^T \mathbf{X}_A & \mathbf{X}_A^T \mathbf{Y}_A \mathbf{Y}_B^T \mathbf{X}_B \\ \mathbf{X}_B^T \mathbf{Y}_B \mathbf{Y}_A^T \mathbf{X}_A & \mathbf{X}_B^T \mathbf{Y}_B \mathbf{Y}_B^T \mathbf{X}_B \end{bmatrix} \mathbf{w} = \lambda \mathbf{w} \quad (5.19)$$

The solution obtained (Eq. 5.19) reduces to the ordinary PLS solution (5.20) in the absence of one of the sites, or in the case where both sites have the same number and nature of columns and can be joined into one  $\mathbf{X}$  and one  $\mathbf{Y}$ .

$$[\mathbf{X}^T \mathbf{Y} \mathbf{Y}^T \mathbf{X}] \mathbf{w} = \lambda \mathbf{w} \quad (5.20)$$

Once the  $\mathbf{w}_a$  and  $\mathbf{w}_b$  loadings are computed, it is easy to compute the rest of the vectors for the JYPLS model.

$$\mathbf{t}_a = \mathbf{X}_a \mathbf{w}_a (\mathbf{w}_a^T \mathbf{w}_a)^{-1} \quad (5.21)$$

$$\mathbf{t}_b = \mathbf{X}_b \mathbf{w}_b (\mathbf{w}_b^T \mathbf{w}_b)^{-1} \quad (5.22)$$

$$\mathbf{q}_J = \begin{bmatrix} \mathbf{Y}_a \\ \mathbf{Y}_b \end{bmatrix}^T \begin{bmatrix} \mathbf{t}_a \\ \mathbf{t}_b \end{bmatrix} \left( \begin{bmatrix} \mathbf{t}_a \\ \mathbf{t}_b \end{bmatrix}^T \begin{bmatrix} \mathbf{t}_a \\ \mathbf{t}_b \end{bmatrix} \right)^{-1} \quad (5.23)$$

$$\begin{bmatrix} \mathbf{u}_a \\ \mathbf{u}_b \end{bmatrix} = \begin{bmatrix} \mathbf{Y}_a \\ \mathbf{Y}_b \end{bmatrix} \mathbf{q}_J (\mathbf{q}_J^T \mathbf{q}_J)^{-1} \quad (5.24)$$

In order to deflate, the  $\mathbf{p}$  vectors can be also computed as:

$$\mathbf{p}_a = \mathbf{X}_a^T \mathbf{t}_a (\mathbf{t}_a^T \mathbf{t}_a)^{-1} \quad (5.25)$$

$$\mathbf{pb} = \mathbf{Xb}^T \mathbf{tb}(\mathbf{tb}^T \mathbf{tb})^{-1} \quad (5.26)$$

The computed vectors  $\mathbf{wa}$ ,  $\mathbf{wb}$ ,  $\mathbf{ta}$ ,  $\mathbf{tb}$ ,  $\mathbf{q}_J$ ,  $\mathbf{ua}$ ,  $\mathbf{ub}$ ,  $\mathbf{pa}$  and  $\mathbf{pb}$  can now be stored and the data matrices deflated (Eqs. 5.27 to 5.30) before computing the loadings vectors for the next component.

$$\mathbf{Xa} = \mathbf{Xa} - \mathbf{ta} \mathbf{pa}^T \quad (5.27)$$

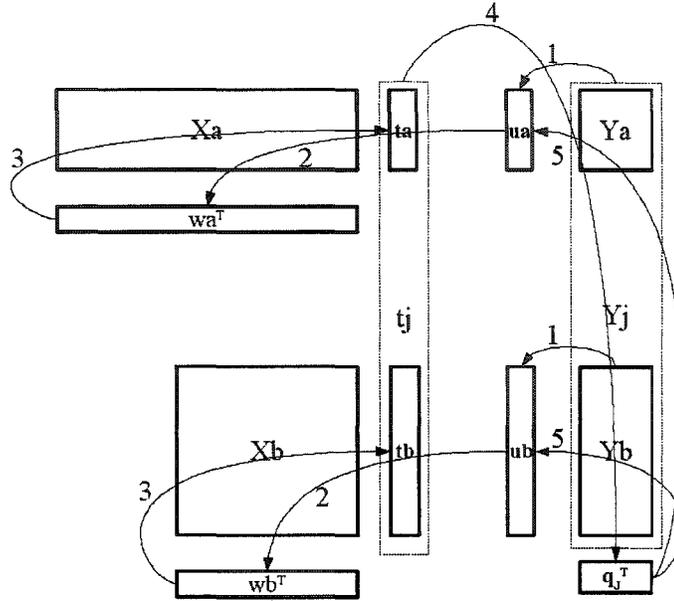
$$\mathbf{Xb} = \mathbf{Xb} - \mathbf{tb} \mathbf{pb}^T \quad (5.28)$$

$$\mathbf{Ya} = \mathbf{Ya} - \mathbf{ta} \mathbf{q}_J^T \quad (5.29)$$

$$\mathbf{Yb} = \mathbf{Yb} - \mathbf{tb} \mathbf{q}_J^T \quad (5.30)$$

Solving equation 5.19 using SVD and then following the calculations in equations 5.21 to 5.30 is a way to build a JYPLS model one component at a time.

Just as with ordinary PLS, an iterative NIPALS algorithm can be formulated that will converge to the solution of the above eigenvalue-eigenvector problem. The major advantage of the NIPALS algorithm would be that it can easily handle missing data in the  $\mathbf{X}$  and  $\mathbf{Y}$  matrices.



**Figure 5.5** NIPALS modified to compute loadings the JYPLS model

The NIPALS algorithm for the JYPLS problem is (Fig. 5.5):

1. Initialize  $u_a$  and  $u_b$  with the first column of  $Y_a$  and  $Y_b$  respectively
2. Regress  $X_a$  and  $X_b$  onto  $u_a$  and  $u_b$  to compute  $w_a$  and  $w_b$  as:

$$w_a = X_a^T u_a (u_a^T u_a)^{-1} \quad (5.31)$$

$$w_b = X_b^T u_b (u_b^T u_b)^{-1} \quad (5.32)$$

Normalize  $w_a$  and  $w_b$  as:

$$\begin{bmatrix} w_a \\ w_b \end{bmatrix} = 1 \quad (5.33)$$

3. Regress  $X_a$  and  $X_b$  onto  $w_a$  and  $w_b$  to obtain  $t_a$  and  $t_b$  (Eqs. 5.21 and 5.22)
4. Regress the joint  $Y$  onto the joint  $t$  vector to obtain the joint  $q$  (Eq 5.23)
5. Regress  $Y_a$  and  $Y_b$  onto  $q_j$  to re-compute  $u_a$  and  $u_b$  (Eq. 5.24), convergence is checked with respect to the original values of  $u_a$  and  $u_b$  if convergence fails, use new values of  $u_a$  and  $u_b$  and go to step 2.
6. If converged, calculate  $p_a$  and  $p_b$  for deflation (Eqs 5.25 and 5.26)
7. Deflate  $X_a$ ,  $X_b$ ,  $Y_a$  and  $Y_b$  (Eqs. 5.27 to 5.30) and estimate next component

The algorithm above is identical to the one used to build two separate PLS models with the exception of step 4, in which the  $\mathbf{q}_J$  loading is computed with the joint score vector. This step will achieve the main goal of the algorithm which is to find the common plane defined by both  $\mathbf{Y}\mathbf{a}$  and  $\mathbf{Y}\mathbf{b}$ , since both matrices are being projected simultaneously on to the score space.

In the case of the NIPALS algorithm for PLS, it is possible to prove that, at convergence, the algorithm will provide the same solution as the eigenvector problem in Eq. 5.20; provided that the scaling of the weight vectors (Eq. 5.17) is done properly. Appendix 5 contains the mathematical proof that, the modified-NIPALS algorithm will provide the same eigenvalue-eigenvector solution as the one in equation 5.19 if the scalar values  $\alpha_1$  to  $\alpha_4$  (Eq. 5.34) are equal.

$$\begin{bmatrix} (\mathbf{w}\mathbf{a}^T\mathbf{w}\mathbf{a})^{-1}(\mathbf{t}_J^T\mathbf{t}_J)^{-1}(\mathbf{q}_J^T\mathbf{q}_J)^{-1}(\mathbf{u}\mathbf{a}^T\mathbf{u}\mathbf{a})^{-1} \\ (\mathbf{w}\mathbf{b}^T\mathbf{w}\mathbf{b})^{-1}(\mathbf{t}_J^T\mathbf{t}_J)^{-1}(\mathbf{q}_J^T\mathbf{q}_J)^{-1}(\mathbf{u}\mathbf{a}^T\mathbf{u}\mathbf{a})^{-1} \\ (\mathbf{w}\mathbf{a}^T\mathbf{w}\mathbf{a})^{-1}(\mathbf{t}_J^T\mathbf{t}_J)^{-1}(\mathbf{q}_J^T\mathbf{q}_J)^{-1}(\mathbf{u}\mathbf{b}^T\mathbf{u}\mathbf{b})^{-1} \\ (\mathbf{w}\mathbf{b}^T\mathbf{w}\mathbf{b})^{-1}(\mathbf{t}_J^T\mathbf{t}_J)^{-1}(\mathbf{q}_J^T\mathbf{q}_J)^{-1}(\mathbf{u}\mathbf{b}^T\mathbf{u}\mathbf{b})^{-1} \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} \quad (5.34)$$

For these scalars to be equal the norms of  $\mathbf{w}\mathbf{a}$  and  $\mathbf{w}\mathbf{b}$  should be equal as well as the norm of  $\mathbf{u}\mathbf{a}$  and  $\mathbf{u}\mathbf{b}$ . Since  $\mathbf{w}\mathbf{a}$  and  $\mathbf{w}\mathbf{b}$  are being normalized jointly (Eq. 5.33) then their individual norms will depend partially on the number of variables per site (and on the degree of correlation of each site with its respective  $y$ ). The norms of  $\mathbf{u}\mathbf{a}$  and  $\mathbf{u}\mathbf{b}$  are an indication of the total leverage per site with respect to the model; this is highly dependant on the number of observations per site. This condition (all  $\alpha$ 's to be equal) as well as intuition (since the JYPLS model is allowing  $\mathbf{X}\mathbf{a}$  and  $\mathbf{X}\mathbf{b}$  to have different number of columns, and each site to have different number of observations) suggests that a special scaling is required in the pre-processing of the data. A similar scaling as the one required for a multi-block model (Westerhuis, Kourti, and MacGregor, 1998) is proposed.

After mean centering each matrix individually, all elements in the matrices should be scaled as:

$$\mathbf{Xa}_{i,j} = \frac{\mathbf{Xa}_{i,j}}{\sqrt{i_a \times m_a}} \quad (5.35)$$

$$\mathbf{Xb}_{i,j} = \frac{\mathbf{Xb}_{i,j}}{\sqrt{i_b \times m_b}} \quad (5.36)$$

$$\mathbf{Ya}_{i,j} = \frac{\mathbf{Ya}_{i,j}}{\sqrt{i_a}} \quad (5.37)$$

$$\mathbf{Yb}_{i,j} = \frac{\mathbf{Yb}_{i,j}}{\sqrt{i_b}} \quad (5.38)$$

where  $i_a$  and  $i_b$  are the number of observations in plant A and B respectively, and  $m_a$  and  $m_b$  are the number of columns in  $\mathbf{Xa}$  and  $\mathbf{Xb}$  respectively.  $\mathbf{Ya}$  and  $\mathbf{Yb}$  must have the same number of columns; these matrices do not require scaling with respect to their number of columns.

A test set is prepared with no noise, in which  $\mathbf{Xa}$  and  $\mathbf{Xb}$  are equally explained by the 3 component JYPLS model ( $R^2Xa_{cum} = 1$  and  $R^2Xb_{cum} = 1$ ) as well as  $\mathbf{Ya}$  and  $\mathbf{Yb}$  which are also equally captured by the model ( $R^2Ya_{cum} = 0.95$  and  $R^2Yb_{cum} = 0.96$ ), the test set contains 100 observations for site A and 50 observations for site B; 8 variables in  $\mathbf{Xa}$  and 13 in  $\mathbf{Xb}$ ;  $\mathbf{Ya}$  and  $\mathbf{Yb}$  contain 6 columns. For this test, and applying the suggested scaling,  $\alpha_1$  and  $\alpha_2$  were equal to 0.33 while  $\alpha_3$  and  $\alpha_4$  were equal to 0.34 (both values for the first component). In spite of the strong differences in the number of observations per site and variables in each of the X spaces the loadings given by the modified NIPALS are a very good approximation of those given by the SVD (by computing the eigenvectors and eigenvalues of the far left hand matrix in Eq 5.19). This verifies that the suggested scaling in Equations 5.35 to 5.38. does indeed make the NIPALS solution equivalent to the eigenvector problem in Equation 5.19.

The loadings for the first component computed by SVD are compared with those obtained with the modified-NIPALS algorithm in Table 5.1. The actual values of  $\mathbf{Xa}$ ,  $\mathbf{Xb}$ ,  $\mathbf{Ya}$  and  $\mathbf{Yb}$  for the test problem are available in the electronic supplement to this thesis.

For 1st Component	Mod-NIPALS	SVD
wa <sub>1</sub>	0.36	0.37
wa <sub>2</sub>	0.2	0.2
wa <sub>3</sub>	0.36	0.37
wa <sub>4</sub>	0.39	0.4
wa <sub>5</sub>	0.2	0.21
wa <sub>6</sub>	0.06	0.06
wa <sub>7</sub>	0.06	0.06
wa <sub>8</sub>	0.06	0.06
wb <sub>1</sub>	0.29	0.29
wb <sub>2</sub>	0.13	0.12
wb <sub>3</sub>	0.16	0.16
wb <sub>4</sub>	0.31	0.3
wb <sub>5</sub>	0.29	0.29
wb <sub>6</sub>	0.01	0.01
wb <sub>7</sub>	0.13	0.12
wb <sub>8</sub>	0.23	0.22
wb <sub>9</sub>	0.33	0.32
wb <sub>10</sub>	0.13	0.13
wb <sub>11</sub>	0.01	0.01
wb <sub>12</sub>	0.01	0.01
wb <sub>13</sub>	0.01	0.01

**Table 5.1 Values for wa and wb obtained with modified-NIPALS and with SVD for a JYPLS model fitted to the test set in Appendix 6**

### 5.3. JYPLS Model Diagnostics

The JYPLS model has been conceptually presented, its mathematical foundations formulated as well as algorithms to compute its parameters. The proper diagnostics should now be established in order to assess the quality of a model. Such diagnostics are the same as those from two separated PLS models with some minor modifications in how the residuals for the Y's are computed.

The diagnostics to be considered are listed in Table 5.2. The R2 diagnostics measure the fraction (or percentage) of the total sum of squares that is being captured (explained) by the model; Q2 diagnostics are obtained by cross validation (Wold, 1978) and give a measure of the prediction capabilities of the model for each of the Y spaces; the square prediction error (SPE) is a measure of the orthogonal distance from the observation to its projection onto the model plane, and the Hotelling's  $T^2$  gives the distance of each observation on the projection plane from the origin of the model plane.

Matrix	R2 and Q2	SPE	Hotelling's $T^2$
<b>Xa</b>	R2X <sub>a</sub> , R2Xacum	SPEX <sub>a<sub>i</sub></sub>	HotT <sup>2</sup> X <sub>a<sub>i</sub></sub>
<b>Xb</b>	R2X <sub>b</sub> , R2Xbcum	SPEX <sub>b<sub>i</sub></sub>	HotT <sup>2</sup> X <sub>b<sub>i</sub></sub>
<b>Ya</b>	R2Y <sub>a</sub> , R2Yacum Q2Y <sub>a</sub> , Q2Yacum	SPEY <sub>a<sub>i</sub></sub>	
<b>Yb</b>	R2Y <sub>b</sub> , R2Ybcum Q2Y <sub>b</sub> , Q2Ybcum	SPEY <sub>b<sub>i</sub></sub>	

**Table 5.2 Diagnostics to be considered in the JYPLS model**

Sub index  $i$  indicates that the diagnostic is computed per observation and sub index  $a$  indicates the diagnostic is computed per component. The only significant difference in the computation of these values from separate PLS models is that the residuals for both of the Y matrices are computed using the same  $\mathbf{Q}_J$  loadings matrix. It is assumed that data underwent proper mean centering and scaling as in Eqs.5.35 to 5.38 before building the JYPLS model. The diagnostics are computed with these mean centered and scaled matrices, the equations to compute these diagnostics are given below.

$$R2Xa_{a|a:1 \rightarrow A} = 1 - \frac{RSSXa_a}{TSSXa} \quad (5.39)$$

$$R2Xb_{a|a:1 \rightarrow A} = 1 - \frac{RSSXb_a}{TSSXb} \quad (5.40)$$

$$R2Ya_{a|a:1 \rightarrow A} = 1 - \frac{RSSYa_a}{TSSYa} \quad (5.41)$$

$$R2Yb_{a|a:1 \rightarrow A} = 1 - \frac{RSSYb_a}{TSSYb} \quad (5.42)$$

$$Q2Ya_{a|a:1 \rightarrow A} = 1 - \frac{PRESSYa_a}{TSSYa} \quad (5.43)$$

$$Q2Yb_{a|a:1 \rightarrow A} = 1 - \frac{PRESSYb_a}{TSSYb} \quad (5.44)$$

$$R2Xa_{cum} = \sum_{a=1}^A R2Xa_a \quad (5.45)$$

$$R2Xb_{cum} = \sum_{a=1}^A R2Xb_a \quad (5.46)$$

$$R2Ya_{cum} = \sum_{a=1}^A R2Ya_a \quad (5.47)$$

$$R2Yb_{cum} = \sum_{a=1}^A R2Yb_a \quad (5.48)$$

$$Q2Ya_{cum} = \sum_{a=1}^A Q2Ya_a \quad (5.49)$$

$$Q2Yb_{cum} = \sum_{a=1}^A Q2Yb_a \quad (5.50)$$

$$SPEXa_i \Big|_{i:1 \rightarrow ia} = \sum_{a=1}^{ma} (\mathbf{Xa}_i - \mathbf{Ta}_i \mathbf{Pa}^T)^2 \quad (5.51)$$

$$SPEXb_i \Big|_{i:1 \rightarrow ib} = \sum_{a=1}^{mb} (\mathbf{Xb}_i - \mathbf{Tb}_i \mathbf{Pb}^T)^2 \quad (5.52)$$

$$SPEYa_i \Big|_{i:1 \rightarrow ia} = \sum_{a=1}^n (\mathbf{Ya}_i - \mathbf{Ta}_i \mathbf{Q}_J^T)^2 \quad (5.53)$$

$$SPEYb_i \Big|_{i:1 \rightarrow ib} = \sum_{a=1}^n (\mathbf{Yb}_i - \mathbf{Tb}_i \mathbf{Q}_J^T)^2 \quad (5.54)$$

$$HOTT^2 Xa_i \Big|_{i:1 \rightarrow ia} = \sum_{a=1}^A \frac{\mathbf{ta}_{i,a}^2}{\sigma_{tja}^2} \quad (5.55)$$

$$HOTT^2 Xb_i \Big|_{i:1 \rightarrow ib} = \sum_{a=1}^A \frac{\mathbf{tb}_{i,a}^2}{\sigma_{tja}^2} \quad (5.56)$$

$$RSSXa_a = \sum_{i=1}^{ia} \sum_{m=1}^{ma} (\mathbf{Xa} - \mathbf{Ta}_{[1:a]} \mathbf{Pa}_{[1:a]}^T)_{i,m}^2 \quad (5.57)$$

$$TSSXa = \sum_{i=1}^{ia} \sum_{m=1}^{ma} \mathbf{Xa}_{i,m}^2 \quad (5.58)$$

$$RSSXb_a = \sum_{i=1}^{ib} \sum_{m=1}^{mb} (\mathbf{Xb} - \mathbf{Tb}_{[1:a]} \mathbf{Pb}_{[1:a]}^T)_{i,m}^2 \quad (5.59)$$

$$TSSXb = \sum_{i=1}^{ib} \sum_{m=1}^{mb} \mathbf{Xb}_{i,m}^2 \quad (5.60)$$

$$RSSYa_a = \sum_{i=1}^{ia} \sum_{m=1}^n (\mathbf{Ya} - \mathbf{Ta}_{[1:a]} \mathbf{Q}_{J[1:a]}^T)_{i,m}^2 \quad (5.61)$$

$$TSSYa = \sum_{i=1}^{ia} \sum_{m=1}^n \mathbf{Ya}_{i,m}^2 \quad (5.62)$$

$$RSSYb_a = \sum_{i=1}^{ib} \sum_{m=1}^n (\mathbf{Yb} - \mathbf{Tb}_{[1:a]} \mathbf{Q}_{J[1:a]}^T)_{i,m}^2 \quad (5.63)$$

$$TSSYb = \sum_{i=1}^{ia} \sum_{m=1}^n \mathbf{Yb}_{i,m}^2 \quad (5.64)$$

$$\sigma_{tja}^2 = \frac{1}{i_a + i_b - 1} \sum_{i=1}^{ia+ib} [\mathbf{T}_{Ji,a}^2] \quad (5.65)$$

$$\mathbf{T}_J = \begin{bmatrix} \mathbf{Ta} \\ \mathbf{Tb} \end{bmatrix} \quad (5.66)$$

### 5.4. Modifications to the JYPLS Model

Once the JYPLS model has been defined, other modifications can be done to the model to achieve specific objectives. Three modifications are considered here: the extension to multiple sites, the multi-block JYPLS and the non-linear version of the JYPLS model.

#### 5.4.1. Multiple Sites

As mentioned before, the JYPLS model is not restricted to two sites. Any number of sites can be considered (Fig. 5.6). The eigenvector eigenvalue solution becomes more complex (Eq. 5.67 for  $\eta$  sites). In this case it is much easier to consider the modified-NIPALS for the computation of the loadings for multiple sites. The modifications are quite straightforward; they involve regressing each of the multiple set of loadings and scores. The normalization of the loadings should be done in group as well (Eq. 5.68).

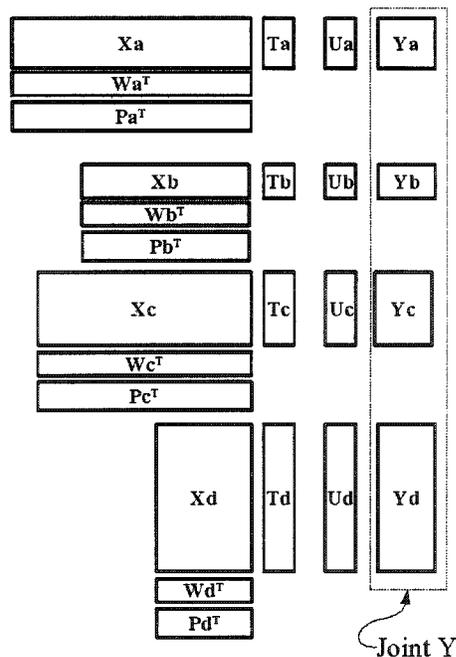


Figure 5.6 JYPLS built with multiple sites

$$\begin{bmatrix} \mathbf{Xa}^T \mathbf{Ya} \mathbf{Ya}^T \mathbf{Xa} & \mathbf{Xa}^T \mathbf{Ya} \mathbf{Yb}^T \mathbf{Xb} & \cdots & \mathbf{Xa}^T \mathbf{Ya} \mathbf{Y\eta}^T \mathbf{X\eta} \\ \mathbf{Xb}^T \mathbf{Yb} \mathbf{Ya}^T \mathbf{Xa} & \mathbf{Xb}^T \mathbf{Yb} \mathbf{Yb}^T \mathbf{Xb} & & \\ \vdots & & \ddots & \\ \mathbf{X\eta}^T \mathbf{Y\eta} \mathbf{Ya}^T \mathbf{Xa} & \cdots & & \mathbf{X\eta}^T \mathbf{Y\eta} \mathbf{Y\eta}^T \mathbf{X\eta} \end{bmatrix} \mathbf{w} = \lambda \mathbf{w} \quad (5.67)$$

$$\begin{bmatrix} \|\mathbf{wa}\| \\ \|\mathbf{wb}\| \\ \vdots \\ \|\mathbf{w\eta}\| \end{bmatrix} = 1 \quad (5.68)$$

#### 5.4.2. Multi-Block JYPLS Models (MBJYPLS)

Multi-block PLS (MBPLS) models (Wangen and Kowalski, 1989; Westerhuis, Kourti, and MacGregor, 1998; Westerhuis and Smilde, 2001) are useful to analyze data which can naturally be separated in blocks. These models have been successfully applied in the analysis of industrial data (Kourti, Nomikos, and MacGregor, 1995; MacGregor *et al.*, 1994). The JYPLS model is easily extended to handle multiple blocks per site (Fig. 5.7). The only alteration to the original JYPLS model algorithm is the scaling of each of the blocks per site. For the example of two blocks per site, the scaling given in equations 5.35 and 5.36 now becomes as follows:

$$\mathbf{Xa}_{i,j} = \frac{\mathbf{Xa}_{i,j}}{\sqrt{i_a \times (m_a + l_a) \times m_a}} \quad (5.69) \quad \mathbf{Xb}_{i,j} = \frac{\mathbf{Xb}_{i,j}}{\sqrt{i_b \times (m_b + l_b) \times m_b}} \quad (5.70)$$

$$\mathbf{Za}_{i,j} = \frac{\mathbf{Za}_{i,j}}{\sqrt{i_a \times (m_a + l_a) \times l_a}} \quad (5.71) \quad \mathbf{Zb}_{i,j} = \frac{\mathbf{Zb}_{i,j}}{\sqrt{i_b \times (m_b + l_b) \times l_b}} \quad (5.72)$$

where  $l_a$  and  $l_b$  are the number of columns in  $\mathbf{Za}$  and  $\mathbf{Zb}$  respectively (Fig. 5.7). The scaling of the  $\mathbf{Y}$  matrices does not change. After this scaling is applied, a JYPLS model can be built and the block scores and loadings can be computed as in the PLS to MBPLS case as given by (Westerhuis, Kourti, and MacGregor, 1998).

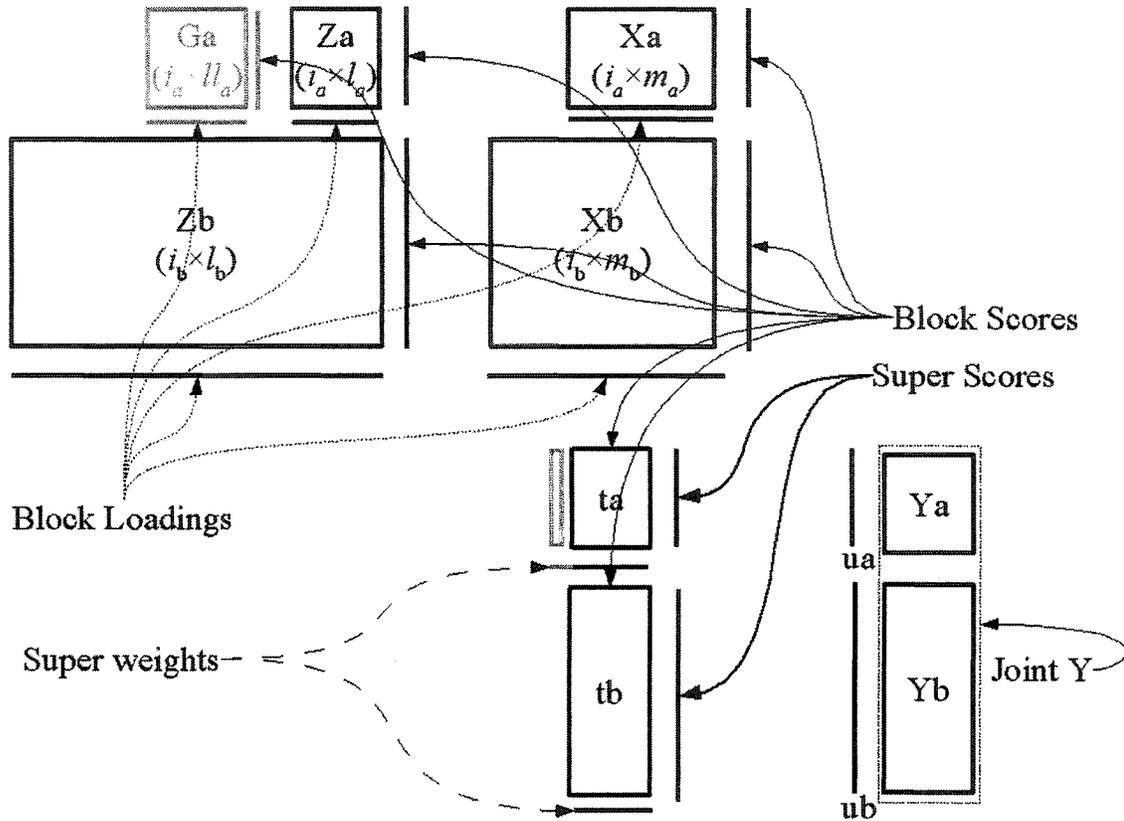


Figure 5.7 Multi-block JYPLS model data structures

The MBJYPLS model does not restrict all sites to have the same number of blocks (as it does not restrict the number of columns). The block score matrix will contain different columns per site, and the super-weights will have different number of elements (extra block  $G_a$  with  $i_a$  rows and  $l_a$  columns illustrated in gray in Fig. 5.7).

A MBJYPLS model is useful, for example, when the process involved operates in batch mode, and the data contains information of the initial conditions as well as the variable trajectories.

### 5.4.3. Non-Linear JYPLS

The justification to perform multivariate linear analysis in process monitoring is the that, even if the process behavior is non-linear, the deviations from a set point or a nominal trajectory will be locally linear (although the covariance structure will change from time point to time point throughout a batch). In the case of product transfer, the region being spanned by the data is wider; and therefore nonlinearities may become an issue. The nonlinearities among the process variables can be dealt with transformations (as it is usually done) and the nonlinearities between process variables (X's) and final product characteristics (Y's) can be dealt with transformations as well, or/and with a nonlinear version of the JYPLS.

The nonlinear relationships among the scores ( $\mathbf{t}$ 's and  $\mathbf{u}$ 's) can be established as in a conventional nonlinear PLS model. For example, a quadratic relationship can be used as in the quadratic nonlinear PLS (Hoskuldsson, 1992), in this particular case, the scores are related by a quadratic function (Eqs.5.73 and 5.74). The algorithm to compute the loadings and scores for the nonlinear quadratic JYPLS is a natural extension to the one proposed by Wold *et al.* (1989) and is included for reference in appendix 6.

$$\mathbf{u}\mathbf{a}_i^a = b\mathbf{a}_0^a + b\mathbf{a}_1^a \mathbf{t}\mathbf{a}_i^a + b\mathbf{a}_2^a (\mathbf{t}\mathbf{a}_i^a)^2 \quad (5.73)$$

$$\mathbf{u}\mathbf{b}_i^a = b\mathbf{b}_0^a + b\mathbf{b}_1^a \mathbf{t}\mathbf{b}_i^a + b\mathbf{b}_2^a (\mathbf{t}\mathbf{b}_i^a)^2 \quad (5.74)$$

### 5.5. Product Transfer using JYPLS

The justification to study the inclusion of all data structures in the product transfer problem was given (see Section 5.1.2) and a modeling technique has been developed for this purpose (see Section 5.2). This section illustrates how the JYPLS model can be used to solve the product transfer problem. Throughout this chapter, the scenario considered is

the one where a grade from plant A is to be transferred to plant B; the sought operating conditions are those of plant B; the grade to be transferred  $\mathbf{y}_{Bdes}$  is a row from  $\mathbf{Ya}$ .

### 5.5.1. Transfer Technique

1. Mean center  $\mathbf{Xa}$ ,  $\mathbf{Xb}$ ,  $\mathbf{Ya}$  and  $\mathbf{Yb}$  with respect to their own mean, and scale according to equations 5.35 to 5.38.
2. Ensure that  $\mathbf{Ya}$  and  $\mathbf{Yb}$  share the same correlation structure. This can be done by building a PCA model with  $\mathbf{Yb}$  and projecting  $\mathbf{Ya}$  on to the model, the SPE for  $\mathbf{Ya}$  should be within limits. Individual PCA models will give a good idea on the effective rank of each of the Y's, a PCA model on the joint  $\begin{bmatrix} \mathbf{Ya} \\ \mathbf{Yb} \end{bmatrix}$  matrix should have a rank equal to  $\max\{rank(Ya), rank(Yb)\}$  if  $\mathbf{Ya}$  and  $\mathbf{Yb}$  do share a common correlation structure. The PCA model on the joint  $\mathbf{Y}$  matrix may explain less percentage of the data since the signal to noise ratios may differ from site A to site B.
3. Fit a JYPLS model to the data using the modified-NIPALS described in section 5.2.1.
4. Assess the model using the diagnostics such as the  $SPEYa$ ,  $SPEYb$ ,  $SPEXa$  and  $SPEXb$  to find possible outliers. Particular attention should be given to the observation to be transferred.
5. Mean center and scale  $\mathbf{y}_{Bdes}$  using the same mean and scaling that was applied to  $\mathbf{Yb}$ .
6. Invert the JYPLS model to estimate  $\tau\mathbf{b}_{new}$  which is the score vector which best predicts  $\mathbf{y}_{Bdes}$ . The inversion of the JYPLS model can be done using a generalized pseudo-inverse of  $(\mathbf{Q}_J^T \mathbf{Q}_J)$  (Eq 5.75) or via optimization as illustrated in chapter 4 (Eqs 4.7 and 4.7b), with the same effects as those illustrated in chapter 4 for the product design exercise.

$$\hat{\boldsymbol{\tau}}_{\mathbf{b}_{\text{new}}} = (\mathbf{Q}_J^T \mathbf{Q}_J)^{-1} \mathbf{Q}_J^T \mathbf{y}_{\text{Bdes}} \quad (5.75)$$

7. Estimate the null-space ( $\Delta \hat{\boldsymbol{\tau}}_{\text{null}}$ ) if any (section 5.5.2)
8. Construct  $\hat{\mathbf{x}}_{\mathbf{B}_{\text{new}}}$  as

$$\hat{\mathbf{x}}_{\mathbf{B}_{\text{new}}} = \mathbf{P} \mathbf{b} (\hat{\boldsymbol{\tau}}_{\mathbf{b}_{\text{new}}} + \Delta \hat{\boldsymbol{\tau}}_{\text{null}}) \quad (5.76)$$

9. Remove scale and add the proper mean to  $\hat{\mathbf{x}}_{\mathbf{B}_{\text{new}}}$  in the inverse way as it was done with  $\mathbf{X} \mathbf{b}$ .

### 5.5.2. Null Space in JYPLS

Jaeckle and MacGregor (1998) give an extensive discussion about the null space, referring to it as the subspace of the score space in which  $\boldsymbol{\tau} \mathbf{b}_{\text{new}}$  can *move* and still produce the same  $\hat{\mathbf{y}}_{\text{Bdes}}$ .

One case when this null space will exist is when the effective rank of  $\mathbf{Y}$  is smaller than the effective rank of  $\mathbf{X}$ , e.g.  $\text{rank}(\mathbf{Y}) = A$ ,  $\text{rank}(\mathbf{X}) = B$  and  $A < B$ , in which case there is a  $B-A$  dimensional null space. Another case where the null space will exist is when one or more dimensions of  $\mathbf{Y}$  do not overlap with  $\mathbf{X}$ . In this case the rank of  $\mathbf{Y}$  may be equal to rank of  $\mathbf{X}$ , but the rank of  $[\mathbf{X} \ \mathbf{Y}]$  is greater than the rank of  $\mathbf{Y}$  (or  $\mathbf{X}$ ) e.g.  $\text{rank}(\mathbf{Y}) = \text{rank}(\mathbf{X}) = A$ ,  $\text{rank}([\mathbf{X} \ \mathbf{Y}]) = B$ , and  $B > A$ ; in this case there are  $B-A$  dimensions which do not overlap between  $\mathbf{X}$  and  $\mathbf{Y}$ , giving a  $B-A$  dimensional null space. For the JYPLS model, these circumstances may happen, therefore the same considerations as those pointed out in section 4.3.3 apply for the JYPLS case.

If there is a null space, the desired operating conditions should be computed as:

$$\hat{\mathbf{x}}_{\text{Bnew}} = \mathbf{P}\mathbf{b}(\hat{\boldsymbol{\tau}}\mathbf{b}_{\text{new}} + \Delta\hat{\boldsymbol{\tau}}\mathbf{b}_{\text{null}}) \quad (5.77)$$

where the null movement from the original solution ( $\Delta\hat{\boldsymbol{\tau}}\mathbf{b}_{\text{null}}$ ) lies in a subspace defined by SVD on  $\mathbf{Q}_J^T$  (in Eqs. 5.78 and 5.79 below)  $\lambda$  is an arbitrary vector which is relative to the magnitude of the movement in the null space, and B and C are the dimensions of the latent spaces in X and Y respectively.

$$\Delta\hat{\boldsymbol{\tau}}\mathbf{b}_{\text{null}} = \lambda \mathbf{G}_2^T \quad (5.78)$$

$$\mathbf{Q}_J^T = \begin{bmatrix} \mathbf{G}_1 & \mathbf{G}_2 \end{bmatrix} \cdot \begin{matrix} \Sigma_{\mathbf{Q}_J^T} \\ \mathbf{V}_{\mathbf{Q}_J^T}^T \end{matrix} \quad (5.79)$$

(B×C)    (B×C)(B×(B-C))    (B×C)    (C×C)

Another (more intuitive) way to find the null space is to map each univariate null space by building the system of linear equations described by  $\mathbf{Q}_J \hat{\boldsymbol{\tau}}\mathbf{b}_{\text{null}} = 0$ . In a pure multivariate null space, the univariate null spaces for those correlated columns of Y should be perfectly collinear (see section 4.3.3 for an extensive discussion on univariate null spaces and multivariate null and pseudo-null spaces).

## 5.6. Product Transfer using EPCR vs. JYPLS

The initial motivation to develop the JYPLS method was to include the information about the operating conditions at the source plant ( $\mathbf{Xa}$  in Fig. 5.1), because it was not used in the EPCR approach of Jaeckle and MacGregor (2000b). Intuitively, the solution obtained when such information is considered will be better if the extra data ( $\mathbf{Xa}$ ) enhance the observability of latent structures present in the problem. This will heavily depend on the number of observations of  $\mathbf{Xa}$  relative to those in  $\mathbf{Xb}$ , and the amount of information on the latent variable structure that  $\mathbf{Xa}$  brings, relative to that in  $\mathbf{Xb}$ ,  $\mathbf{Ya}$ , and

**Yb.** If **Xa** does not bring significant new information into play, and because a JYPLS model is fitting two sites simultaneously, it is then expected that the estimate obtained with JYPLS will be no better than the estimate with EPCR, because the target site (**Xb**) and product data (**Ya** and **Yb**) already contains all the information necessary to find the latent structures involved.

To illustrate this, consider the data sets described in Table 5.3 (and available in the electronic supplement to this thesis); all data structures are constructed from three latent variables (Appendix 7), and **Xc** is built in such way that the effect of the third latent variable on it is very low (notice low R2cum for **Xc** in Table 5.3); all observations are independent and normally distributed. In Table 5.3 also notice how the predictability of **X** and **Y** decrease dramatically when only 10 observations are used.

Matrix	Observations	Variables	R2cumPCA	Q2cumPCA
Xa	100	8	100%	96%
Ya	100	6	100%	85%
Xb	100	13	97%	94%
Yb	100	6	92%	64%
Xb (10obs)	10	13	97%	67%
Yb (10obs)	10	6	91%	-59%
Xc	100	13	75%	53%
Yc	100	6	92%	62%

**Table 5.3 Data structures used to compare EPCR with JYPLS**

Three transfers are performed with these data sets: *i*) Transfer all observations of **Ya** to site B, *ii*) transfer all observations of **Ya** to site B, using only 10 observations from site B, and *iii*) transfer all **Ya** to site C; and each of these three transfers are done with EPCR and with JYPLS. The notation used to refer each of the estimates for are listed in Table 5.4

Transfer	Estimate with EPCR	Estimate with JYPLS
Ya → B	$\hat{\mathbf{X}}\mathbf{Bnew}_{EPCR}$	$\hat{\mathbf{X}}\mathbf{Bnew}_{JYPLS}$
Ya → B (10 obs in B)	$\hat{\mathbf{X}}\mathbf{Bnew10}_{EPCR}$	$\hat{\mathbf{X}}\mathbf{Bnew10}_{JYPLS}$
Ya → C	$\hat{\mathbf{X}}\mathbf{Cnew}_{JYPLS}$	$\hat{\mathbf{X}}\mathbf{Cnew}_{JYPLS}$

**Table 5.4** Notation to refer the estimates in the JYPLS-EPCR comparison exercise

The solutions to all three transfers ( $\mathbf{XBnew}$ ,  $\mathbf{XBnew10}$  and  $\mathbf{XCnew}$ ) are known before hand, and because all X and Y spaces have the same effective rank, the solution to the transfer of each observation is unique, and therefore, to benchmark the estimate against the know solution is valid. A square prediction error (Eq. 5.80) is calculated for all observations, for each estimated matrix in Table 5.4.

$$\begin{aligned}
SPEi_{EPCR} &= (\hat{\mathbf{X}}\mathbf{Bnew}_{EPCR} - \mathbf{XBnew})^2 \\
SPEi_{JYPLS} &= (\hat{\mathbf{X}}\mathbf{Bnew}_{JYPLS} - \mathbf{XBnew})^2 \\
SPEii_{EPCR} &= (\hat{\mathbf{X}}\mathbf{Bnew10}_{EPCR} - \mathbf{XBnew10})^2 \\
SPEii_{JYPLS} &= (\hat{\mathbf{X}}\mathbf{Bnew10}_{JYPLS} - \mathbf{XBnew10})^2 \\
SPEiii_{EPCR} &= (\hat{\mathbf{X}}\mathbf{Cnew}_{EPCR} - \mathbf{XCnew})^2 \\
SPEiii_{JYPLS} &= (\hat{\mathbf{X}}\mathbf{Cnew}_{JYPLS} - \mathbf{XCnew})^2
\end{aligned} \tag{5.80}$$

and three analysis of variance (ANOVA) are performed: a) between  $SPEi_{EPCR}$  and  $SPEi_{JYPLS}$ , b) between  $SPEii_{EPCR}$  and  $SPEii_{JYPLS}$  and c) between  $SPEiii_{EPCR}$  and  $SPEiii_{JYPLS}$ . The mean values for the SPE's for each problem are given in Table 5.15.

For these three ANOVA exercises the  $p$  value obtained is close to zero (7.6694e-013, 0.0049 and 3.0732e-004 respectively) and hence, the null hypothesis that each of the compared SPE's to be drawn from the same population is rejected (Mathworks 1999).

Transfer	$mean(SPE_{EPCR})$	$mean(SPE_{JYPLS})$	$mean(SPE_{EPCR})$ $- mean(SPE_{JYPLS})$
Ya → B	39.122	875.06	-835.94
Ya → B (10 obs in B)	1707.9	1101.4	606.48
Ya → C	5625.5	2657.4	2968

**Table 5.5 Mean square prediction error for all the estimates in Table 5.4**

For the first transfer case ( $Y_a \rightarrow B$ ), the mean square prediction error (Table 5.5) obtained with EPCR is less than the one obtained with JYPLS; in this case, all the information about the latent structures was already in  $X_b$  and  $Y_J$  and the inclusion of  $X_a$  leads to a poorer estimate of  $XB_{new}$  most likely due to the additional need to model  $X_a$ . Figure 5.8 illustrates 3 of the known solutions for the observations transferred in the case with their estimates with EPCR and JYPLS. Notice in this Figure (5.8) that even though the estimate for each of the variables in  $XB_{new}$  with JYPLS is not as good as the one obtained with EPCR, the former estimate is still very close to the true solution.

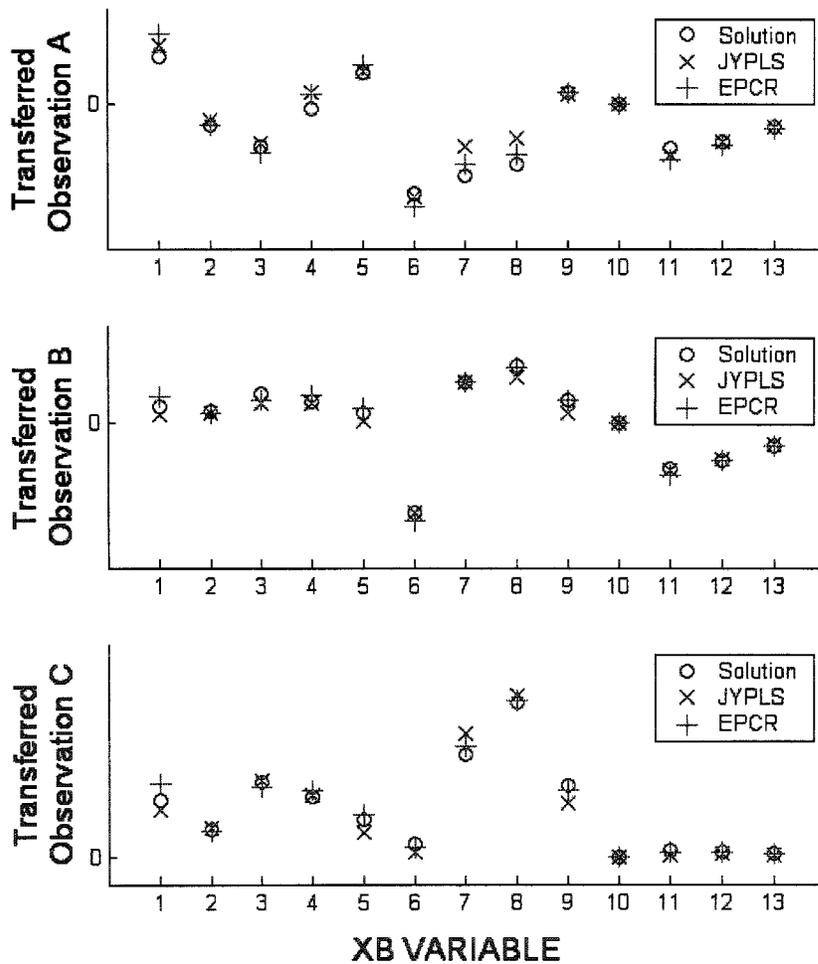


Figure 5.8 Case  $Y_a \rightarrow B$ : Known and estimated values for  $XB_{new}$  for EPCR and JYPLS for three observations at random

For the second transfer case ( $Y_a \rightarrow B$  10 obs in B), the mean squared prediction error (Table 5.5) in JYPLS is less than the one obtained with EPCR (JYPLS outperforms EPCR). Using only 10 samples from plant B results in poor estimation of latent structure in the data (Q2 in Table 5.3 drops from 94% to 67%). This will directly affect the EPCR model which (for modeling the X space) only focuses in the latent structures observable from  $X_b$  and  $Y_j$ . The inclusion of the observations from the much larger  $X_a$  brings more information about the latent structures giving as a result a better estimate for the new conditions in plant B. Figure 5.9 illustrates the known and the estimated values of  $XB_{new10}$  for three observations at random for both techniques.

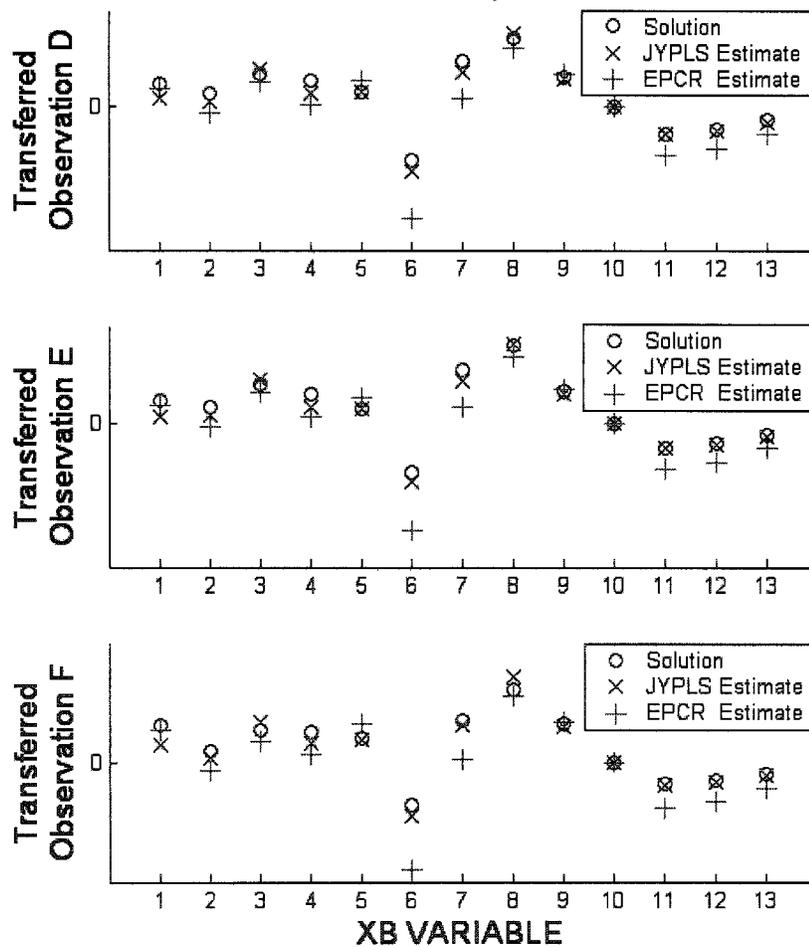


Figure 5.9 Case  $Y_a \rightarrow B$  (10 obs. in B): Known and estimated values for  $XB_{new10}$  for EPCR and JYPLS for three observations at random

Finally, for the third transfer case ( $Y_a \rightarrow C$ ), the difference between JYPLS and EPCR is even greater. In this exercise, the inclusion of  $X_a$  reinforces the latent structure, which is poorly represented in  $X_c$  (particularly the third LV), and hence the solution with the combined spaces (JYPLS) outperforms the one obtained when only site C is considered. Three random observations for this transfer are illustrated in Figure 5.10, notice in this plot, how the estimate improves for JYPLS for variables 5,7 and 9 (which are the ones related with the third latent variable).

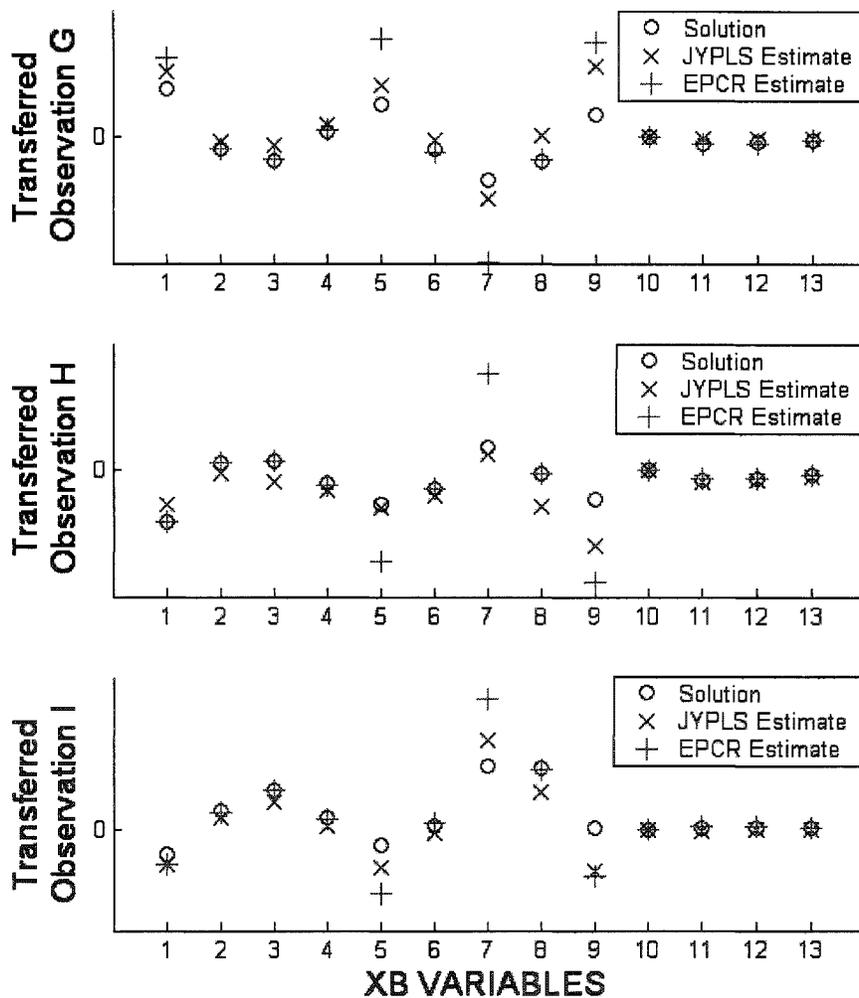


Figure 5.10 Case  $Y_a \rightarrow C$ : Known and estimated values for  $X_{Cnew}$  for EPCR and JYPLS for three observations at random

The estimates for the new conditions obtained with JYPLS are not always better than those obtained with EPCR. However, the scenario where EPCR was better (which could be considered a “best case” scenario for EPCR) was when all the information about the latent structure was already in XB and the data was nearly noise-free. But the estimates obtained with JYPLS were not too far from those obtained with EPCR.

In practice, the case of having a *source* site with rich and precise information and a *target* site which contains more uncertainty is typical of the scale-up between a pilot plant and a full scale plant. A pilot plant will likely have many measurements from the system (with well calibrated and non defective sensors) and the runs will be performed under tight control and with very low unwanted disturbances entering the system; while the full scale plant may have less observations and measurements (or less sensitive sensors) and the system will likely be exposed to more unwanted disturbances. In this scale-up case, the JYPLS model will enrich the poor structured data from the full scale equipment with the well conditioned and structured information from the pilot plant, thereby providing a better transfer.

In the following section, an example from simulated data and an industrial case from a pulp and paper plant are presented.

### 5.7. Examples

Following in this section are two examples to illustrate the use of the JYPLS model to solve the product transfer problem. The first example is built using the detailed deterministic simulation for a fluidized bed linear low density polyethylene (LLDPE) process, developed by McAuley *et al.* (1990).

The second example is constructed with data from an industrial pulp and paper site and a pilot plant used where the product being produced at the mill is often reproduced in the pilot equipment to calibrate analytical instruments used in the process.

### 5.7.1. Simulation of an LLDPE Process

LLDPE resin is commercially characterized by two properties: melt index (MI) and density (D). In the work by McAuley and MacGregor (1992) several process conditions are considered to achieve the best possible path for the transition from grade to grade. The product transfer problem does not consider dynamics, only the final steady-state. Without considering the dynamics, the fluidized bed reactor model consists of two process variables: hydrogen and butene flow. These can be manipulated to achieve fairly independent changes in MI and D.

As it is, this model does not make the best example to consider since it is a very low dimensional system (2 by 2) where the variables for each space are independent of each other (since MI and D are close to being independent of each other). However, if artificial correlation is introduced in the quality variables, it becomes a good example for illustration purposes since the results obtained with the JYPLS model can be contrasted with the results obtained with a highly complex non-linear controller for MI and D. This controller was built into the model in the work by McAuley and MacGregor (1993) and will be used to obtain *exact* solutions for the product transfer problem and will be used as a benchmark.

For this example two different plants are used with the characteristics listed in Table 5.6. Besides the differences in the capacity, the plants operate at different temperature due to differences in heat transfer equipment. Different impurity levels are used which is to be expected if each plant has its own material supplier, and differences in the catalyst kinetics are also used assuming that the plants use catalyst from a different

source. For plant A, the same kinetics as the ones used in the original model by McAuley *et al.* (1990) are used; for the second site the formation, initiation and transfer kinetics for active site 2 of the catalyst is modified.

Property	Plant A	Plant B
Capacity	70 tons	40 tons
Temp.	343 K	350K
Impurities	$5.5 \times 10^{-4}$ mol/sec	$3.0 \times 10^{-4}$ mol/sec
Kinetics	As in McAuley <i>et al.</i> (1990)	Site 2 is modified to be less active

**Table 5.6** Characteristics of the sites for the LLDPE example

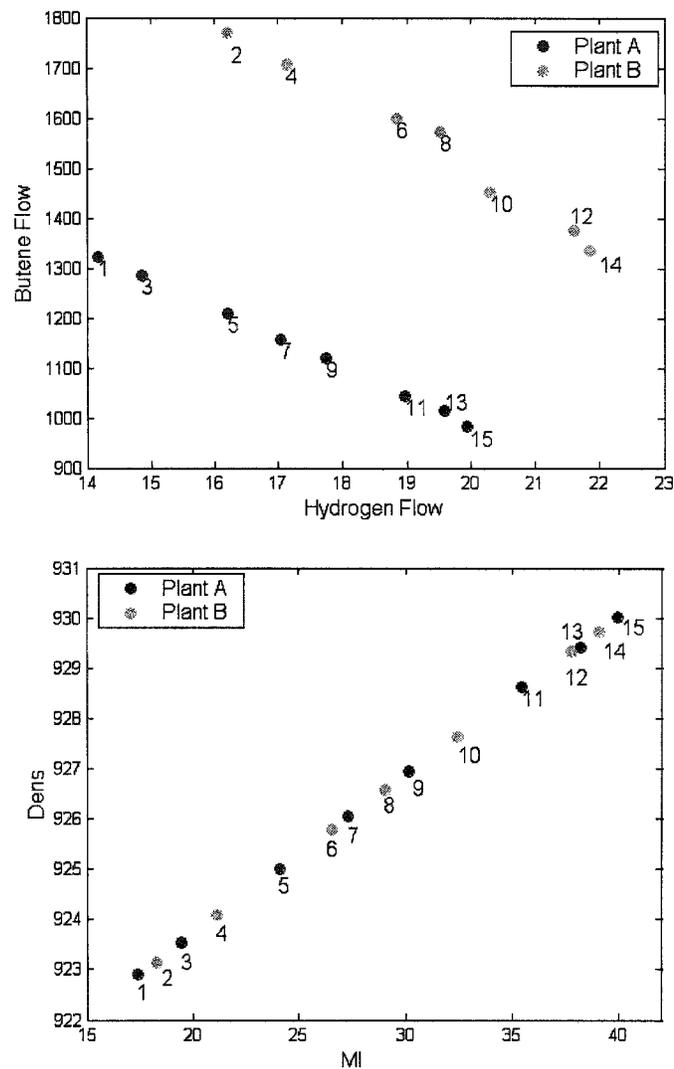
Once again, for illustration purposes, a linear relation between MI and D is introduced to create a one dimensional latent space in the Y's. The author acknowledges this is not a representative scenario of the commercially available LLDPE grades. What is representative is the fact that for a set of quality measurements, there usually is a lower dimensional latent space in the Y's. Because of the linear relationship in the Y's and the low dimensionality of the problem (one) the results obtained by the JYPLS model approach are the same as those obtained with the model inversion approach (Jaeckle and MacGregor, 2000b).

To build each observation in the data sets 1000 points of steady-state normal operating conditions (per grade) are averaged, each of these 1000 points includes disturbances; the non-linear controller is used to calculate the hydrogen and butene flow required in each case. For all cases in this example, the X matrix is formed with the values for hydrogen and butene flow and the Y matrix contains the values of MI and D.

*LLDPE Transfer - case 1*

This is the most simple of the cases presented; the set consists of 15 grades (Fig. 5.11) odd number grades are produced in plant B and even number grades are produced in plant A. Notice that, even though the properties are very close to each other, the process conditions to obtain them are very different from one plant to the other due to the differences mentioned before (Table 5.6). A JYPLS model is built with the following diagnostics:  $R2X_{A\text{cum}} = 99.52\%$ ,  $R2X_{B\text{cum}} = 99.98\%$ ,  $R2Y_{A\text{cum}} = 99.57\%$ , and  $R2Y_{B\text{cum}} = 99.60\%$  with only one significant component.

The exercise now is to transfer all grades from plant A to plant B, and vice versa. The new conditions  $\mathbf{X}_{A\text{new}}$ , and  $\mathbf{X}_{B\text{new}}$  are shown in Fig. 5.12 (top) as well as the exact solution for each of the cases (obtained by using the non-linear controller in close loop). These  $\mathbf{X}_{A\text{new}}$ , and  $\mathbf{X}_{B\text{new}}$  conditions are now passed through the simulation of each plant with the non-linear controller in open loop. The quality of the products obtained in plant A and B when running under the conditions  $\mathbf{X}_{A\text{new}}$  and  $\mathbf{X}_{B\text{new}}$  respectively, are referred to as  $Y_{B\text{inA}}$  and  $Y_{A\text{inB}}$ , these quality properties obtained are shown in Fig. 5.12 (middle and bottom).



**Figure 5.11 Operating conditions and quality produced for simulated LLDPE transfer case 1**

Despite the simplicity of the case it does illustrate the fact that the solution obtained using latent variable modeling (in this case using the JYPLS model) is very close to the *true* solution, which in this case required a very complex deterministic model and a non-linear controller. In this particular case, none of the transferred grades are outside the region already covered by each of the plants.

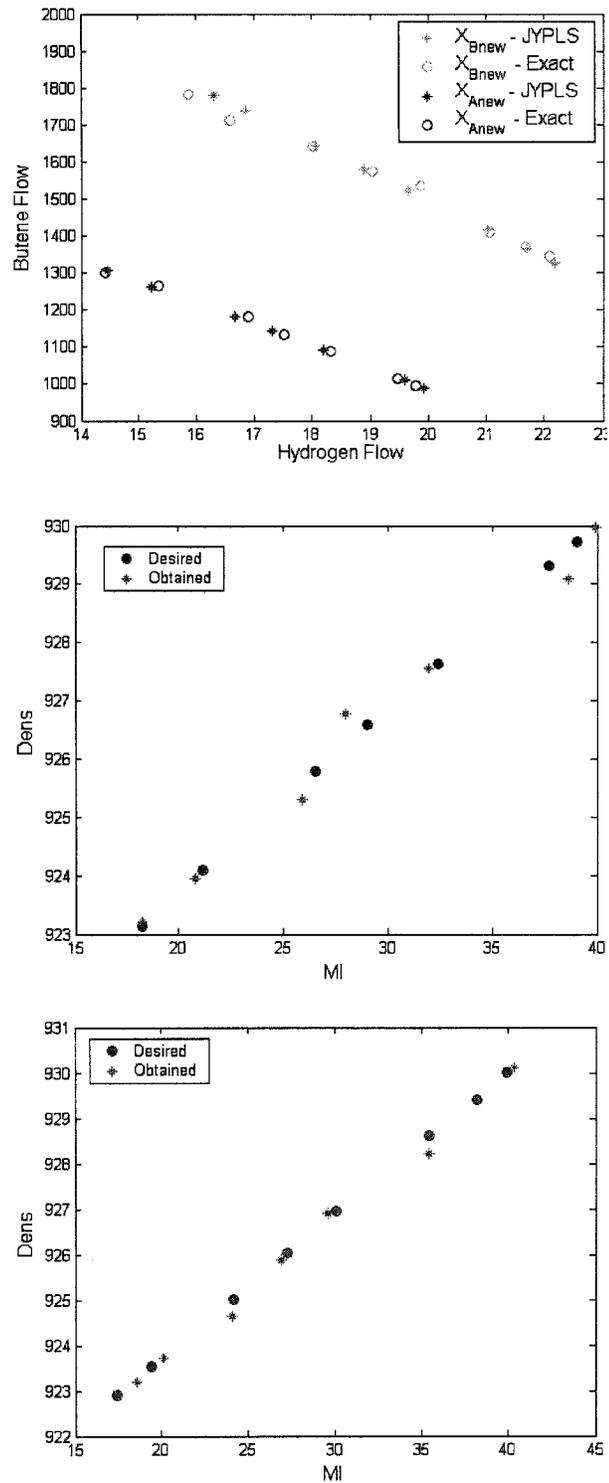
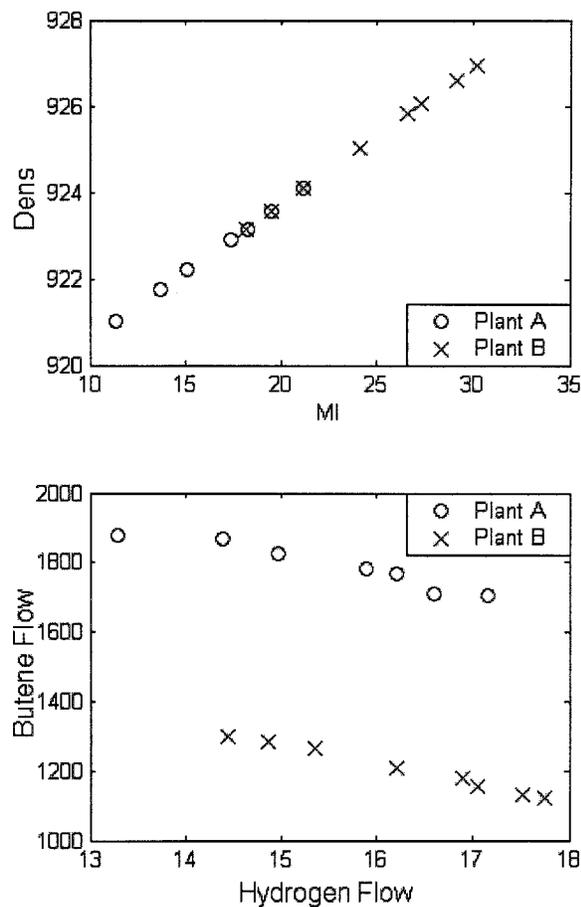


Figure 5.12. New operating conditions  $X_{Anew}$  and  $X_{Bnew}$  (top), and quality after transfer ( $Y_{AinB}$  middle,  $Y_{BinA}$  bottom) for LLDPE case 1

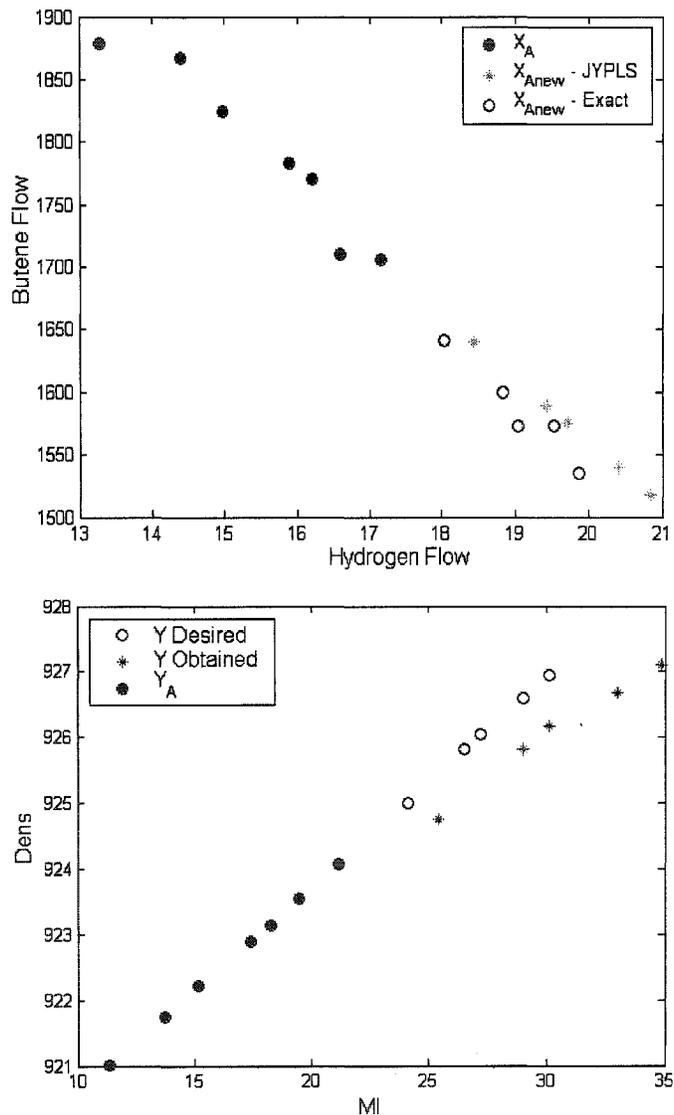
*LLDPE Transfer - case 2*

For this example, the data set consists of 15 grades, three of which are common to both sites; the product quality properties as well as the process operation are shown in Figure 5.13. The exercise is to estimate the process conditions in plant A to produce the five grades of plant B which are not yet being produced in plant A. Each of these new grades represents an extrapolation from existing plant A grades.



**Figure 5.13 Operating conditions and quality for LLDPE case 2**

The JYPLS model is built ( $R2X_{Acum} = 98.20\%$ ,  $R2X_{Bcum} = 99.80\%$ ,  $R2Y_{Acum} = 99.12\%$ , and  $R2Y_{Bcum} = 99.86\%$ ) with one significant component. The same notation as in last case is used; the new operating conditions estimated for plant A ( $X_{Anew}$ ) are now passed thru the simulation of plant A with the non-linear controller in open loop. The obtained product quality properties are shown in Figure 5.14.



**Figure 5.14 Operating conditions (top) and quality obtained after transfer for LLDPE case 2**

The new predicted operating conditions are shown together with the exact solutions (obtained by using the non-linear controller in closed loop). Notice that the new exact conditions have a slight non-linear behavior with respect to the ones already in use in plant A. Because this non-linearity is not evident in the dataset used to produce the model and because the new grades represent an extrapolation from plant A, the solutions obtained with the latent variable approach are not as good as the ones obtained in the case 1. However, the solutions obtained by using the JYPLS model are a good starting point in the search of the true (or *exact*) solution; this by itself may be saving the engineers months of experimentation.

### 5.7.2. Scale-up of an Industrial Pulp Digester from Pilot Plant Experiments

This example is build with industrial data, collected from a full scale pulp digester with a nominal capacity of  $249\text{m}^3$ , and from a pilot digester tank with a capacity of 20 L. Data on ten different pulp grades are available. In the full scale plant, several dozen runs where averaged to build each observation per grade. In the pilot plant, two or three runs per grade were used to construct each observation.

The reactors are used to produce pulp from wood chips. The process in the full scale digester consists of several stages: *a*) Charging of the raw materials; *b*) first temperature ramp; *c*) second temperature ramp; *d*) cooking; and *e*) removal of product from the equipment. For this process, only stages *c* and *d* are considered since all the rest of the stages are automated and equal for all grades of pulp. The process in the pilot equipment is much simpler (due to the volume of the tank). There is no charging stage (since it is almost instantaneous), the complete cycle is considered for the pilot plant data.

Grades of pulp are characterized by two properties: kappa number and viscosity. To achieve a change in these characteristics, three conditions (mainly) are manipulated: a) the initial acidity of the liquor, b) the cooking temperature and c) the cooking time. Due

to limitations in the data, the acidity of the liquor is not considered in this study (no acidity measurements available). However, if the acidity data was available, then it could be taken into account as a constraint in the design since this is a difficult condition to adjust in the full scale equipment. The trajectories will then be computed to be consistent with the given initial acidity in the batch (assuming that the temperature trajectory has an effect strong enough to compensate the effect of the initial acidity). Unfortunately the acidity levels were not available in the data and therefore the only manipulated variables considered for both sites are the temperature profile and the cook time. Both of these process conditions are contained in the temperature profiles from each site. By averaging many runs per grade the effect of varying acidities should be averaged out.

Data for ten grades of pulp (total) are available (Fig. 5.15); six of which are produced in the pilot reactor, and 7 in the full scale equipment, with three common grades (D, F and I). The temperature profiles related to each of these grades for the pilot and full scale equipment are illustrated in Figures 5.16 and 5.7 respectively.

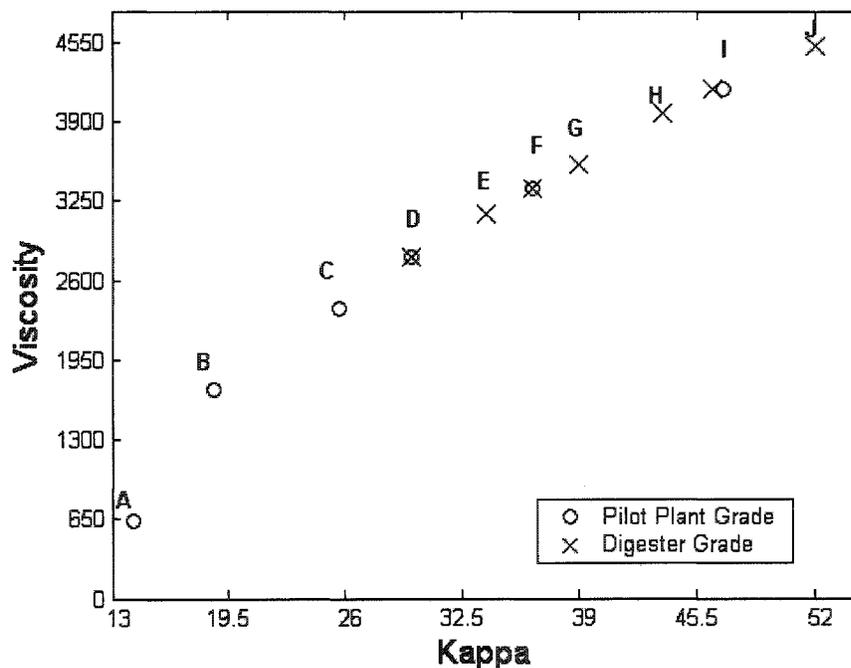


Figure 5.15 Grades of pulp produced at a full scale digester and a pilot plant

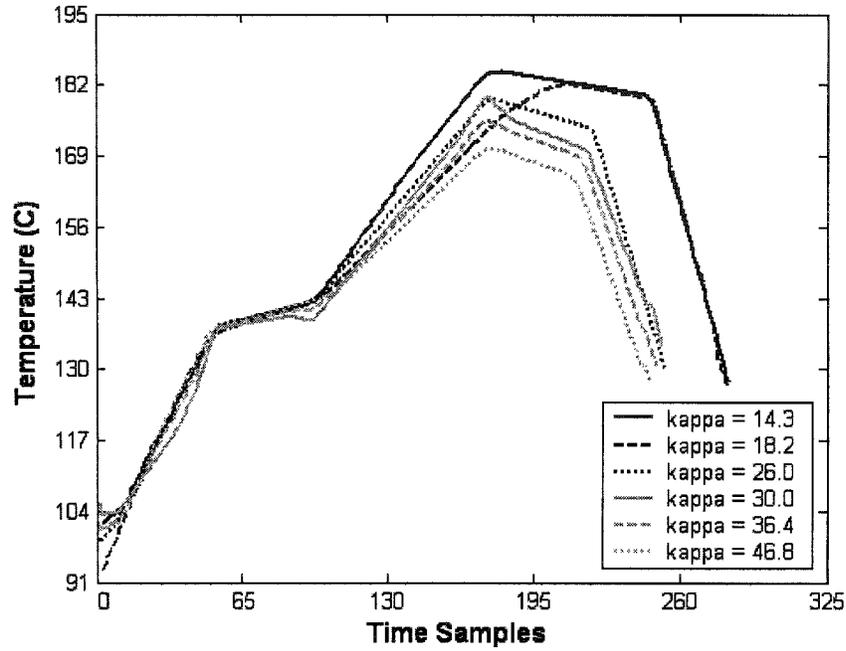


Figure 5.16 Temperature trajectories for the pilot plant

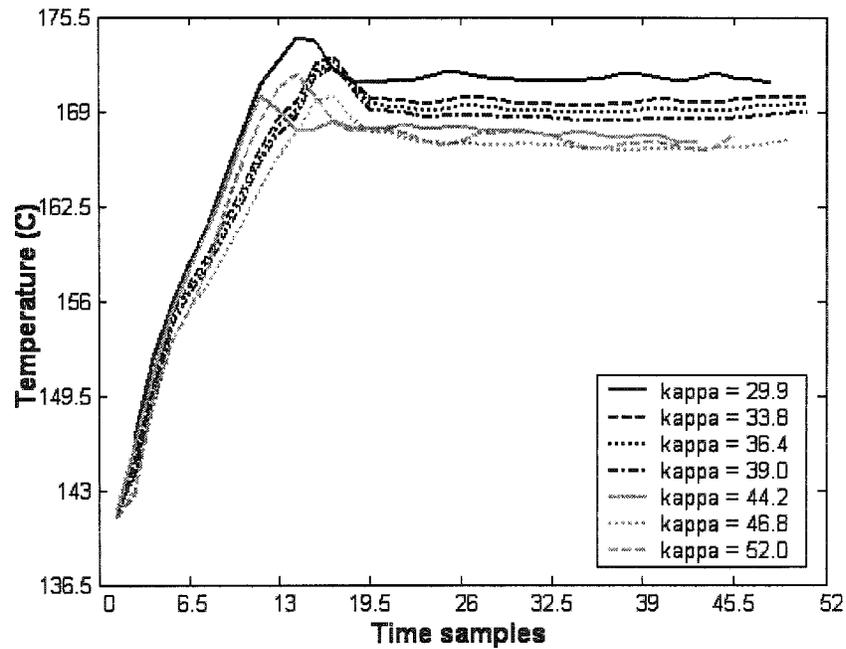


Figure 5.17 Temperature profiles for the full scale digester

The exercise to be performed is the scale-up of grades D, F and I (the common grades for which full real solutions are available). In each case, the grade to be designed is removed from the training data set. Because the process conditions are batch trajectories with different time durations, it is mandatory to align these trajectories to the same number of samples. For simplicity each set of trajectories is re-sampled to 50 and 200 samples for the full scale and pilot scale respectively (linearly interpolating when necessary). The total time is stored in a separate ( $Z$ ) matrix and taken into account by using a multi-block scaling for the process conditions (the reader is referred to section 2.5 for a deeper insight to batch process data alignment).

Performing individual PCA analyses on the aligned temperature trajectories leads to the conclusion that, the temperature profiles can be modeled with two latent variables (approximately 75% explained for the 1<sup>st</sup> LV and 18% for the 2<sup>nd</sup> LV). Doing PCA on the  $Y$  matrices points to the fact that the  $Y$  only requires one latent variable (99% explained in the 1<sup>st</sup> LV). Since the process conditions contain 2 latent variables and the quality properties contains only 1, a null space (see section 4.33) has to be considered.

The correlation between kappa and viscosity is not perfect, and so if the  $Y$ -space is considered as two dimensional, there will be one unique analytical solution to the design problem (Eq. 5.75). However, as discussed in section 4.3.3 this solution will be very ill-conditioned as there is effectively a pseudo-null space. Other solutions can be found along the pseudo-null space, but (as illustrated in Fig. 4.10) these solutions will have a prediction error different than zero. It is left to the judgment of the user to decide whether these designs are acceptable or not.

For each design exercise, there will be one *analytical* solution and one solution with *minimal distance to the origin* (which is Euclidean in the EPCR case and Mahalanobis in the JYPLS case). These solutions and the pseudo-null spaces for the scale-up of grade D using EPCR and JYPLS are illustrated in Figure 5.18.

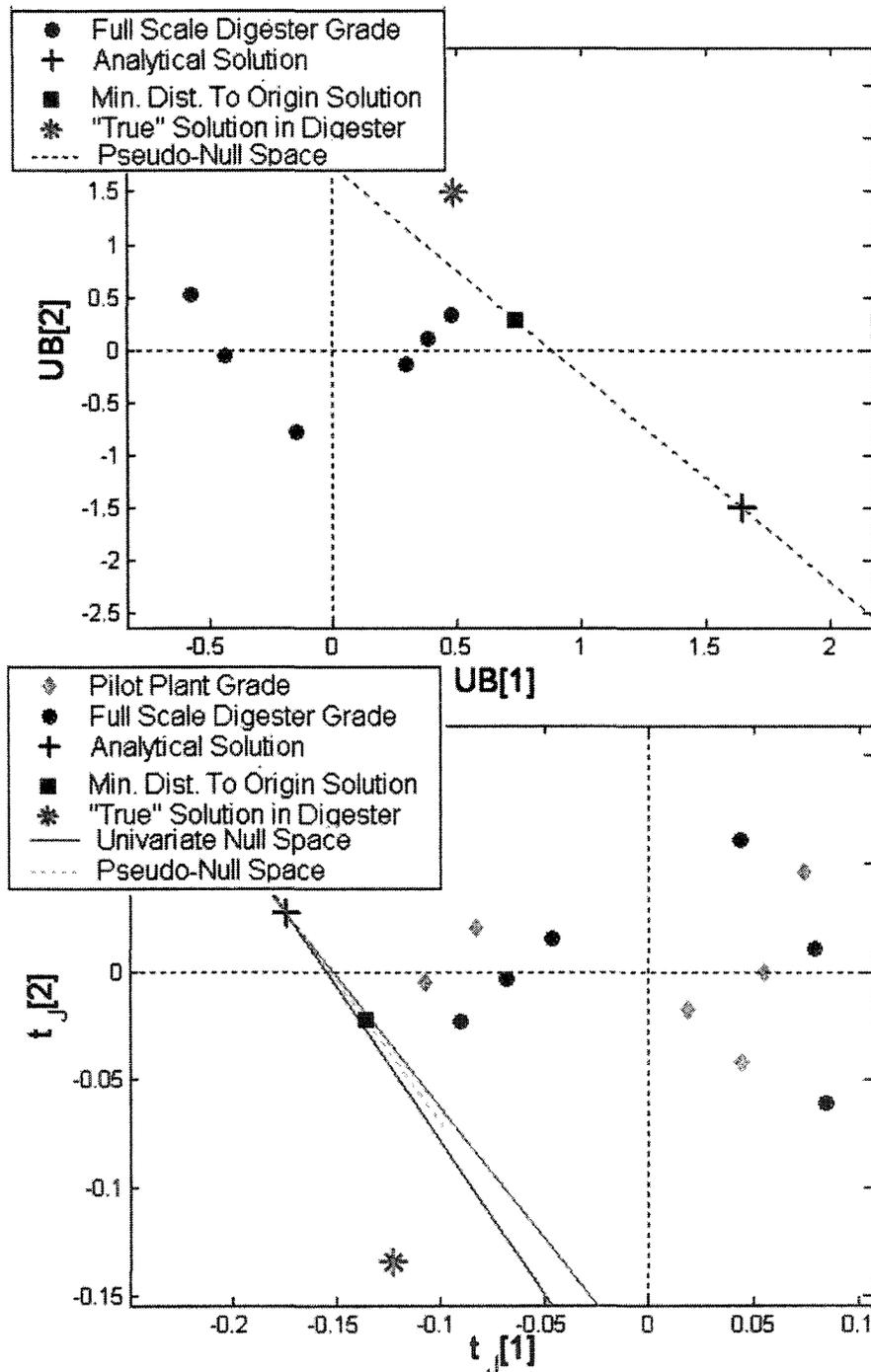
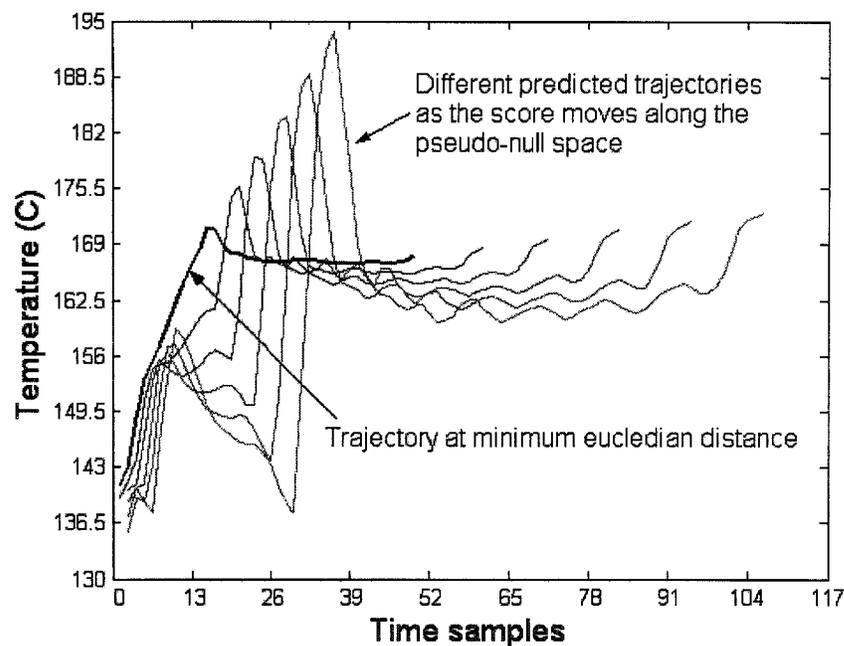


Figure 5.18 Score spaces for the transfer of grade D for EPCR (top) and JYPLS (bottom)

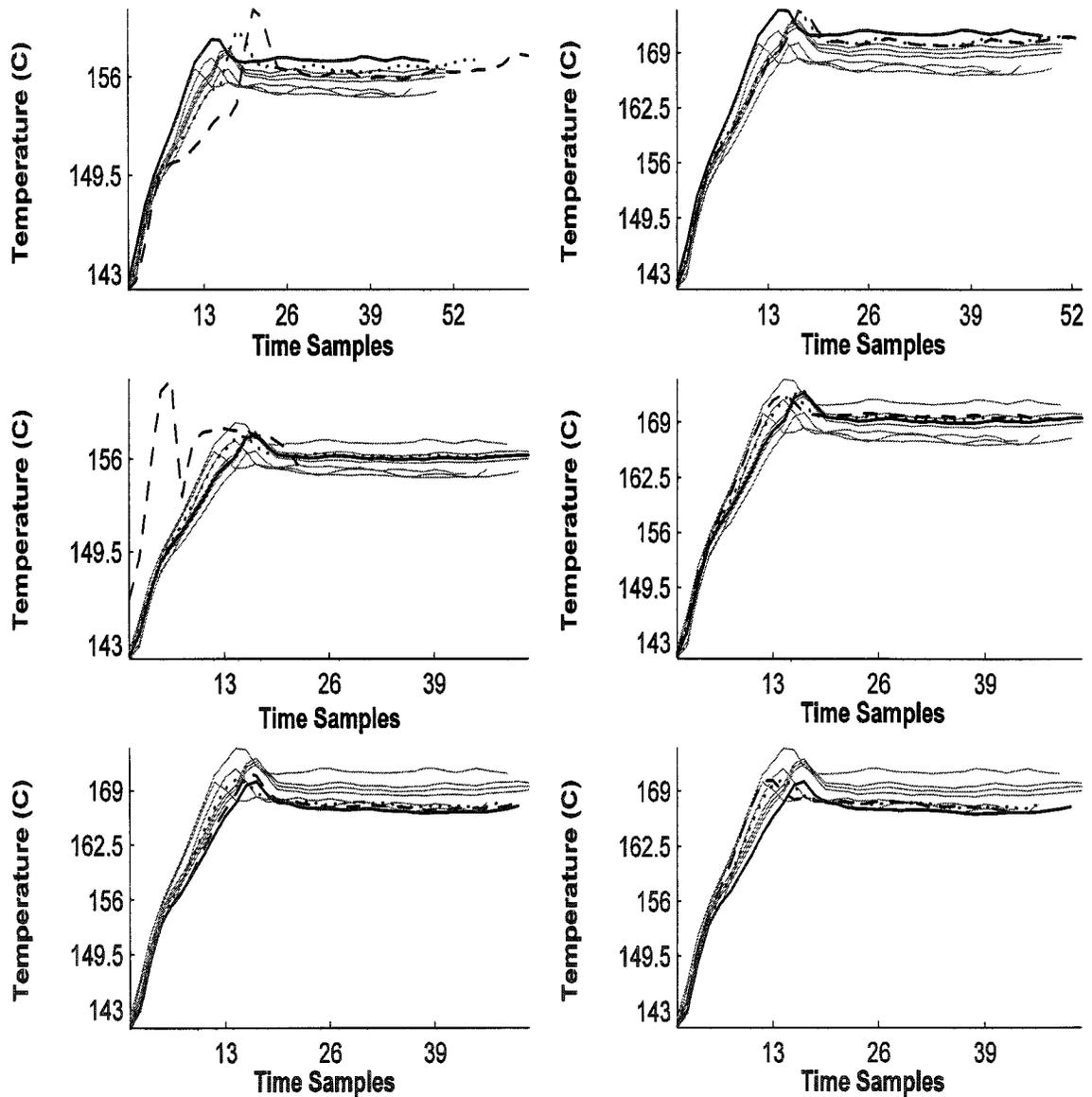
The estimated trajectory depends on which solution along the null space in the score-space is used. As the estimated score ( $\hat{\tau}_{\text{new}}$ ) moves along a null-space (or a pseudo-null space) it is expected that the estimated trajectory will not only change in shape, but also in length (Fig. 5.19), and these changes can be significant. For estimates of the score vector ( $\hat{\tau}_{\text{new}}$ ) that are far from the origin (extrapolations), the predicted temperature trajectories are quite irregular.



**Figure 5.19 Temperature trajectory for full scale digester when the score ( $\tau_{\text{new}}$ ) is moved along the null space**

For each of the scale-up cases, the two estimates (analytical and the null space solution with minimum distance to origin) are computed for each method (EPCR and JYPLS). These estimated trajectories along with the known solutions are shown in Figure 5.20. This comparison is not the optimal way of benchmarking the estimates (the best way is to actually implement those conditions in the plant) but this is the only comparison which could be done in this case.

Notice that the estimated minimum distance trajectories along the pseudo-null spaces (right side plots in Fig. 5.20) are extremely similar for both methods.



**Figure 5.20** Temperature trajectories design estimate for JYPLS (dotted) and EPCR (dashed) for grades D (top), F (middle) and I (bottom). Analytical unique solution (left side plots) and min. distance from origin a long the null space solution (right side plots), known solution (black bold) and grades used to build the model (gray)

The fact that the solutions with minimal distance to the origin for EPCR and JYPLS are very similar does not mean that both methods *perform* identical for this case. As illustrated in section 4.3.3, as the solution moves along the pseudo-null space the prediction of  $y_{des}$  will change as well. This is the strongest differential among the JYPLS suggested in this work and the EPCR technique proposed by Jaeckle and MacGregor (2000b). Table 5.7 lists the prediction for viscosity and kappa for the analytical solution, and the solution with minimal distance to the origin along the pseudo-null space.

Desired Viscosity	Viscosity Predicted JYPLS	Viscosity Predicted EPCR	Viscosity Predicted JYPLS Min. Dist.	Viscosity Predicted EPCR Min. Dist.
2796.3	2796.3	2796.3	2815.15	2895.23
3347.63	3347.63	3347.63	3345.03	3325.53
4175.21	4175.21	4175.21	4177.29	4219.8
Desired Kappa	Kappa Predicted JYPLS	Kappa Predicted EPCR	Kappa Predicted JYPLS Min. Dist.	Kappa Predicted EPCR Min. Dist.
29.76	29.76	29.76	29.52	30.67
36.4	36.4	36.4	36.44	36.21
47	47	47	46.97	47.4
Desired Viscosity	$SPE_{JYPLS}$	$SPE_{EPCR}$	$SPE_{JYPLSNULL}$	$SPE_{EPCRNULL}$
2796.3	0	0	210.86	5791.4
3347.63	0	0	3.85	287.34
4175.21	0	0	2.73	1178.5
Desired Kappa	$SPE_{JYPLS}$	$SPE_{EPCR}$	$SPE_{JYPLSNULL}$	$SPE_{EPCRNULL}$
29.76	0	0	0.03	0.49
36.4	0	0	0	0.02
47	0	0	0	0.09

**Table 5.7 Viscosity and kappa prediction and square prediction error (SPE) for the analytical solution and minimal distance to the origin along the null space solution.**

Even though the temperature trajectories close the origin along the pseudo-null space are very similar for JYPLS and EPCR (right side plots in Fig. 5.20), for the later estimates, the expected value of kappa and viscosity is different from the desired ones

(with much higher SPE than the equivalent for JYPLS). This is most likely to the fact that the JYPLS method improves the predictability of kappa and viscosity by including the information from the pilot plant into the model.

### 5.8. Assessing Parallel Plants using JYPLS

Besides the product transfer, the JYPLS model can also be used in assessing parallel sites which are supposed to produce the same grades of product. This is done by analyzing the loadings from the JYPLS.

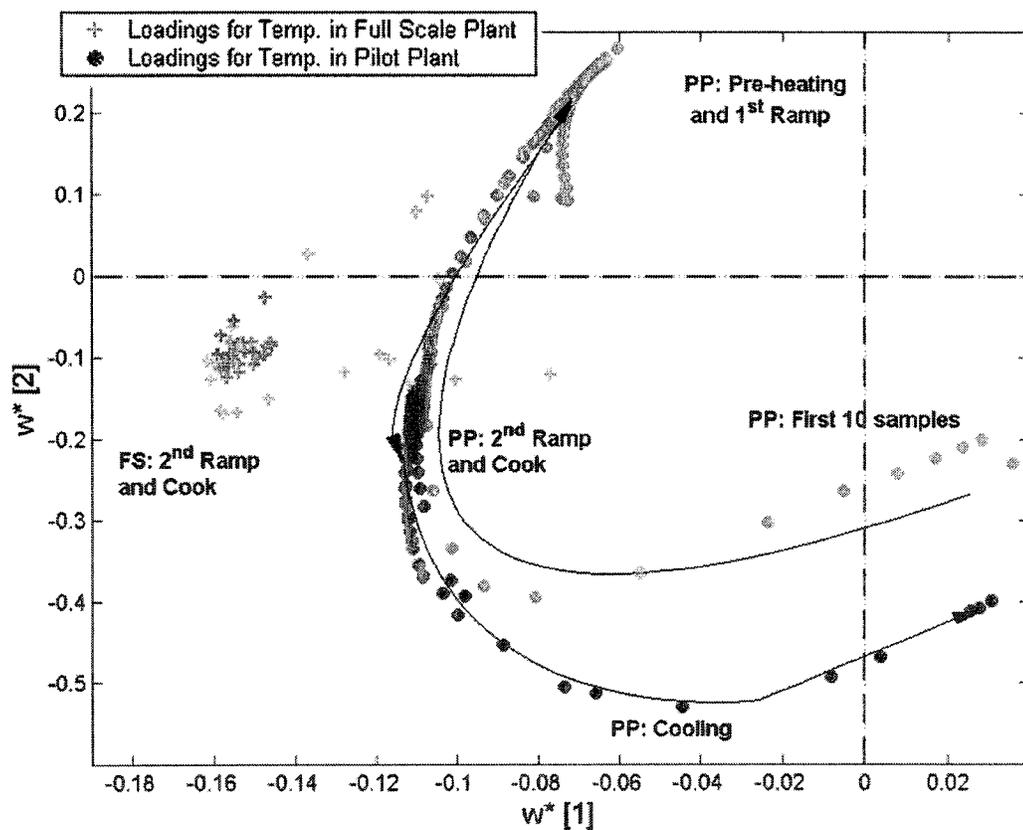
The loadings from two separate PLS models cannot be super-imposed to infer correlation between the variables in site A and those in site B. The reason for this is the fact that each PLS is not capturing the exact same plane of Y, and if they are, the Q loadings from each PLS model might still be rotated versions of each other. Since the plane in Y is not the same, hence the loadings in X cannot be analyzed jointly (even if the Q's for separate PLS models were identical for  $Y_a$  and  $Y_b$ , the loadings on X can still rotate and have different sign).

The loadings from JYPLS model however can be simultaneously plotted in the same graph because the model is capturing that plane in each of the X spaces which best explains the common plane in Y (hence the joint plane in Y is unique to both sites). This feature of the loadings from a JYPLS model is very powerful since it will provide a plot to determine which process variables among sites are related, in their effect to quality. Two examples are provided.

Taking as a first example, the one provided in Section 5.7.2 for the pulp digesters: the stages for the temperature profile included in the data for the full scale digester are the so called “2<sup>nd</sup> ramp” (from first sample to approximately sample number 20) and “Cook” (from sample 20 to the end) . While for the pilot plant, the data included a “pre-heating”

stage (from first sample to sample number 42); a “1<sup>st</sup> ramp” stage (for sample 43 to sample 75); a “2<sup>nd</sup> Ramp” stage (sample 75 to approximately 130); a “Cook” stage (from sample 131 to approximately sample 175); and a “Cool down” stage (from sample 175 to the end) for the temperature profile (see Fig. 5.17).

The fact is that from all these stages, only the “2<sup>nd</sup> ramp” and the “Cook” stage have an influence in the final quality of the pulp. An analysis of the loadings (Fig. 5.21) for the temperature for both sites shows the loadings for the temperature at two stages to be correlated (close in the plot) and separated from the other stages. In Figure 5.21 The arrowed line illustrates the “movement” of the temperature loading as the batch evolves for the pilot digester.



**Figure 5.21** Loadings plot for the temperature in a JYPLS model for the pulp digester example used in Section 5.7.2.

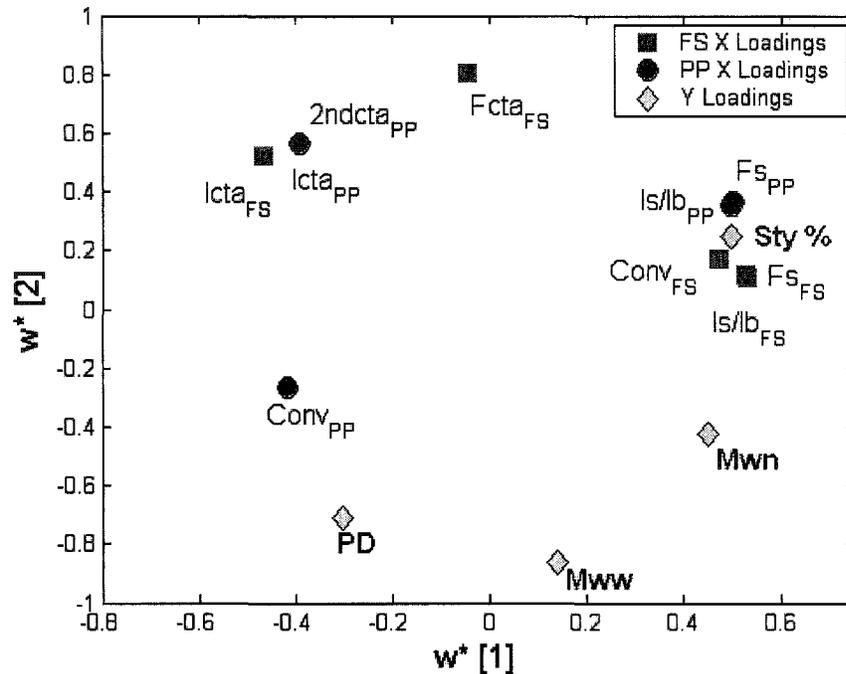
In another study (not entirely shown in this thesis) two batch plants running an emulsion polymerization are simulated using a highly detailed deterministic model developed by Broadhead *et al.* (1985). The simulation is used to determine the process conditions in each of the plants to produce a certain number of grades with similar structure in their final quality. The quality of the latex being produced is determined by the styrene content in the copolymer (Sty %), the average molecular weight in number (M<sub>wn</sub>), the average molecular weight in weight (M<sub>ww</sub>) and the polydispersity of the resin (PD).

Two plants are simulated a full scale and pilot plant. The operation of the two plants is similar: An initial charge of chain transfer agent (CTA), seeded particles, styrene and butadiene are fed to the reactor at time zero. During the run, styrene and butadiene are fed together in semi-batch mode until the tank is full.

The main difference (besides geometry) among the plants is the way the CTA is injected: the full scale plant injects small but continuous flow of CTA during the run, while the pilot plant only does a second injection of CTA (proportional to the initial CTA charge). This operational difference has a strong impact in how the operation in each plant has to be modified to “adjust” the different properties of the polymer.

The operation of the plants is then described by: Initial CTA ( $I_{cta_{PP}}$  and  $I_{cta_{FS}}$ ); ratio of styrene flow to butadiene flow ( $I_s/I_{b_{PP}}$  and  $I_s/I_{b_{FS}}$ ); flow of styrene ( $F_{s_{PP}}$  and  $F_{s_{FS}}$ ); final conversion ( $Conv_{PP}$  and  $Conv_{FS}$ ); and flow of CTA for the full scale plant ( $F_{cta_{FS}}$ ) and 2<sup>nd</sup> injection of CTA for the pilot plant ( $2nd_{cta_{PP}}$ ).

A JYPLS model is built with the data obtained by several runs (which were adjusted by trial and error to produce a set of Y's with the approximately the same correlation structure). An analysis of the loadings plot for both sites (Fig. 5.22) provides interesting insight into the operational “equivalences” among the plants.



**Figure 5.22** Loadings plot for two simulated batch plants producing SBR latex

From an analysis of Figure 5.22, it is clear that:

*i)* The flow of styrene and the ratio of styrene flow to butadiene flow determines the final content of styrene in the copolymer in both sites. Since the loadings for Is/Ib<sub>PP</sub>, Is/Ib<sub>FS</sub>, Fs<sub>PP</sub> and Fs<sub>FS</sub> are all close to the loading corresponding to Sty %. Additionally, for the full scale plant, the final conversion has a strong impact on this property as well.

*ii)* For the full scale plant, the Initial CTA and Flow of CTA (Icta<sub>FS</sub> and Fcta<sub>FS</sub>) are both inversely proportional to polydispersity (PD), but not entirely correlated with each other. This corresponds with the experience of the author in adjusting the polydispersity for this site, the initial charge of CTA was used for “gross” tuning and the flow of CTA was used for “fine” tuning of the polydispersity (PD). For the pilot plant however, the two injections of CTA were correlated and hence could not be used as separate variables to adjust the PD. It was not expected by the author (who is not

knowledgeable in polymer science) to find from this loadings plot that the final conversion in the pilot plant would have a positive effect in polydispersity and could hence be used for fine tuning along with the injections of CTA. This was later verified with the simulation to be true.

The analysis of the loadings from a JYPLS may provide interesting insight into the operational equivalences among sites with different configuration or (like in the case of the SBR simulation) different operational policies. This kind of analysis is unique to the JYPLS loadings plot and cannot be achieved by separate PLS models.

### 5.9. Conclusions

To transfer the production of a grade from one site to other is a difficult industrial problem, and may involve several months of engineering and costly experimentation. To speed up this process will have a positive effect on the business, especially on those that are sensitive to this step of manufacturing like the specialty chemicals sector for whom the scale-up and transfer may represent competitive advantage.

A data driven approach that uses existing plant and quality data from the different sites is developed to achieve this transfer. The JYPLS model is introduced to solve the product transfer problem. The fact that the JYPLS model is capturing the common plane in  $Y$  for all sites, and that this plane is explained by each of the individual  $X$  spaces is a unique characteristic to the JYPLS model and provides a powerful tool to analyze and troubleshoot *parallel* sites with different configuration, and to show how their variables are related with respect to the product family being produced.

The product transfer technique using JYPLS will not always give better results than the previous proposal using EPCR, however in these cases the estimates are not too different, and still represent a good start point if experimental surface response methods

are to be used for *fine tuning* of the operating conditions. However, for cases in particular when there are few samples or limited information on the latent structure in the plant to which transfer is to be made, the JYPLS approach can offer a superior solution. It does so by considering the process data ( $\mathbf{X}_a$ ) on the transfer site to stabilize its latent variable structure.

The features of the JYPLS model were illustrated with an industrial case where the difference between EPCR and JYPLS was not in the estimated operating conditions, but in the expected value for the quality properties.

The JYPLS is powerful tool to solve the product transfer problem, and surely can have other applications in engineering, or chemistry such as calibration transfer.

# Chapter 6

## Conclusions

Multivariate statistical methods based on latent variable (LV) models are applied throughout this thesis to solve problems related to the analysis, operation and monitoring of a batch process. Four different areas are investigated: *i*) the troubleshooting of the process operation, *ii*) the monitoring of new batches, *iii*) the optimal re-design of the operating conditions, and *iv*) the effective transfer of valuable information from one production site to another.

The necessary steps to build an effective monitoring scheme involve *a*) a pre-assessment of data which is representative of the process operation; *b*) taking the necessary steps to correct the process if any unwanted behavior is diagnosed in the pre-assessment; and *c*) re-sample the process for model building and further monitoring. From these three steps, the pre-assessment of the process is critical for a robust monitoring scheme to be built. Chapter 2 is an illustration of such pre-assessment and the kind of conclusions that might come from such an analysis when it is done on a batch system.

The analysis shown in Chapter 2 presents a practical example that illustrates the state of the art of batch process analysis using latent variable methods. This application involves a complex industrial process from which data is available from several sources. The available historical data from the process is analyzed using both conventional and multi-block LV methods in order to determine which of the multiple sources of variability is related with the production of off-specification product.

Several indicator variables are used to synchronize the trajectories, along with a new technique that is applied to compute the used amount of time for a certain evolution level in the batch. Doing so creates an additional profile (referred to as the *time usage*) and is to be included into the multivariate model of the process variable trajectories. This new trajectory is analyzed as an additional variable along with the rest variable trajectories of the batch. The time usage trajectory enables the model to monitor and identify the periods of operation where the batch system consumes more or less time. This piece of information is valuable for process improvement purposes as illustrated in this chapter.

The approach taken to include the *time consumption* differences among batches allows a detailed troubleshooting of the process. By interpreting the several loadings matrices and contribution plots, it is found that the timing of the different stages in which the unit was operated was the source of off-specification product, and not the initial chemical conditions of the raw material as it was thought in the beginning.

Taking into account the time variable is shown (in Chapter 2) to be powerful and necessary to effectively capture the operation of a batch system. And this is not restricted to the pre-assessment (or off-line analysis) stage, but also in the monitoring stage.

Chapter 3 presents a thorough analysis on the implications of using the missing data handling capabilities of the PCA model in monitoring a new run before its completion. Six different missing data methods are considered, extending the previous work by Nomikos and MacGregor (1994) where only one method is considered.

When monitoring a new batch, at any time before the completion of the run, it is possible to “fill in” the future unknown measurements with *missing data* and estimate the final score; there are several methods available for this task. The mechanism behind the estimation of the final score uses an implicit forecast of the unknown samples of the

batch trajectories; this forecast is uncovered and studied in detailed. This study shows that (when filling in with missing data) any of the possible *f step ahead forecasts* is performed using an adaptive-expanding multivariate time series which will evolve non-linearly as more samples are available. This forecasting model will include tailored coefficients for each of the possible *f step-ahead* estimates that can be computed at any sample before the completion of the batch, and the accuracy of such forecasting mechanism is shown to be superior of the one obtained by other means.

The score estimates obtained in the same way are proven to have the best set of properties among the considered methods. The score estimates obtained when the missing data option is effectively used will preserve their orthogonality, stability and coherence throughout the entire batch. Such performance is only obtained by using an appropriate method (such as conditional mean replacement, iterative imputation or projection to the model plane) to handle the missing measurement scenario.

A final comparison exercise in Chapter 3 is performed to contrast the considered methods in the fault detection exercise. For this purpose, a new data set is created with a detailed deterministic simulation of an emulsion polymerization. The goal is to simulate a set of subtle faults which will also resemble realistic situations. The methods are benchmarked in detecting five faults which differ in magnitude and nature. At the end, there is no statistical evidence of any difference in the mean time to detection among methods. The fact that the control limits are tailored for each of the methods might result in similar detection capabilities as shown in this chapter. If there is any difference, an extensive study is required to uncover them.

The first half of this thesis presents solid evidence of the modeling power of a multiway method in capturing the operation of a batch process, not only in its variable time-profiles but also in its initial conditions. These modeling capabilities are used in the second half of the thesis for design purposes.

In previous related work (Jaekle and MacGregor, 1998), latent variable methods were used to design the operating conditions to achieve a desired quality in the product. In Chapter 4, these techniques are now applied to the design of batch process trajectories. This Chapter presents the reformulation of the earlier technique in order to include constraints in the desired quality characteristics; and to consider operational constraints and an optimal criterion in the design of the operational conditions for the batch.

The analysis of the multiple solutions available in the exercise of estimating operating conditions for a certain industrial problem is shown. This is the first work that further illustrates the concept of the null space and its application in estimating process conditions for an industrial case.

The augmentation of the model with the derivatives of the trajectories is proposed to strengthen the information about the trends into the model and hence avoid the analysis of a null space. Such augmentation is proven to greatly enhance the structure of the trajectories and how this structure is passed to the new estimates. Derivative augmented models will not increase or decrease in a significant amount the predictability of the final properties in a PLS model.

The given reformulation of the problem allows the inclusion of an optimal criterion in the sought operating conditions. This feature of the technique is illustrated with an industrial case, where the operating conditions of an emulsion polymerization process are estimated to fulfill the customer requirements for the final product, while consuming the minimal amount of time for its execution.

This new optimal trajectory design methodology represents an attractive alternative to the use of deterministic models and represents the equivalent *off-line*

*technique* to the iterative on-line method proposed by Flores-Cerrillo and MacGregor (2004).

Finally Chapter 5 revisits and solves the product transfer problem studied before by Jaeckle and MacGregor (2000) by proposing a new multivariate latent variable regression method: the Joint-Y PLS model.

The conceptual form of the JYPLS model is shown and its mathematical foundations are established. The analytical solution to the parameter estimation problem is given as an eigenvector-eigenvalue equation. This work provides a modification to the NIPALS algorithm (traditionally used to compute the loadings of a PLS model) to compute the loadings of the new JYPLS model. The proposed algorithm is proven to give under certain conditions, the same solution as the analytical eigenvector-eigenvalue solution.

Several modifications to the JYPLS model are given, such as the multi-block and the quadratic non-linear forms of the model. The extension of the JYPLS model to consider several multiple sites is also given.

After the JYPLS model has been properly defined, a thorough analysis is done to determine under which conditions, the JYPLS model will give a better estimate of the sought process conditions, than the earlier proposed EPCR technique (Jaeckle and MacGregor, 2000). This analysis is done with simulated data. It is shown that the estimate from the JYPLS model will be better in the case where the matrix corresponding to the operating conditions from the *source plant* contains more information (than the rest of the data structures) about the latent structures which are driving the system.

An industrial scale-up and a simulation case are used to illustrate the application of the JYPLS model to the product transfer scenario. The results from these two cases

show that the estimates obtained by using the JYPLS are very close to the “known” solutions (which in the simulated case required a highly detailed deterministic model to find). The concepts detailed in chapter four to estimate optimal trajectories will apply straightforward when using a JYPLS model.

In summary, this thesis contains the following novel concepts and their application to industrially relevant cases:

- The time usage for a batch system and its interpretation
- The interpretation of the forecast equation in a multiway-PCA model for batch trajectories as a multivariate adaptive time series equation.
- The use of forecast-accuracy measures such as the FPRESS and FPRMSE, and the concepts of coherence and stability of the score estimate for a new evolving batch
- The reformulation of the product design equations (Eq. 4.7b, 4.15 ), and the optimal trajectory design formulation (Eq. 4.30)
- The derivative-augmented multi-way model and its application in trajectory estimation
- The Joint-Y PLS model concept
- The solution to the JYPLS parameter estimation problem
- The application of the JYPLS model to solve the product transfer problem
- The application of the JYPLS model in assessing parallel plants

The results shown from such applications are encouraging and should give practitioners (and vendors) the confidence to try them out in commercial solutions.

# Appendix 1

## Analysis and Troubleshooting of Batch Processes using Multi-block Latent Regression Models

This appendix discusses the batch process presented in Chapter 1 and presents the reasoning sequence to get to the same conclusion in regards of the changes needed by the process in order to reduce the production of off-specification product; but by using a single multi-block PLS model (Westerhuis, Kourti, and MacGregor, 1998). Many conclusions can be obtained from multi-block models; however, the discussion in this appendix is conducted to reach the same conclusions as in Chapter 1.

### A1.1 Multi-block PLS models for batch processes

Kourti *et al* (1995) introduced the analysis of batch data with multi-block models in order to include the initial conditions available at the beginning of the batch run, acknowledging that: *i*) these conditions may have a strong impact in the final product characteristics and *ii*) these initial conditions are certainly correlated with the process trajectories.

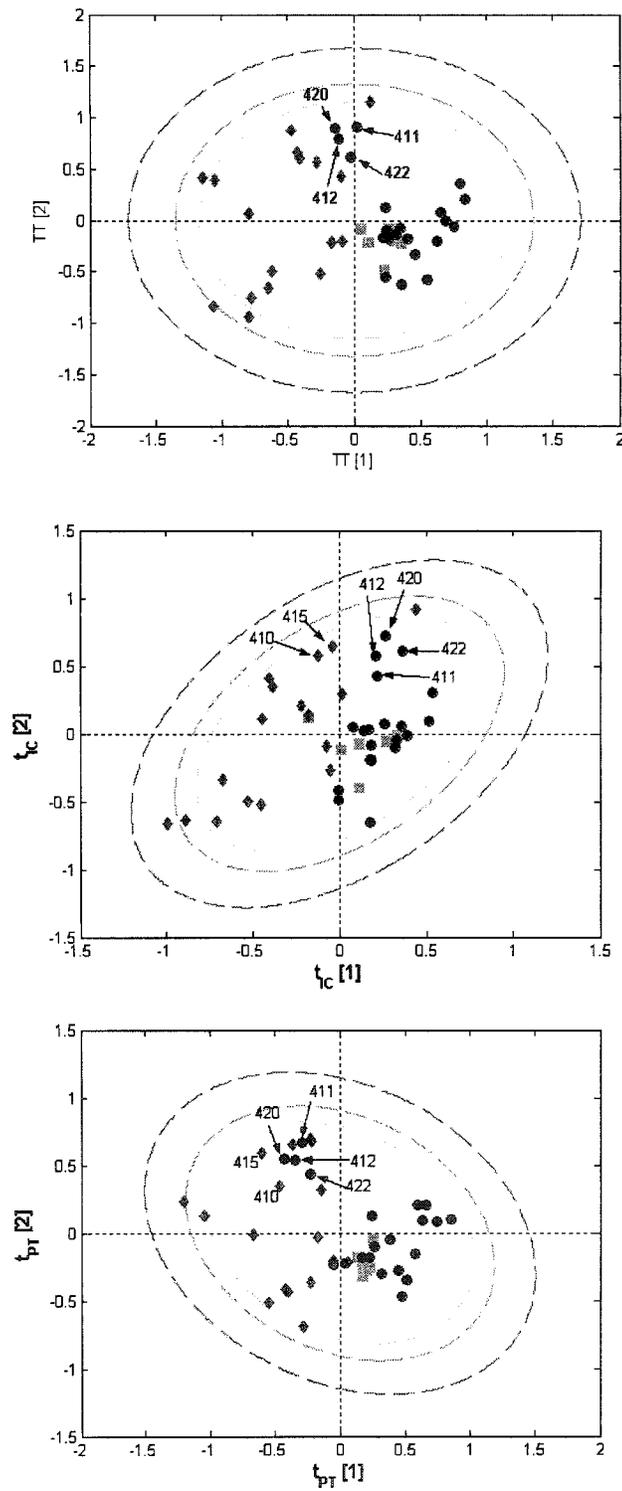
For a given batch data set, there will be a set of  $M$  of initial conditions, and  $J$  process variables measured  $K$  times during the batch (Fig.2.1 and Fig. 2.2), a simple approach to handle this data set is simply to augment the unfolded X matrix (Fig 2.1) by adding the of the initial condition columns to it, and doing PLS. However, it is very likely

that  $M \ll J \times K$  and hence, the PLS model will not extract all the possible information from the initial conditions because these are outnumbered by the  $J \times K$  columns from the unfolded trajectories. By using a multi-block model the weight is being balanced between  $Z$  and  $X$ ; and the  $M$  columns of  $Z$  and the  $J \times K$  columns of the unfolded  $X$  matrix will both have the same weight in the regression model since each block has variance equal to one (Westerhuis, Kourti, and MacGregor, 1998). Furthermore, the weight given to a certain block can be modified to increase the prediction of the model.

In the analysis of the batch data from chapter 1, the 20 initial conditions the 11 variables (sampled 350 times) and the 11 quality variables are used to build a multi-block PLS (MBPLS) model. Two components capture 55% of the quality data, 32% of the process trajectories and 26% of the initial conditions. These two components are enough to appreciate the differences between the product that is within specifications (on-spec) and less value added product, which properties are outside the desired established ranges (off-spec). In the following sections, the model is interpreted to obtain valuable information about the process conditions and how they affect the quality of the product.

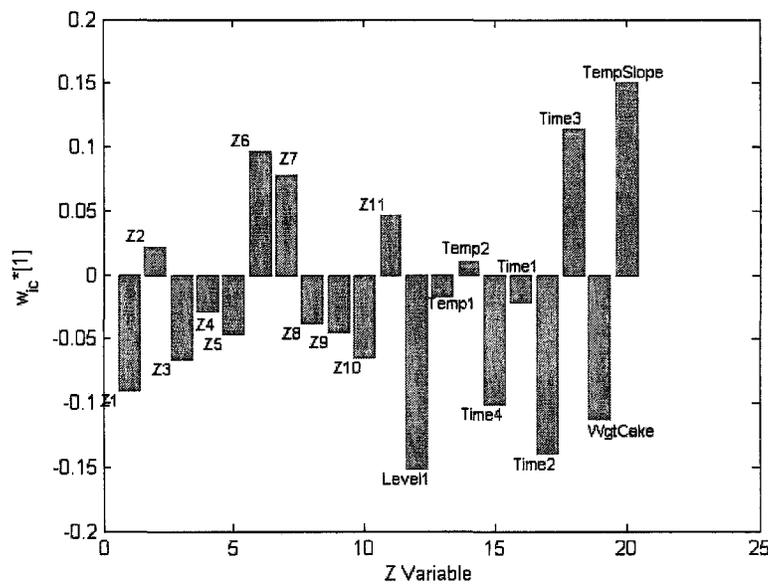
### **A1.2 Score and loadings plots**

Multi-block PLS models have two sets of scores to be analyzed: the super-scores and the block scores. The super scores give an overall picture of the observations accounting the effect from all blocks in explaining the  $Y$  matrix; the block scores will give information about the effect of a particular block on the  $Y$  matrix. Figure A1 (top) shows a score plot for the super-scores; notice that on-spec batches cluster in the positive quadrant of  $t_1$  while off-spec ones cluster in the negative side of  $t_1$  – with the exception of four anomalous on-spec-batches that cluster in the negative side of  $t_1$ . Analyzing the block loadings for  $t_1$  it is possible to find out which characteristics of each block are related to high values of  $t_1$  and hence to on-spec product.



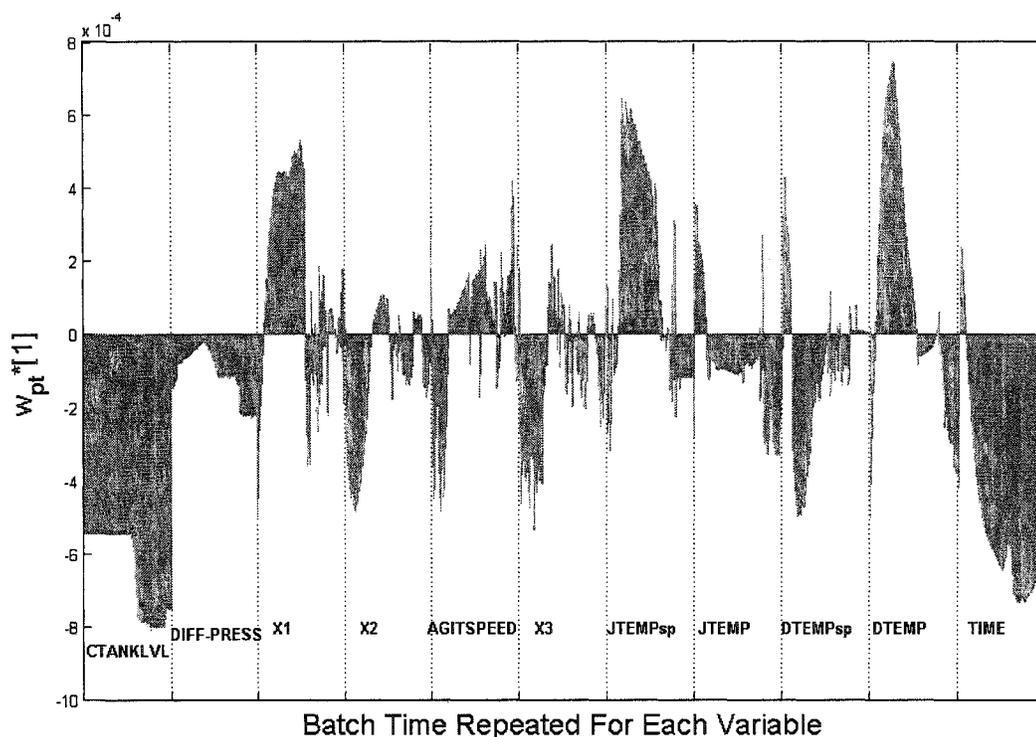
**Figure A1. Score plot for overall scores (top) the initial conditions block scores (middle) and the process trajectory block scores (bottom)**

Analyzing the loadings for the first principal component corresponding to the initial conditions block (Fig. A2) helps to determine when are high values of  $t_1$  obtained. High values of  $t_1$  are obtained with low levels in the collector tank at the end of stage 1 which is possible with low quantities of wet cake fed to the reactor; this plot also suggests that a fast evaporation during stage two (high slope and low time 2) and an extended cooling down time (high time 3) are also related with on-spec product.



**Figure A2. Loadings plot to the first component for the initial conditions**

The loadings for the same component but for the process trajectories (Fig. A3) suggest some of the same conclusions, low levels in the collector tank, and low time usage (fast batches) with high temperatures in the dryer and the heating jacket during the first two thirds of the batch along with high values of  $X_1$  during the same period of time approximately. Notice that these conclusions are no different than those obtained from separated PLS models in chapter 2 of this thesis.



**Figure A3. Loadings plot to the first component for the process trajectories**

### A1.3 Contribution plots for troubleshooting

When plotting the super-scores, four anomalous on-spec batches were spotted. More insight can be obtained by plotting the block scores; notice that these four batches fall within the on-spec region for the score plot corresponding to the initial conditions (Fig. A1. middle) and at the same time fall in the off-spec region in the score plot corresponding to the process trajectories block (Fig. A1. bottom), this suggests that, the reason for these batches to yield on-spec product is not present in their trajectories, but in the information captured by the Z matrix (notice the unusual shape of the confidence limits in these two plots, due to the fact that the block scores are not entirely orthogonal and the limits must be computed using the complete covariance of the block scores).

The troubleshooting exercise is now to figure out which were the characteristics in those four anomalous on-spec batches that were enough to counteract the effect of the

process trajectories and yield good quality in the final product. To answer this, four batches are selected: anomalous batches 420 and 412, and *off-spec* batches 415 and 410 (Fig A1 middle and bottom). These batches cluster together in the score plot corresponding to the process trajectories, and cluster separately in the score plot corresponding to the initial conditions.

Figure A4 is a plot of the contributions to the first component in the process trajectories block scores for the off-spec batches and figure A5 contains similar contributions, but corresponding to the anomalous on-spec batches. Each pair of plots contains the traditional contribution plots where the time effect is appreciated, and the integral under the curve for each variable, which gives an overall idea of which was the process variable that contributed the most. To the best of the knowledge of the author, this is the first time the contributions are presented in such way (integrated with respect to time), a similar variable-wise contribution plot is shown in (Nomikos, 1996) and (Kourti, Nomikos, and MacGregor, 1995), however these contribution plots are to the instantaneous SPE chart.

The contribution plots shown in figure A4 and A5 do suggest that the two off-spec batches and the two anomalous on-spec batches are in the same space of the block score plot for the same reasons: high values of CTANKLVL and TIME, which are contrary characteristics to those found necessary to achieve high values of  $t_1$ .

This comparison between the contribution plots to the first component (the one that separates on-spec from off-spec) for these batches: 420, 412, 415 and 410 validates the fact that the process trajectories for the anomalous ones have the same characteristics of the off-spec batches and hence isolating the sought answer to the initial conditions block.

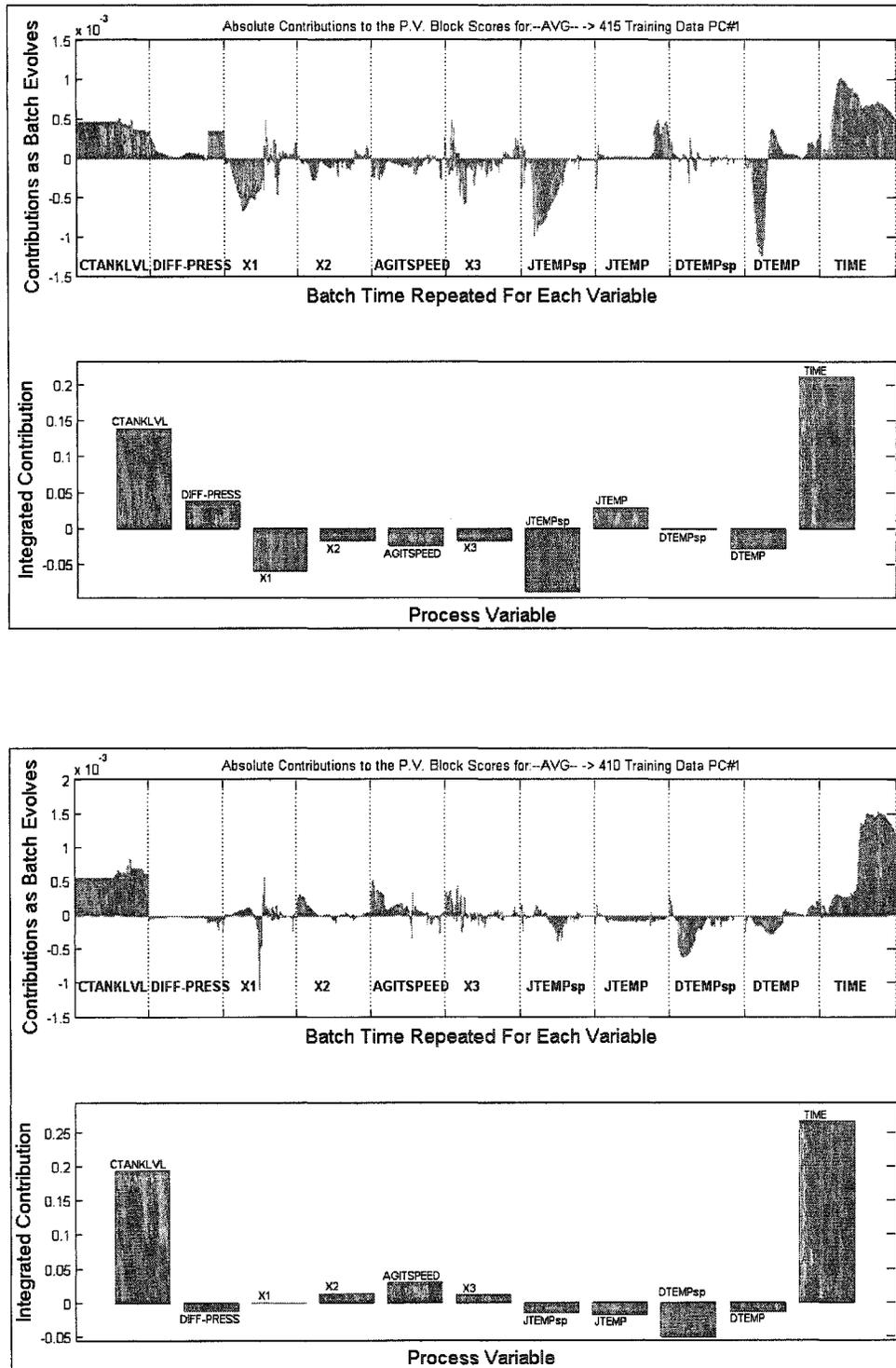


Figure A4. Contribution plots to the process trajectory block scores for two *off-spec* batches 415 (top) and 410 (bottom)

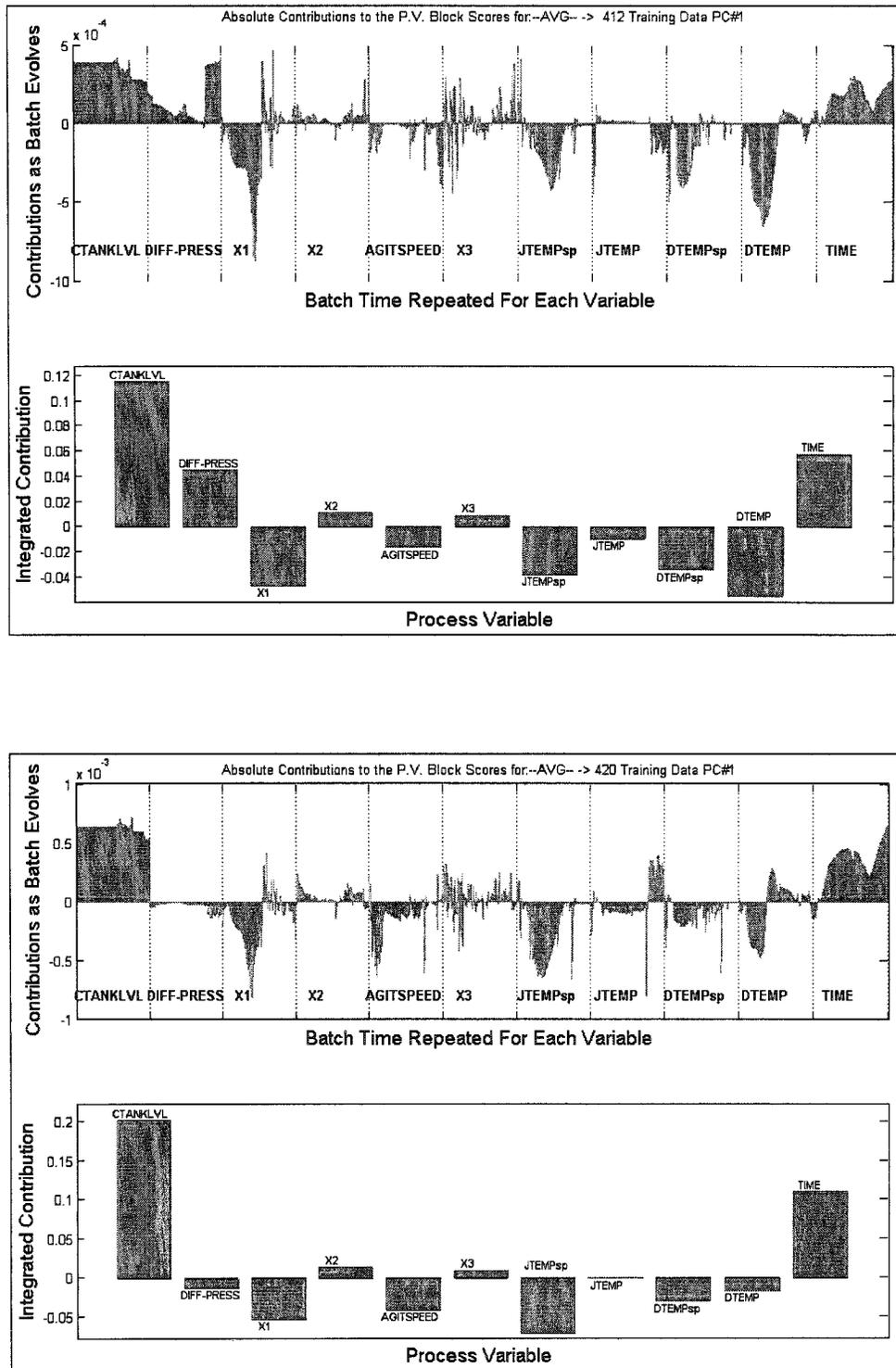
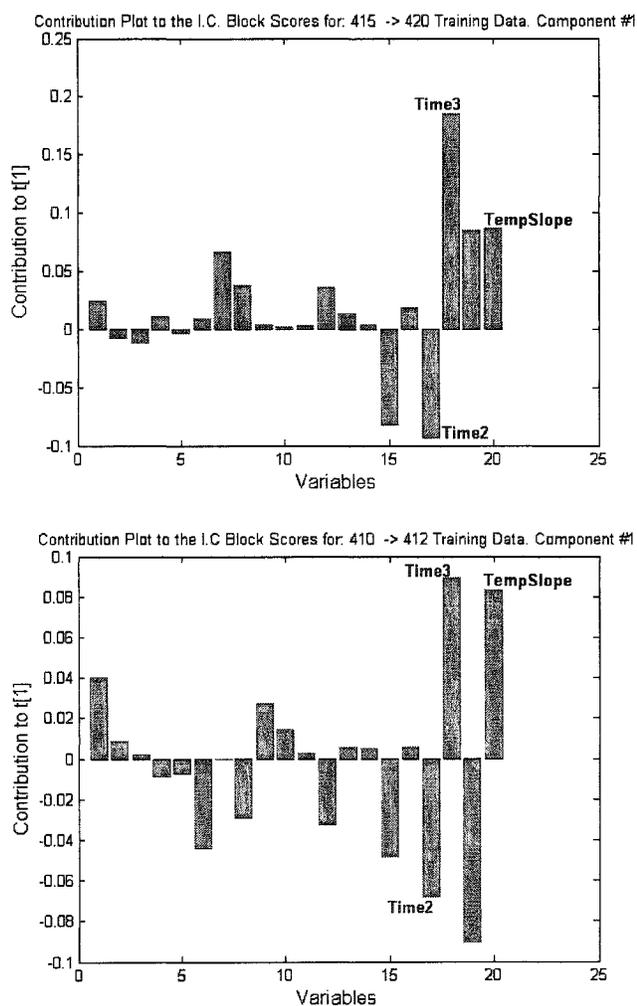


Figure A5. Contribution plots to the process trajectory block scores for two anomalous *on-spec* batches 412 (top) and 420 (bottom)

And so it is that the initial conditions block is now *interrogated* to find out which were the conditions in the anomalous *on-spec* batches such that the final product was within specifications, in spite of having all the characteristics of an *off-spec* batch in the process trajectories. Analyzing the contributions (in the initial conditions block score space) to the movement *from* an off-spec batch *to* an anomalous on-spec batch in the first component (Fig. A6) will point to those conditions that must change in the process variables to achieve such movement in the score space.



**Figure A6. Contribution plots from off-spec batches to anomalous on-spec batches: from 415 to 420 (top) and from 410 to 412 (bottom)**

The two contribution plots being analyzed are those to the movement *from* off-spec batch 410 *to* anomalous on-spec batch 422 (Fig. A6 top); and *from* off-spec batch 415 *to* anomalous on-spec batch 420 (Fig. A6 bottom); and both plots suggest the same answer: fast heating during stage two and long cooling time. These two characteristics were “strong” enough to compensate for all the other adverse characteristics present in each of the anomalous batches, and made the process yield product within specification. And this is the same conclusion obtained in chapter 1 using separate PLS models.

## Appendix 2

### ***On-line* Hotelling's T<sup>2</sup> Statistic Calculation using the time-varying variance- covariance of the scores for Batch Process Monitoring**

In the early work by Nomikos and MacGregor (1995), the Hotelling's statistic for the analysis of batch process data is shown to be calculated as (Eq. A1):

$$D = \mathbf{t}_R^T \mathbf{S}^{-1} \mathbf{t}_R \mathbf{I} / (\mathbf{I} - 1)^2 \quad (\text{A1})$$

where  $\mathbf{t}_R$  is a vector containing the R retained components of the model, and S represents the covariance matrix of the R retained score vectors. The S covariance matrix is referred as a diagonal matrix due to the orthogonality of the scores, which is true for the final score estimate. However, when computing this statistic for the on-line monitoring of batches, two considerations are to be taken: a) the covariance of the scores changes with time, and b) the scores might become non-orthogonal (Fig. 3.15). In this work, the Hotelling's statistic is computed using the correct and complete variance-covariance matrix that corresponds to each time sample, as suggested by Nomikos and MacGregor (1995). This time-varying variance-covariance is computed using the reference set of batches. The Hotelling's statistic is then computed as (Eq. A2):

$$D_{ki} = \hat{\mathbf{t}}_{Rki}^T \mathbf{S}_k^{-1} \hat{\mathbf{t}}_{Rki} \mathbf{I} / (\mathbf{I} - 1)^2 \quad (\text{A2})$$

$$D_{ki} = \hat{\tau}_{Rki}^T \mathbf{S}_k^{-1} \hat{\tau}_{Rki} \mathbf{I} / (\mathbf{I} - 1)^2 \quad (\text{A2})$$

where the estimate of the Hotelling statistic at time  $k$  for batch  $i$  ( $D_{ki}$ ) is a function of the estimate of the score vector for the  $R$  retained components at time  $k$  ( $\hat{\tau}_{Rki}$ ) for batch  $i$ , and the covariance matrix of the scores at time  $k$  ( $\mathbf{S}_k$ ). This Hotelling's statistic is distributed as an  $F(R, I-R)$  distribution. Notice that  $D_{ki}$  will change depending on the selected method to solve the missing data problem (or the option selected to "fill in"), since it is a function of  $\hat{\tau}_{Rki}$ , which has been shown throughout this work, to differ from method to method, and option to option. Using this corrected version of the Hotelling's statistic dramatically improves abnormality detection.

## Appendix 3

### Numerical Derivative and Partial Derivative Calculation for Batch Trajectories

In this appendix, the derivatives of a set of batch trajectories with respect to its evolution index ( $ei$ ) are formulated as a linear function of the trajectories. Consider, to begin with, the case of a single trajectory  $\chi$  (column vector) which has  $k$  measurements ( $\chi \in \mathbb{R}^{k \times 1}$ ). It is assumed that each sample of this trajectory corresponds to a unique point in the evolution of the batch, which can be time, or conversion in the case of a chemical reactor (*e.g.* for time =  $k$  there is only one corresponding value of  $\chi$ , referred to as  $\chi_k$ ), and for simplicity, assume that the evolution index is equally sampled at  $h$  intervals (this is not a requirement, and the formulation can still be obtained if this assumption does not hold for specific case).

A simple solution in the computation of the derivative of  $\chi_k$  with respect to its  $ei$  (referred to as  $d\chi_k$ ) is to calculate a single point derivative (Eq. A3). This however may yield very noisy derivatives due to presence of noise in the process measurements. A better calculation of the numerical derivatives is required.

$$d\chi_k = \chi_{k-1} - \chi_k \tag{A3}$$

High accuracy numerical derivatives can be obtained if a Taylor series is analytically differentiated. The higher the order of the Taylor series, the more points the numerical derivative will require for its calculation. In this work, a three point derivative ahead (Eq. A4) is considered for the first two samples in  $\chi$ , three point behind (Eq. A5) for the last two samples of  $\chi$ , and a four points centered (Eq. A6) derivative for the rest of the samples(Chapra and Canale 1985).

$$d\chi_k = \frac{-\chi_{k+2} + 4\chi_{k+1} - 3\chi_k}{2} \tag{A4}$$

$$d\chi_k = \frac{\chi_{k-2} - 4\chi_{k-1} + 3\chi_k}{2} \tag{A5}$$

$$d\chi_k = \frac{-\chi_{k+2} + 8\chi_{k+1} - 8\chi_{k-1} + \chi_{k-2}}{12} \tag{A6}$$

The fact that the actual magnitude of  $h$  is not considered is so that the derivatives will have the same units as the evolution index. These formulas can be formulated as matrix operations. For example, if  $\chi$  consists of 5 samples,

$$d\chi^T = \chi^T \begin{bmatrix} -3 & 0 & 1 & 0 & 0 & 0 & 0 \\ 4 & -3 & -8 & 1 & 0 & 0 & 0 \\ -1 & 4 & 0 & -8 & 1 & 0 & 0 \\ 0 & -1 & 8 & 0 & -8 & 1 & 0 \\ 0 & 0 & -1 & 8 & 0 & -4 & 1 \\ 0 & 0 & 0 & -1 & 8 & 3 & -4 \\ 0 & 0 & 0 & 0 & -1 & 0 & 3 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \\ 12 \\ 12 \\ 0 \\ 12 \\ 2 \\ 2 \end{bmatrix} \tag{A7}$$

If all the two numerical matrices are referred to as **D**,

$$d\chi = \mathbf{D} \chi \tag{A8}$$

$\mathbf{D}$  is a square matrix which has as many elements as samples in  $\chi$ . In the case where the batch trajectory  $\chi$  is preceded by  $L$  elements corresponding to the initial conditions, the  $\mathbf{D}$  matrix can be modified by adding  $L$  columns of zeros. For the case where there are  $J$  variable trajectories, a bigger matrix  $\mathbf{D}_A$  can be built with  $\mathbf{D}$  and as many columns of zeros as  $L$  initial conditions in  $\mathbf{Z}$  (Eq. A9 illustrating the case for  $L=2$  and  $J=6$ ) such that, the complete derivative of the trajectory can be expressed as Eq. A11, by previously ordering the trajectories as Eq. A10, where  $\mathbf{X}\mathbf{u}^T$  corresponds to the transpose of the unfolded batch data matrix where the columns are grouped by variables (Fig. 4.14).

$$D_A = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{D} & & & & & & \\ & \mathbf{D} & & & & & \\ & & \mathbf{D} & & & & \\ & & & \mathbf{D} & & & \\ & & & & \mathbf{D} & & \\ & 0 & & & & \mathbf{D} & \\ & & & & & & \mathbf{D} \end{bmatrix} \quad (\text{A9})$$

$$\mathbf{X}\mathbf{u}^T = \begin{bmatrix} \chi_{1,1} & \chi_{1,2} & \chi_{1,3} & \cdots & \chi_{1,J} \\ \chi_{2,1} & \chi_{2,2} & \chi_{2,3} & \cdots & \chi_{2,J} \\ \chi_{3,1} & \chi_{3,2} & \chi_{3,3} & \cdots & \chi_{3,J} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \chi_{J,1} & \chi_{J,2} & \chi_{J,3} & \cdots & \chi_{J,J} \end{bmatrix} \text{ which is a } JK \times I \text{ matrix.} \quad (\text{A10})$$

$$d\mathbf{X}\mathbf{u}^T = \mathbf{D}_A \begin{bmatrix} \mathbf{Z}^T \\ \mathbf{X}\mathbf{u}^T \end{bmatrix} \quad (\text{A11})$$

$$[JK \times I] = [JK \times (JK + L)] \times [(JK + L) \times I]$$

The numerical partial derivative formula used was obtained with the analytical derivative of a third order Lagrange polynomial interpolation (which does not require equally spaced data) which uses four points  $(x_0, x_1, x_2, x_3)$ :

$$\begin{aligned}
 \left. \frac{\delta f(x)}{\delta x} \right|_x &= f(x_0) \frac{3x^2 - 2x(x_1 + x_2 + x_3) + x_1x_2 + x_1x_3 + x_2x_3}{(x_0 - x_1)(x_0 - x_2)(x_0 - x_3)} \\
 &+ f(x_1) \frac{3x^2 - 2x(x_0 + x_2 + x_3) + x_0x_2 + x_0x_3 + x_2x_3}{(x_1 - x_0)(x_1 - x_2)(x_1 - x_3)} \\
 &+ f(x_2) \frac{3x^2 - 2x(x_0 + x_1 + x_3) + x_0x_1 + x_0x_3 + x_1x_3}{(x_2 - x_0)(x_2 - x_1)(x_2 - x_3)} \\
 &+ f(x_3) \frac{3x^2 - 2x(x_0 + x_1 + x_2) + x_0x_1 + x_0x_2 + x_1x_2}{(x_3 - x_0)(x_3 - x_1)(x_3 - x_2)}
 \end{aligned} \tag{A12}$$

## Appendix 4

### Analytical Solution to the Optimization Problem in JYPLS

The objective function for the first component of the JYPLS model can be written as :

$$\begin{aligned} \max_{\mathbf{w}\mathbf{a}_j, \mathbf{w}\mathbf{b}_j} & \left\{ \begin{bmatrix} \mathbf{t}\mathbf{a}_j \\ \mathbf{t}\mathbf{b}_j \end{bmatrix}^T \begin{bmatrix} \mathbf{Y}_A \\ \mathbf{Y}_B \end{bmatrix} \begin{bmatrix} \mathbf{Y}_A \\ \mathbf{Y}_B \end{bmatrix}^T \begin{bmatrix} \mathbf{t}\mathbf{a}_j \\ \mathbf{t}\mathbf{b}_j \end{bmatrix} \right\} \\ \text{s.t.} & \quad \mathbf{t}\mathbf{a}_j = \mathbf{X}_A \mathbf{w}\mathbf{a}_j \\ & \quad \mathbf{t}\mathbf{b}_j = \mathbf{X}_B \mathbf{w}\mathbf{b}_j \\ & \quad \mathbf{w}\mathbf{a}_j^T \mathbf{w}\mathbf{a}_j = \alpha \\ & \quad \mathbf{w}\mathbf{b}_j^T \mathbf{w}\mathbf{b}_j = \beta \end{aligned} \tag{A13}$$

To obtain the eigenvector-eigenvalue solution, a change of variable is needed:

$$\mathbf{w} = \begin{bmatrix} \mathbf{w}\mathbf{a}_j \\ \mathbf{w}\mathbf{b}_j \end{bmatrix} \tag{A14}$$

Substituting into Eq. A13, we obtain Eq. A15.

$$\begin{aligned} \max_{\mathbf{w}} & \left\{ \mathbf{w}^T \begin{bmatrix} \mathbf{X}_A^T & 0 \\ 0 & \mathbf{X}_B^T \end{bmatrix} \begin{bmatrix} \mathbf{Y}_A \\ \mathbf{Y}_B \end{bmatrix} \begin{bmatrix} \mathbf{Y}_A \\ \mathbf{Y}_B \end{bmatrix}^T \begin{bmatrix} \mathbf{X}_A & 0 \\ 0 & \mathbf{X}_B \end{bmatrix} \mathbf{w} \right\} \\ \text{s.t.} & \quad \mathbf{w}^T \mathbf{w} = 1 \end{aligned} \tag{A15}$$

Notice that now the  $\mathbf{w}$  vector is being normalized to one, this constraint will have the same function as the normalization of the individual loading vectors, which is to keep the solution from being infinity. After algebraic manipulation and including the constraint into the objective function with the use of a Lagrangian multiplier, Eq. A15 becomes Eq. A16,

$$\max_{\mathbf{w}} \left\{ \mathbf{w}^T \begin{bmatrix} \mathbf{X}_A^T \mathbf{Y}_A \mathbf{Y}_A^T \mathbf{X}_A & \mathbf{X}_A^T \mathbf{Y}_A \mathbf{Y}_B^T \mathbf{X}_B \\ \mathbf{X}_B^T \mathbf{Y}_B \mathbf{Y}_A^T \mathbf{X}_A & \mathbf{X}_B^T \mathbf{Y}_B \mathbf{Y}_B^T \mathbf{X}_B \end{bmatrix} \mathbf{w} - \lambda(\mathbf{w}^T \mathbf{w} - 1) \right\} \quad (\text{A16})$$

$$\Phi = \mathbf{w}^T \begin{bmatrix} \mathbf{X}_A^T \mathbf{Y}_A \mathbf{Y}_A^T \mathbf{X}_A & \mathbf{X}_A^T \mathbf{Y}_A \mathbf{Y}_B^T \mathbf{X}_B \\ \mathbf{X}_B^T \mathbf{Y}_B \mathbf{Y}_A^T \mathbf{X}_A & \mathbf{X}_B^T \mathbf{Y}_B \mathbf{Y}_B^T \mathbf{X}_B \end{bmatrix} \mathbf{w} - \lambda(\mathbf{w}^T \mathbf{w} - 1)$$

To obtain an analytical solution, we now take the derivative of the objective function ( $\Phi$ ) with respect to  $\mathbf{w}$  gives equation A5, and solving this to zero gives the eigenvector-eigenvalue solution to the JYPLS model (Eqs. A17 to A20). The  $\mathbf{W}$  vector is in fact the eigenvector corresponding to the highest eigenvalue of  $\Sigma$ .

$$\frac{\delta \Phi}{\delta \mathbf{w}} = 2 \begin{bmatrix} \mathbf{X}_A^T \mathbf{Y}_A \mathbf{Y}_A^T \mathbf{X}_A & \mathbf{X}_A^T \mathbf{Y}_A \mathbf{Y}_B^T \mathbf{X}_B \\ \mathbf{X}_B^T \mathbf{Y}_B \mathbf{Y}_A^T \mathbf{X}_A & \mathbf{X}_B^T \mathbf{Y}_B \mathbf{Y}_B^T \mathbf{X}_B \end{bmatrix} \mathbf{w} - 2\lambda \mathbf{w} \quad (\text{A17})$$

$$\begin{bmatrix} \mathbf{X}_A^T \mathbf{Y}_A \mathbf{Y}_A^T \mathbf{X}_A & \mathbf{X}_A^T \mathbf{Y}_A \mathbf{Y}_B^T \mathbf{X}_B \\ \mathbf{X}_B^T \mathbf{Y}_B \mathbf{Y}_A^T \mathbf{X}_A & \mathbf{X}_B^T \mathbf{Y}_B \mathbf{Y}_B^T \mathbf{X}_B \end{bmatrix} \mathbf{w} = \lambda \mathbf{w} \quad (\text{A18})$$

$$\Sigma = \begin{bmatrix} \mathbf{X}_A^T \mathbf{Y}_A \mathbf{Y}_A^T \mathbf{X}_A & \mathbf{X}_A^T \mathbf{Y}_A \mathbf{Y}_B^T \mathbf{X}_B \\ \mathbf{X}_B^T \mathbf{Y}_B \mathbf{Y}_A^T \mathbf{X}_A & \mathbf{X}_B^T \mathbf{Y}_B \mathbf{Y}_B^T \mathbf{X}_B \end{bmatrix} \quad (\text{A19})$$

$$\Sigma \mathbf{w} = \lambda \mathbf{w} \quad (\text{A20})$$

## Appendix 5

### Proof of the NIPALS-Modified Algorithm to Give the Analytical Eigenvector Solution to the Optimization Problem in JYPLS

At convergence, in the JYPLS model  $w=Xu(u'u)^{-1}$  in each site

$$\begin{bmatrix} \mathbf{w}\mathbf{a} \\ \mathbf{w}\mathbf{b} \end{bmatrix} = \begin{bmatrix} \mathbf{X}\mathbf{a}^T\mathbf{u}\mathbf{a}(\mathbf{u}\mathbf{a}^T\mathbf{u}\mathbf{a})^{-1} \\ \mathbf{X}\mathbf{b}^T\mathbf{u}\mathbf{b}(\mathbf{u}\mathbf{b}^T\mathbf{u}\mathbf{b})^{-1} \end{bmatrix} = \begin{bmatrix} \mathbf{X}\mathbf{a}^T\mathbf{u}\mathbf{a} & 0 \\ 0 & \mathbf{X}\mathbf{b}^T\mathbf{u}\mathbf{b} \end{bmatrix} \begin{bmatrix} (\mathbf{u}\mathbf{a}^T\mathbf{u}\mathbf{a})^{-1} \\ (\mathbf{u}\mathbf{b}^T\mathbf{u}\mathbf{b})^{-1} \end{bmatrix} \quad (\text{A21})$$

at convergence  $\mathbf{u}=\mathbf{Y}\mathbf{q}(\mathbf{q}'\mathbf{q})^{-1}$  for the joint space of Y

$$\begin{aligned} \begin{bmatrix} \mathbf{w}\mathbf{a} \\ \mathbf{w}\mathbf{b} \end{bmatrix} &= \begin{bmatrix} \mathbf{X}\mathbf{a}^T\mathbf{Y}\mathbf{a}\mathbf{q}(\mathbf{q}'\mathbf{q})^{-1} & 0 \\ 0 & \mathbf{X}\mathbf{b}^T\mathbf{Y}\mathbf{b}\mathbf{q}(\mathbf{q}'\mathbf{q})^{-1} \end{bmatrix} \begin{bmatrix} (\mathbf{u}\mathbf{a}^T\mathbf{u}\mathbf{a})^{-1} \\ (\mathbf{u}\mathbf{b}^T\mathbf{u}\mathbf{b})^{-1} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{X}\mathbf{a}^T\mathbf{Y}\mathbf{a}\mathbf{q} & 0 \\ 0 & \mathbf{X}\mathbf{b}^T\mathbf{Y}\mathbf{b}\mathbf{q} \end{bmatrix} \begin{bmatrix} (\mathbf{q}'\mathbf{q})^{-1} & 0 \\ 0 & (\mathbf{q}'\mathbf{q})^{-1} \end{bmatrix} \begin{bmatrix} (\mathbf{u}\mathbf{a}^T\mathbf{u}\mathbf{a})^{-1} \\ (\mathbf{u}\mathbf{b}^T\mathbf{u}\mathbf{b})^{-1} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{X}\mathbf{a}^T\mathbf{Y}\mathbf{a}\mathbf{q} & 0 \\ 0 & \mathbf{X}\mathbf{b}^T\mathbf{Y}\mathbf{b}\mathbf{q} \end{bmatrix} \mathbf{q}'\mathbf{u}' \end{aligned} \quad (\text{A22})$$

at convergence  $q=Yt(t't)^{-1}$  for the joint Y space,

$$\begin{bmatrix} \mathbf{wa} \\ \mathbf{wb} \end{bmatrix} = \begin{bmatrix} \mathbf{Xa}^T \mathbf{Ya} [\mathbf{Ya}^T \mathbf{Yb}^T] \begin{bmatrix} \mathbf{ta} \\ \mathbf{tb} \end{bmatrix} \left( \begin{bmatrix} \mathbf{ta}^T & \mathbf{tb}^T \end{bmatrix} \begin{bmatrix} \mathbf{ta} \\ \mathbf{tb} \end{bmatrix} \right)^{-1} & 0 \\ 0 & \mathbf{Xb}^T \mathbf{Yb} [\mathbf{Ya}^T \mathbf{Yb}^T] \begin{bmatrix} \mathbf{ta} \\ \mathbf{tb} \end{bmatrix} \left( \begin{bmatrix} \mathbf{ta}^T & \mathbf{tb}^T \end{bmatrix} \begin{bmatrix} \mathbf{ta} \\ \mathbf{tb} \end{bmatrix} \right)^{-1} \end{bmatrix} \mathbf{q}' \mathbf{u}'$$

$$= \begin{bmatrix} \mathbf{Xa}^T \mathbf{Ya} [\mathbf{Ya}^T \mathbf{Yb}^T] \begin{bmatrix} \mathbf{ta} \\ \mathbf{tb} \end{bmatrix} & 0 \\ 0 & \mathbf{Xb}^T \mathbf{Yb} [\mathbf{Ya}^T \mathbf{Yb}^T] \begin{bmatrix} \mathbf{ta} \\ \mathbf{tb} \end{bmatrix} \end{bmatrix} \mathbf{tj}' \mathbf{q}' \mathbf{u}'$$

(A23)

and working the internal matrix operations,

$$\begin{bmatrix} \mathbf{wa} \\ \mathbf{wb} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \mathbf{Xa}^T \mathbf{Ya} \mathbf{Ya}^T & \mathbf{Xa}^T \mathbf{Ya} \mathbf{Yb}^T \end{bmatrix} \begin{bmatrix} \mathbf{ta} \\ \mathbf{tb} \end{bmatrix} & 0 \\ 0 & \begin{bmatrix} \mathbf{Xb}^T \mathbf{Yb} \mathbf{Ya}^T & \mathbf{Xb}^T \mathbf{Yb} \mathbf{Yb}^T \end{bmatrix} \begin{bmatrix} \mathbf{ta} \\ \mathbf{tb} \end{bmatrix} \end{bmatrix} \mathbf{tj}' \mathbf{q}' \mathbf{u}'$$

(A24)

at convergence,  $t=\mathbf{Xw}(w'w)^{-1}$  for each considered site

$$\begin{bmatrix} \mathbf{ta} \\ \mathbf{tb} \end{bmatrix} = \begin{bmatrix} \mathbf{Xa} \mathbf{wa} (\mathbf{wa}^T \mathbf{wa})^{-1} \\ \mathbf{Xb} \mathbf{wb} (\mathbf{wb}^T \mathbf{wb})^{-1} \end{bmatrix} \quad (A25)$$

Substituting Eq. A25 in A24,

$$\begin{aligned}
 \begin{bmatrix} \mathbf{w}_a \\ \mathbf{w}_b \end{bmatrix} &= \\
 \begin{bmatrix} \mathbf{X}_a^T \mathbf{Y}_a \mathbf{Y}_a^T \mathbf{X}_a \mathbf{w}_a & \mathbf{X}_a^T \mathbf{Y}_a \mathbf{Y}_b^T \mathbf{X}_b \mathbf{w}_b & 0 & 0 \\ 0 & 0 & \mathbf{X}_b^T \mathbf{Y}_b \mathbf{Y}_a^T \mathbf{X}_a \mathbf{w}_a & \mathbf{X}_b^T \mathbf{Y}_b \mathbf{Y}_b^T \mathbf{X}_b \mathbf{w}_b \end{bmatrix} & \begin{bmatrix} (\mathbf{w}_a^T \mathbf{w}_a)^{-1} \\ (\mathbf{w}_b^T \mathbf{w}_b)^{-1} \\ (\mathbf{w}_a^T \mathbf{w}_a)^{-1} \\ (\mathbf{w}_b^T \mathbf{w}_b)^{-1} \end{bmatrix} \mathbf{t}_j' \mathbf{q}' \mathbf{u}' \\
 &= \begin{bmatrix} \mathbf{X}_a^T \mathbf{Y}_a \mathbf{Y}_a^T \mathbf{X}_a \mathbf{w}_a & \mathbf{X}_a^T \mathbf{Y}_a \mathbf{Y}_b^T \mathbf{X}_b \mathbf{w}_b & 0 & 0 \\ 0 & 0 & \mathbf{X}_b^T \mathbf{Y}_b \mathbf{Y}_a^T \mathbf{X}_a \mathbf{w}_a & \mathbf{X}_b^T \mathbf{Y}_b \mathbf{Y}_b^T \mathbf{X}_b \mathbf{w}_b \end{bmatrix} \mathbf{w}' \mathbf{t}_j' \mathbf{q}' \mathbf{u}'
 \end{aligned} \tag{A26}$$

Expanding the right hand terms in Eq. A26,

$$\mathbf{w}' \mathbf{t}_j' \mathbf{q}' \mathbf{u}' = \begin{bmatrix} (\mathbf{w}_a^T \mathbf{w}_a)^{-1} (\mathbf{t}_j^T \mathbf{t}_j)^{-1} (\mathbf{q}^T \mathbf{q})^{-1} (\mathbf{u}_a^T \mathbf{u}_a)^{-1} \\ (\mathbf{w}_b^T \mathbf{w}_b)^{-1} (\mathbf{t}_j^T \mathbf{t}_j)^{-1} (\mathbf{q}^T \mathbf{q})^{-1} (\mathbf{u}_a^T \mathbf{u}_a)^{-1} \\ (\mathbf{w}_a^T \mathbf{w}_a)^{-1} (\mathbf{t}_j^T \mathbf{t}_j)^{-1} (\mathbf{q}^T \mathbf{q})^{-1} (\mathbf{u}_b^T \mathbf{u}_b)^{-1} \\ (\mathbf{w}_b^T \mathbf{w}_b)^{-1} (\mathbf{t}_j^T \mathbf{t}_j)^{-1} (\mathbf{q}^T \mathbf{q})^{-1} (\mathbf{u}_b^T \mathbf{u}_b)^{-1} \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix}, \text{ where } \mathbf{t}_j = \begin{bmatrix} \mathbf{t}_a \\ \mathbf{t}_b \end{bmatrix} \tag{A27}$$

Substituting these scalars from Eq. A27 in Eq. A26,

$$\begin{bmatrix} \mathbf{w}_a \\ \mathbf{w}_b \end{bmatrix} = \begin{bmatrix} \mathbf{X}_a^T \mathbf{Y}_a \mathbf{Y}_a^T \mathbf{X}_a \mathbf{w}_a \alpha_1 + \mathbf{X}_a^T \mathbf{Y}_a \mathbf{Y}_b^T \mathbf{X}_b \mathbf{w}_b \alpha_2 \\ \mathbf{X}_b^T \mathbf{Y}_b \mathbf{Y}_a^T \mathbf{X}_a \mathbf{w}_a \alpha_3 + \mathbf{X}_b^T \mathbf{Y}_b \mathbf{Y}_b^T \mathbf{X}_b \mathbf{w}_b \alpha_4 \end{bmatrix} \tag{A28}$$

If  $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = \alpha^{-1}$  in Eq. A28, then this expression can be reduced to an eigenvalue- eigenvector form:

$$\begin{bmatrix} \mathbf{w}_a \\ \mathbf{w}_b \end{bmatrix} \alpha = \begin{bmatrix} \mathbf{X}_a^T \mathbf{Y}_a \mathbf{Y}_a^T \mathbf{X}_a & \mathbf{X}_a^T \mathbf{Y}_a \mathbf{Y}_b^T \mathbf{X}_b \\ \mathbf{X}_b^T \mathbf{Y}_b \mathbf{Y}_a^T \mathbf{X}_a & \mathbf{X}_b^T \mathbf{Y}_b \mathbf{Y}_b^T \mathbf{X}_b \end{bmatrix} \begin{bmatrix} \mathbf{w}_a \\ \mathbf{w}_b \end{bmatrix} \tag{A29}$$

$$\text{if } \begin{bmatrix} \mathbf{w}_a \\ \mathbf{w}_b \end{bmatrix} = \mathbf{w} \text{ then } \begin{bmatrix} \mathbf{X}_a^T \mathbf{Y}_a \mathbf{Y}_a^T \mathbf{X}_a & \mathbf{X}_a^T \mathbf{Y}_a \mathbf{Y}_b^T \mathbf{X}_b \\ \mathbf{X}_b^T \mathbf{Y}_b \mathbf{Y}_a^T \mathbf{X}_a & \mathbf{X}_b^T \mathbf{Y}_b \mathbf{Y}_b^T \mathbf{X}_b \end{bmatrix} \mathbf{w} = \alpha \mathbf{w} \tag{A30}$$

## Appendix 6

### Nonlinear JYPLS Algorithm

This algorithm can be modified to include any non-linear function between the T's and the U's. The present algorithm is specifically for the case of  $\mathbf{u}$ 's being a quadratic function of the  $\mathbf{t}$ 's (Eqs. A30 and A31), each of the scores for each plant holds its own nonlinearity having each set its corresponding vector  $\mathbf{b}$ .

$$\mathbf{ua}^a = ba_0^a + ba_1^a \mathbf{ta}^a + ba_2^a (\mathbf{ta}^a)^2 \quad (\text{A30})$$

$$\mathbf{ub}^a = bb_0^a + bb_1^a \mathbf{tb}^a + bb_2^a (\mathbf{tb}^a)^2 \quad (\text{A31})$$

The algorithm is similar to the one for the nonlinear quadratic PLS proposed by Wold (1989) as follows (subindex  $a$  denotes component) :

1. Define  $\mathbf{Y}_j = [\mathbf{Y}_a^T \ \mathbf{Y}_b^T]^T$ , define  $n$  as rows in  $\mathbf{Y}_a$  and  $m$  as rows in  $\mathbf{Y}_b$ .
2. Mean center and scale  $\mathbf{X}_a$ ,  $\mathbf{X}_b$ ,  $\mathbf{Y}_a$  and  $\mathbf{Y}_b$
3.  $a=1$
4. Initialize  $\mathbf{ua}^a$ ,  $\mathbf{ub}^a$ ,  $\mathbf{wa}^a$ ,  $\mathbf{wb}^a$ ,  $\mathbf{ta}^a$ ,  $\mathbf{tb}^a$ , with the linear JYPLS algorithm
5. Fit  $\mathbf{ba}_i^a$  and  $\mathbf{bb}_i^a$  with the initial  $\mathbf{ua}^a$ ,  $\mathbf{ub}^a$ ,  $\mathbf{ta}^a$ , and  $\mathbf{tb}^a$  using ordinary least squares and Eqs. A30 and A31.
6. Calculate  $\hat{\mathbf{u}}\mathbf{a}^a$  and  $\hat{\mathbf{u}}\mathbf{b}^a$  using  $\mathbf{ba}_i^a$ ,  $\mathbf{bb}_i^a$ ,  $\mathbf{ta}^a$ , and  $\mathbf{tb}^a$  and Eqs. A30 and A31
7. Regress the columns of the Joint Y onto the joint  $\hat{\mathbf{u}}\mathbf{j}^a = \begin{bmatrix} \hat{\mathbf{u}}\mathbf{a}^a \\ \hat{\mathbf{u}}\mathbf{b}^a \end{bmatrix}$ , to obtain  $\mathbf{q}^a$  :

$$\mathbf{q}_a^T = \hat{\mathbf{u}}\mathbf{j}_a^T \mathbf{Y}\mathbf{j} / (\hat{\mathbf{u}}\mathbf{j}_a^T \hat{\mathbf{u}}\mathbf{j}_a)$$

8. Compute  $\mathbf{u}\mathbf{j}_a = \mathbf{Y}\mathbf{j} \mathbf{q}_a / (\mathbf{q}_a^T \mathbf{q}_a)$

9. Update  $\mathbf{w}\mathbf{a}^a$  and  $\mathbf{w}\mathbf{b}^a$  by nonlinear regression such that

$$\mathbf{u}\mathbf{a}^a = b\mathbf{a}_0^a + b\mathbf{a}_1^a(\mathbf{x}\mathbf{a} \mathbf{w}\mathbf{a}^a) + b\mathbf{a}_2^a(\mathbf{x}\mathbf{a} \mathbf{w}\mathbf{a}^a)^2$$

$$\mathbf{u}\mathbf{b}^a = b\mathbf{a}_0^a + b\mathbf{a}_1^a(\mathbf{x}\mathbf{a} \mathbf{w}\mathbf{a}^a) + b\mathbf{a}_2^a(\mathbf{x}\mathbf{a} \mathbf{w}\mathbf{a}^a)^2$$

10. normalize  $\mathbf{w}\mathbf{a}^a$  and  $\mathbf{w}\mathbf{b}^a$  with Eq. 5.33

11. Compute new t scores:  $\mathbf{t}\mathbf{a}^a = \mathbf{X}\mathbf{a} \mathbf{w}\mathbf{a}^a$  and  $\mathbf{t}\mathbf{b}^a = \mathbf{X}\mathbf{b} \mathbf{w}\mathbf{b}^a$

12. Calculate  $\hat{\mathbf{u}}\mathbf{a}\mathbf{n}^a$  and  $\hat{\mathbf{u}}\mathbf{b}\mathbf{n}^a$

$$\hat{\mathbf{u}}\mathbf{a}\mathbf{n}^a = b\mathbf{a}_0^a + b\mathbf{a}_1^a \mathbf{t}\mathbf{a}^a + b\mathbf{a}_2^a (\mathbf{t}\mathbf{a}^a)^2 \quad \hat{\mathbf{u}}\mathbf{b}\mathbf{n}^a = b\mathbf{b}_0^a + b\mathbf{b}_1^a \mathbf{t}\mathbf{b}^a + b\mathbf{b}_2^a (\mathbf{t}\mathbf{b}^a)^2$$

13. Regress the columns of the Joint Y onto the joint  $\hat{\mathbf{u}}\mathbf{n}\mathbf{j}_a = \begin{bmatrix} \hat{\mathbf{u}}\mathbf{a}\mathbf{n}^a \\ \hat{\mathbf{u}}\mathbf{b}\mathbf{n}^a \end{bmatrix}$ , to obtain  $\mathbf{q}_a$  :

$$\mathbf{q}_a^T = \hat{\mathbf{u}}\mathbf{n}\mathbf{j}_a^T \mathbf{Y}\mathbf{j} / (\hat{\mathbf{u}}\mathbf{n}\mathbf{j}_a^T \hat{\mathbf{u}}\mathbf{n}\mathbf{j}_a)$$

14. Compute  $\mathbf{u}\mathbf{n}\mathbf{j}_a = \mathbf{Y}\mathbf{j} \mathbf{q}_a / (\mathbf{q}_a^T \mathbf{q}_a)$

15. Check convergence in  $\mathbf{t}\mathbf{a}^a$ ,  $\mathbf{t}\mathbf{b}^a$  and  $\mathbf{t}\mathbf{a}\mathbf{n}^a$ ,  $\mathbf{t}\mathbf{b}\mathbf{n}^a$

16.  $\mathbf{t}\mathbf{a}^a = \mathbf{t}\mathbf{a}\mathbf{n}^a$ ,  $\mathbf{t}\mathbf{b}^a = \mathbf{t}\mathbf{b}\mathbf{n}^a$ ,  $\mathbf{u}\mathbf{a}^a = \mathbf{u}\mathbf{n}\mathbf{j}_a|_{(1:n)}$ ,  $\mathbf{u}\mathbf{b}^a = \mathbf{u}\mathbf{n}\mathbf{j}_a|_{(n+1:m)}$

17. If not converged return to step 5

18. If converged, calculate  $\mathbf{p}\mathbf{a}^a$  and  $\mathbf{p}\mathbf{b}^a$  as  $\mathbf{p}\mathbf{a}^a = \mathbf{X}\mathbf{a} \mathbf{t}\mathbf{a}^a / (\mathbf{t}\mathbf{a}^{aT} \mathbf{t}\mathbf{a}^a)$  and

$$\mathbf{p}\mathbf{b}^a = \mathbf{X}\mathbf{b} \mathbf{t}\mathbf{b}^a / (\mathbf{t}\mathbf{b}^{aT} \mathbf{t}\mathbf{b}^a)$$

19. Deflate  $\mathbf{Y}\mathbf{j} = \mathbf{Y}\mathbf{j} - \begin{bmatrix} \mathbf{u}\mathbf{a}^a \\ \mathbf{u}\mathbf{b}^a \end{bmatrix} \mathbf{q}_a$ ,  $\mathbf{X}\mathbf{a} = \mathbf{X}\mathbf{a} - \mathbf{t}\mathbf{a}^a \mathbf{p}\mathbf{a}^{aT}$ ,  $\mathbf{X}\mathbf{b} = \mathbf{X}\mathbf{b} - \mathbf{t}\mathbf{b}^a \mathbf{p}\mathbf{b}^{aT}$

20.  $a=a+1$ , got to step 4 to compute next component.

## Appendix 7

### Building the test set for chapter 5

The data set used in chapter 5 was constructed starting with a latent space. A latent space consists of an  $n \times A$  matrix ( $\mathbf{T}$ ) where each column is independent and the variance of the first column is greater than the variance of the second, and so on. The construction of such matrix is illustrated with the MATLAB<sup>®</sup> code used. The code lines are numerated for clarity in table A1.

Line Num	Code
1	clear
2	T1=randn(100,1)*5;
3	T2=randn(100,1)*3;
4	T3=randn(100,1)*1.5;
5	[T1 T2 T3]'*[T1 T2 T3]
6	T2=[T2-T1*inv(T1'*T1)*T1'*T2];
7	[T1 T2 T3]'*[T1 T2 T3]
8	T3=[T3-[T1 T2]*inv([T1 T2]'*[T1 T2])*[T1 T2]'*T3];
9	[T1 T2 T3]'*[T1 T2 T3]
10	T=[T1 T2 T3];

**Table A1. MATLAB<sup>®</sup> code to build a three dimensional latent space ( $\mathbf{T}$ )**

The goal is to generate three column vectors with decreasing variance (as column number increases), and to introduce orthogonality among the columns.

**Lines 1 to 4** will generate three column vectors with independent and normally distributed elements, the scalars 5, 3 and 1.5 are introduced to modify the variance of each one of the vectors. Once the vectors are generated, these will not be orthogonal; this is evident when executing **line 5**, which will give an estimate of the correlation among columns (a sample output is shown in table A2), notice that the off-diagonal terms are different from zero which indicates a certain degree of correlation among the column vectors.

2813.7	-92.97	-27.77
-92.97	854.46	-87.97
-27.77	-87.97	281.43

**Table A2. Correlation among T1 T2 and T3 showing some non-orthogonality**

To introduce orthogonality between **T2** and **T1**, a regression coefficient is estimated to fit the equation  $\mathbf{T2} = \mathbf{T1}\beta$ , and now **T2** is substituted with the residuals from this regression function ( $\mathbf{T2} = \mathbf{T2} - \mathbf{T1}\beta$ ), this is achieved with **line 6** of the provided code (where OLS is used to estimate  $\beta$ ). Once this is done, **T1** and **T2** will be orthogonal and this can be seen by executing **line 7** (same as line 5), a sample output is shown in table A3. Notice that the off-diagonal terms corresponding to the T1-T2 relationship are now zero.

2155.8	0	25.23
0	511.49	-17.35
25.23	-17.35	112.55

**Table A3. Correlation among T1 T2 and T3 after orthogonality is introduced between T1 and T2**

The same exercise is done in **line 8** between T3 and the matrix formed with T1 and T2. T3 is substituted with the residuals from an OLS fit between T3 and [T1 T2].

Once this line is executed, the output from **line 9** will show all the off-diagonal terms equal to zero, a sample output of this command is shown in table A4.

2155.8	0	0
0	511.49	0
0	0	111.66

**Table A4. Correlation among T1 T2 and T3 after orthogonality is introduced between T1 and T2 and T3**

All the column vectors are orthogonal to each other, the latent space **T** is constructed with these column vectors (**line 10**). These lines will result in a **T** matrix with variances of approximately 25, 9 and 2 for T1, T2 and T3 respectively.

Once the latent space **T** is constructed, the idea is to generate one **Y** matrix where each observation is a linear transformation of **T** as:

$$\begin{array}{ccccc} \mathbf{Y} & = & \mathbf{T} & \times & \mathbf{Q} & \text{(A32)} \\ (m \times n) & & (m \times A) & & (A \times n) & \end{array}$$

The **Q** matrix will define how much is each column of **Y** correlated with the columns of **T**. The **Q** matrix used in this work is given in table A5. The resultant **Y** space will become the **Y<sub>J</sub>** matrix. Two **X** spaces will be generated in the same way as the **Y** space was generated (but with different number of columns each). These **X** spaces will be highly correlated with the **Y<sub>J</sub>** space.

2.43	5.09	0.5	2.53	2.02E-005	0
0.44	1.52	4.55	10.33	3.50E-006	0
0	0	0	0	0	5.03E-005

**Table A5. Q matrix used to generate the Y space**



In this work, the original structures consist of 200 rows, and  $m_a$  and  $m_b$  were taken as 100 each. The *unused* part of each of the structures (last  $m_b$  rows of the original  $\mathbf{Xa}$ , and first  $m_a$  of the original  $\mathbf{Xb}$ ) are the known solutions for the transfer problem.

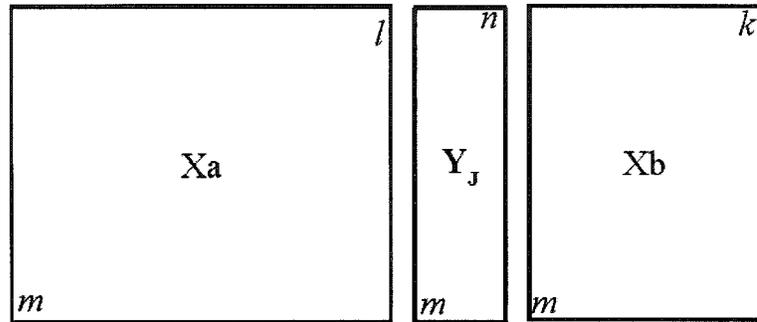


Figure A7. Data structure generated

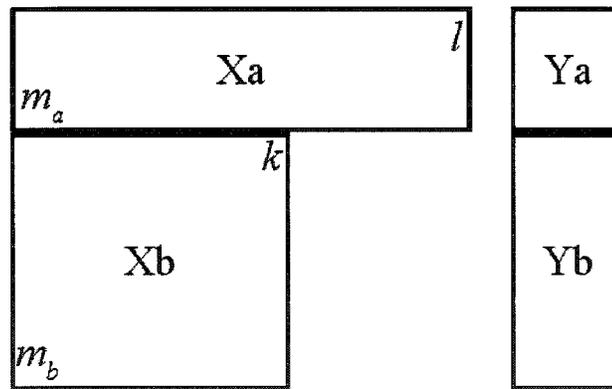


Figure A8. Partition of original data structures to generate  $\mathbf{Xa}$ ,  $\mathbf{Xb}$ ,  $\mathbf{Ya}$  and  $\mathbf{Yb}$

Finally, noise is introduced in  $\mathbf{Xb}$  and  $\mathbf{Yb}$ . The noise is introduced per column is independent white noise with standard deviation equal to the 8% and 2% (for  $\mathbf{Xb}$  and  $\mathbf{Yb}$  respectively) of the standard deviation of the corresponding column.

## References

- Akaike, H., "A New Look at the Statistical Model Identification", IEEE Transactions on Automatic Control, **AC-19**, 716-723. 1974.
- Arteaga, F. and A. Ferrer, "Dealing With Missing Data in MSPC: Several Methods, Different Interpretations, Some Examples", Journal of Chemometrics, **16**, 408-418. 2002.
- Biegler, L. T., A. M. Cervantes, and A. Wachter, "Advances in Simultaneous Strategies for Dynamic Process Optimization", Chemical Engineering Science, **57**, (4), 575-593. 2002.
- Boque, R. and A. K. Smilde, "Monitoring and Diagnosing Batch Processes With Multiway Covariates Regression Models", AIChE Journal, **45**, (7), 1504-1520. 1999.
- Burnham, A., J. F. MacGregor, and R. Viveros, "Latent Variable Multivariate Regression Modeling", Chemometrics and Intelligent Laboratory Systems, **48**, 167-180. 1999.
- Burnham, A., R. Viveros, and J. F. MacGregor, "Frameworks for Latent Variable Multivariate Regression", Journal of Chemometrics, **10**, 31-45. 1996.
- Chapra, S. C. and R. P. Canale. *Numerical Methods for Engineers : With Personal Computer Applications*, New York: McGraw Hill ,1985.
- Chen, F. Z. and X. Z. Wang, "Discovery of Operational Spaces From Process Data for Production of Multiple Grades of Products", Industrial & Engineering Chemistry Research, **39**, 2378-2383. 2000.
- Chin, I. S., K. S. Lee, and J. H. Lee, "A Technique for Integrated Quality Control, Profile Control and Constraint Handling for Batch Processes", Industrial & Engineering Chemistry Research, **39**, 693-705. 2000.
- Cho, H. and K Kim, "A Method for Predicting Future Observations in the Monitoring of a Batch Process", Journal of Quality Technology, **35**, (1), 59-69. 2003.
- Clarke-Pringle, T. and J. F. MacGregor, "Nonlinear Adaptive Temperature Control of Multi-Product, Semi-Batch Polymerization Reactors", Computers and Chemical Engineering, **21**, (12), 1395-1409. 1997.

- Clarke-Pringle, T. and J. F. MacGregor, "Optimization of Molecular-Weight Distribution Using Batch-to-Batch Adjustments", *Industrial & Engineering Chemistry Research*, **37**, 3660-3669. 1998.
- Duchesne, C. and J. F. MacGregor, "Establishing Multivariate Specification Regions for Incoming Materials", *Journal of Quality Technology*, **36**, (1), 78-94. 2004.
- Eriksson, L., E. Johansson, N. Kettaneh, and S. Wold. *Multi and Megavariate Data Analysis: Principles and Applications*, Umea: UMETRICS AB ,1999.
- Filippi, C., J. L. Greffe, J. Bordet, and J. Villermaux, "Tendency Modeling of Semibatch Reactors for Optimization and Control", *Chemical Engineering Science*, **41**, (4), 913-920. 1986.
- Flores-Cerrillo, J. and J. F. MacGregor, "Control of Particle Size Distributions in Emulsion Semibatch Polymerization Using Mid-Course Correction Policies.", *Industrial & Engineering Chemistry Research*, **41**, 1805-1814. 2002.
- Flores-Cerrillo, J. and J. F. MacGregor, "Within-Batch and Batch-to-Batch Inferential-Adaptive Control of Semi-Batch Reactors: A Partial Least Squares Approach.", *Industrial & Engineering Chemistry Research*, **42**, (14), 3334-3345. 2003.
- Flores-Cerrillo, J. and J. F. MacGregor, "Control of Batch Product Quality by Trajectory Manipulation Using Latent Variable Models", *Journal of Process Control*, **14**, 539-553. 2004.
- Floudas, C. A. and P. M. Pardalos. *Encyclopedia of Optimization*, vol. 4, Dordrecht-London-Boston: Kluwer Academic Publishers ,2001.
- Fotopoulos, J., C. Georgakis, and H. G. Stenger, "Use of Tendency Models and Their Uncertainty in the Design of State Estimators for Batch Reactors", *Chemical Engineering and Processing*, **37**, 545-558. 1998.
- Garcia, S., T. Kourti, and J. F. MacGregor, "Model Predictive Monitoring for Batch Processes", *Industrial & Engineering Chemistry Research*, **43**, 5929-5941. 2003.
- Garcia, S., T. Kourti, J. F. MacGregor, A. Matheos, and G. Murphy, "Troubleshooting of an Industrial Batch Process Using Multivariate Methods", *Industrial & Engineering Chemistry Research*, **42**, (15), 3592-3601. 2003.
- Gorret, N., A. K. bin Rosli, S. F. Oppenheim, L. B. Willis, P. A. Lessard, C. K. Rha, and A. J. Sinskey, "Bioreactor Culture of Oil Palm (*Elaeis Guineensis*) and Effects of Nitrogen Source, Inoculum Size, and Conditioned Medium on Biomass Production", *Journal of Biotechnology*, **108**, (3), 253-263. 2004.

- Hoskuldsson, A., "Quadratic PLS Regression", *Journal of Chemometrics*, **6**, 307-334. 1992.
- Jaekle, C. M. and J. F. MacGregor, "Product Design Through Multivariate Statistical Analysis of Process Data", *AICHE Journal*, **44**, (5), 1105-1118. 1998.
- Jaekle, C. M. and J. F. MacGregor, "Industrial Applications of Product Design Through the Inversion of Latent Variable Models", *Chemometrics and Intelligent Laboratory Systems*, **50**, 199-210. 2000a.
- Jaekle, C. M. and J. F. MacGregor, "Product Transfer Between Plants Using Historical Process Data", *AICHE Journal*, **46**, (10), 1989-1997. 2000b.
- Kaistha, N. and Ch Moore, "Extraction of Event Times in Batch Profiles for Time Synchronization and Quality Predictions", *Industrial & Engineering Chemistry Research*, **40**, 252-260. 2001.
- Kassidas, A., J. F. MacGregor, and P. Taylor, "Synchronization of Batch Trajectories Using Dynamic Time Warping", *AICHE Journal*, **44**, (4), 864-875. 1998.
- Kosanovich, K., K. S. Dahl, and M. J. Piovoso, "Improved Process Understanding Using Multiway Principal Component Analysis", *Industrial & Engineering Chemistry Research*, **35**, 138-146. 1996.
- Kourti, T., "Multivariate Dynamic Data Modeling for Analysis and Statistical Process Control of Batch Processes, Start-Ups and Grade Transitions", *Journal of Chemometrics*, **17**, 93-109. 2003.
- Kourti, T., J. Lee, and J. F. MacGregor, "Experiences With Industrial Applications of Projection Methods for Multivariate Statistical Process Control", *Computers and Chemical Engineering*, **20S**, S745-S750. 1996.
- Kourti, T. and J. F. MacGregor, "Multivariate SPC Methods for Process and Product Monitoring", *Journal of Quality Technology*, **28**, (4), 409-428. 1996.
- Kourti, T., P. Nomikos, and J. F. MacGregor, "Analysis, Monitoring and Fault Diagnosis of Batch Processes Using Multiblock and Multiway PLS", *J.Proc.Cont.*, **5**, (4), 277-284. 1995.
- Kozub, D. J. and J. F. MacGregor, "Feedback Control of Polymer Quality in Semi-Batch Copolymerization Reactors", *Chemical Engineering Science*, **47**, (4), 929-942. 1992a.
- Kozub, D. J. and J. F. MacGregor, "State Estimation for Semi-Batch Polymerization Reactors", *Chemical Engineering Science*, **47**, (5), 1047-1062. 1992b.

- Lakshminarayanan, S., H. Grosman Fuji, E. Dassau, and D. R. Lewin, "New Product Design Via Analysis of Historical Databases", *Computers and Chemical Engineering*, **24**, 671-676. 2000.
- Lennox, B., H. G. Hiden, G. A. Montague, G. Kornfeld, and P. R. Goulding, "Application of Multivariate Statistical Process Control to Batch Operations", *Computers and Chemical Engineering*, **24**, 291-296. 2000.
- MacGregor, J. F., C. M. Jaeckle, C. Kiparissides, and M. Koutoudi, "Process Monitoring and Diagnosis by Multiblock PLS Methods", *AICHE Journal*, **40**, (5), 826-838. 1994.
- MacGregor, J. F. and P. Nomikos, "Monitoring Batch Processes", NATO Advance Study Institute for Batch Processing Systems Engineering, Antalya, Turkey, Springer-Verlag, Heidelberg, 1992.
- Mathworks. *Statistics Toolbox User's Guide for Use With MATLAB*, 1999.
- McAuley, K. B. and J. F. MacGregor, "Optimal Grade Transitions in a Gas Phase Polyethylene Reactor", *AICHE Journal*, **38**, (10), 1564-1576. 1992.
- McAuley, K. B. and J. F. MacGregor, "Nonlinear Product Property Control in Industrial Gas-Phase Polyethylene Reactors", *AICHE Journal*, **39**, (5), 855-866. 1993.
- McAuley, K. B., J. F. MacGregor, and A. E. Hamielec, "A Kinetic Model for Industrial Gas-Phase Ethylene Copolymerization", *AICHE Journal*, **36**, (6), 837-850. 1990.
- Meng, X., A. J. Morris, and E. B. Martin, "On-Line Monitoring of Batch Processes Using a PARAFAC Representation", *Journal of Chemometrics*, **17**, 65-81. 2003.
- Miller, P., R. E. Swanson, and C. F. Heckler, "Contribution Plots: The Missing Link in Multivariate Quality Control", *Applied Mathematics and Computer Science*, **8**, 775-782. 1998.
- Nelson, P. R. C., P. Taylor, and J. F. MacGregor, "Missing Data Methods in PCA and PLS: Score Calculations With Incomplete Observations", *Chemometrics and Intelligent Laboratory Systems*, **35**, (1), 45-65. 1996.
- Neogi, D. and C. E. Schlags, "Multivariate Statistical Analysis of an Emulsion Batch Process", *Industrial & Engineering Chemistry Research*, **37**, 3971-3979. 1998.
- Nomikos, P., "Statistical Process Control of Batch Processes", McMaster University Ph.D. Thesis, 1995.
- Nomikos, P., "Detection and Diagnosis of Abnormal Batch Operations Based on Multi-Way Principal Component Analysis", *ISA Transactions*, **35**, 259-266. 1996.

- Nomikos, P. and J. F. MacGregor, "Monitoring Batch Processes Using Multiway Principal Components", *AICHE Journal*, **40**, (8), 1361-1375. 1994.
- Nomikos, P. and J. F. MacGregor, "Multi-Way Partial Least Squares in Monitoring Batch Processes", *Chemometrics and Intelligent Laboratory Systems*, **30**, 97-108. 1995a.
- Nomikos, P. and J. F. MacGregor, "Multivariate SPC Charts for Monitoring Batch Processes", *Technometrics*, **37**, (1), 41-58. 1995b.
- Ramaker, H. J., E. N. M. van Sprang, S. P. Gurden, J. A. Westerhuis, and A. K. Smilde, "Improved Monitoring of Batch Processes by Incorporating External Information", *J.Process Contr.*, **12**, 569-576. 2002.
- Rannar, S., J. F. MacGregor, and S. Wold, "Adaptive Batch Monitoring Using Hierarchical PCA", *Chemometrics and Intelligent Laboratory Systems*, **41**, 73-81. 1998.
- Rao, R., B. Manohar, K. Sambaiah, and B. R. Lokesh, "Enzymatic Acidolysis in Hexane to Produce N-3 or N-6 FA-Enriched Structured Lipids for Coconut Oil: Optimization of Reactions by Response Surface Methodology", *Journal of American Oil Chemists Society*, **79**, (9), 885-890. 2002.
- Schlags, C. E. and M. P. Popule, "Industrial Application of SPC to Batch Polymerization Process", *Proc.ACC (Arlington,VA)*, 25-27. 2001.
- Srinivasan, B., D. Bonvin, E. Visser, and S. Palanki, "Dynamic Optimization of Batch Processes II. Role of Measurements in Handling Uncertainty", *Computers and Chemical Engineering*, **27**, 27-44. 2003.
- Tates, A. A., D. J. Louwse, A. K. Smilde, G. L. M. Koot, and H. Berndt, "Monitoring a PVC Batch Process With Multivariate Statistical Control Charts", *Industrial & Engineering Chemistry Research*, **38**, 4769-4776. 1999.
- UMETRICS AB. *Tutorial to SIMCA-P, SIMCA-P+ Version 10*, Umea,Sweden ,2002.
- Undey, C., S. Ertunc, and A. Cinar, "Online Batch/Fed-Batch Process Performance Monitoring, Quality Prediction, and Variable-Contribution Analysis for Diagnosis", *Industrial & Engineering Chemistry Research*, **42**, (20), 4645-4658. 2003.
- van Sprang, E. N. M., H. J. Ramaker, J. Westerhuis, S. P. Gurden, and A. K. Smilde, "Critical Evaluation of Approaches for on-Line Batch Process Monitoring", *Chemical Engineering Science*, **57**, 3979-3991. 2002.

- Vassiliadis, V. S., W. H. Sargent, and C. C. Pantelides, "Solution of a Class of Multistage Dynamic Optimization Problems. 1 Problems Without Path Constraints", *Industrial & Engineering Chemistry Research*, **33**, 2111-2122. 1994a.
- Vassiliadis, V. S., W. H. Sargent, and C. C. Pantelides, "Solution of a Class of Multistage Dynamic Optimization Problems. 2. Problems With Path Constraints", *Industrial & Engineering Chemistry Research*, **33**, 2123-2133. 1994b.
- Wangen, L. E. and B. R. Kowalski, "A Multiblock Partial Least Squares Algorithm for Investigating Complex Chemical Systems", *Journal of Chemometrics*, **3**, 3-20. 1989.
- Westerhuis, J., T. Kourti, A. Kassidas, P. Taylor, and J. F. MacGregor, "Synchronizing the Trajectories of the Process Variables for on-Line Monitoring of Batch Runs With Unequal Duration", SSC6, Porsgrunn, Norway, August 15- 19, 1999.
- Westerhuis, J., T. Kourti, and J. F. MacGregor, "Analysis of Multiblock and Hierarchical PCA and PLS Models", *Journal of Chemometrics*, **12**, 301-321. 1998.
- Westerhuis, J., T. Kourti, and J. F. MacGregor, "Comparing Alternative Approaches for Multivariate Statistical Analysis of Batch Process Data", *Journal of Chemometrics*, **13**, 397-413. 1999.
- Westerhuis, J. A. and A. K. Smilde, "Short Communication: Deflation in Multiblock", *Journal of Chemometrics*, **15**, 485-493. 2001.
- Wold, S., "Cross Validatory Estimation of the Number of Components in Factor and Principal Components Models", *Technometrics*, **20**, 397-406. 1978.
- Wold, S., P. Geladi, K. Esbensen, and J. Ohman, "Multi-Way Principal Components and PLS-Analysis", *Journal of Chemometrics*, **1**, 41-56. 1987.
- Wold, S., N. Kettaneh, H. Friden, and A. Holmberg, "Modelling and Diagnostics of Batch Process and Analogous Kinetic Experiments", *Chemometrics and Intelligent Laboratory Systems*, **44**, 331-340. 1998.
- Wold, S., N. Kettaneh, and B. Skagerbergt, "Nonlinear PLS Modeling", *Chemometrics and Intelligent Laboratory Systems*, **7**, 53-65. 1989.
- Yabuki, Y. and J. F. MacGregor, "Product Quality Control in Semibatch Reactors Using Midcourse Correction Policies", *Industrial & Engineering Chemistry Research*, **36**, 1268-1275. 1997.

- Yabuki, Y., T. Nagasawa, and J. F. MacGregor, "An Industrial Experience With Product Quality Control in Semi-Batch Processes", *Computers and Chemical Engineering*, **24**, 585-590. 2000.
- Yacoub, F. and J. F. MacGregor, "Product Optimization and Control in the Latent Variable Space of Nonlinear PLS Models", *Chemometrics and Intelligent Laboratory Systems*, **70**, 63-74. 2004.
- Zhou, Y. H. and N. J. Titchener-Hooker, "Visualizing Integrated Bioprocess Designs Through "Windows of Operation"", *Biotechnology and Bioengineering*, **65**, (5), 550-557. 1999.