

Copyright

by

Yang Zhang

2008

The Dissertation Committee for Yang Zhang  
certifies that this is the approved version of the following dissertation:

**Improved Methods in Statistical and First Principles  
Modeling for Batch Process Control and Monitoring**

Committee:

---

Thomas F. Edgar, Supervisor

---

Roger T. Bonnecaze

---

John G. Ekerdt

---

Elmira Popova

---

Isaac C. Sanchez

---

Willy Wojsznis

# **Improved Methods in Statistical and First Principles Modeling for Batch Process Control and Monitoring**

by

**Yang Zhang, B.Eng., M.Eng.**

**Dissertation**

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

**Doctor of Philosophy**

**The University of Texas at Austin**

August 2008

Dedicated to Yan, Mom, and Dad.

# Acknowledgments

It has been my great honor to work with such a diligent and talented group of people during my PhD study in Austin. Foremost is our great leader, *Prof. Thomas F. Edgar*. His broad knowledge and extraordinary insights guided me and his careness toward students made my PhD life enjoyable. I also thank him for helping me to be a better technical writer and a more mature engineer.

I would like to thank those outstanding professors and engineers for serving on my thesis committee: *Prof. Roger T. Bonnecaze, Prof. John G. Ekerdt, Prof. Elmira Popova, Prof. Isaac C. Sanchez, and Dr. Willy Wojsznis*.

I was also fortunate enough to study together with many exceptional graduate students and visiting scholars including: *Sidharth Abrol, Kye Hyun Baek, Ivan Castillo, David Castiniera, Terry Farmer, Doug French, Bhalinder Gill, Hyung Lee, Ximing Liang, Blake Parkinson, Amogh Prabhu, Lina Rueda, Ben Spivey, Dirk Thiele, Dan Weber, Xiaoliang Zhuang, and Sepideh Ziaii*. This is an excellent group of people to be associated with from both academia and personal perspective.

I really appreciate Emerson Process Management engineers, *Terry Blevins, Michael Boudreau, Greg McMillan, Mark Nixon, Randy Reiss, Grant Willson, Willy Wojsznis, Chris Worek*, for supporting my research and sharing valuable industrial

perspective and experience. I enjoyed our collaboration and especially the warmth card.

Finally, I would like to say thank you to all my friends, importantly my former roommates *Xi Chen* and *Wen Li* for their great help and I wish them best luck in future career. I especially thank *Yan* and my family for their continuous support. I cannot measure how much you are meant to me and how fortunate to have you in my heart for all those goods and bads.

YANG ZHANG

*The University of Texas at Austin*

*August 2008*

# **Improved Methods in Statistical and First Principles Modeling for Batch Process Control and Monitoring**

Publication No. \_\_\_\_\_

Yang Zhang, Ph.D.

The University of Texas at Austin, 2008

Supervisor: Thomas F. Edgar

This dissertation presents several methods for improving statistical and first principles modeling capabilities, with an emphasis on nonlinear, unsteady state batch processes. Batch process online monitoring is chosen as a main research area here due to its importance from both theoretical and practical points of view.

Theoretical background and recent developments of PCA/PLS-based online monitoring methodologies are reviewed, along with fault detection metrics, and algorithm variations for different applications. The available commercial softwares are also

evaluated based on the corresponding application area. A detailed Multiway PCA based batch online monitoring procedure is used as the starting point for further improvements.

The issue of dynamic batch profile synchronization is addressed. By converting synchronization into a dynamic optimization problem, Dynamic Time Warping (DTW) and Derivative DTW (DDTW) show the best performance by far. To deal with the singularity point and numerical derivative estimation problems of DTW and DDTW in the presence of noise, a robust DDTW algorithm is proposed by combining Savitzky-Golay filter and DDTW algorithm together. A comparative analysis of robust DDTW and available methods is performed on simulated and real chemical plant data.

As traditional Multiway PCA-based (MPCA) methods consider batch monitoring in a static fashion (fail to consider time dependency between/within process variables with respect to time), an EWMA filtered Hybrid-wise unfolding MPCA (E-HMPCA) is proposed that considers batch dynamics in the model and reduce the number of Type I and II errors in online monitoring. Chemical and biochemical batch examples are used to compare the E-HMPCA algorithm with traditional methods.

First principles modeling is known to be time consuming for development. In order to increase modeling efficiency, dynamic Design of Experiments (DOE) is introduced for Dynamic Algebraic Equation (DAE) system parameter estimation. A new criterion is proposed by combining PCA and parameter sensitivity analysis (P-optimal criterion). The new criterion under certain assumptions reduce to several available criteria and is suitable for designing experiments to improve estimation of specific parameter sets. Furthermore, the criterion systematically decomposes a complex system into small pieces according to PCA. Two engineering examples (one



batch, one continuous) are used to illustrate the idea and algorithm.

# Contents

<b>Acknowledgments</b>	<b>v</b>
<b>Abstract</b>	<b>vii</b>
<b>List of Tables</b>	<b>xiv</b>
<b>List of Figures</b>	<b>xvi</b>
<b>Chapter 1 Introduction</b>	<b>1</b>
1.1 Industrial Process Modeling . . . . .	2
1.1.1 General Modeling Steps . . . . .	2
1.1.2 Modeling Technique Selection . . . . .	5
1.2 First Principles & Statistical Modeling Comparison . . . . .	6
1.3 Outline of this work . . . . .	10
<b>Chapter 2 Statistical Modeling for Process Monitoring</b>	<b>12</b>
2.1 Multivariate Statistical Process Control . . . . .	13
2.2 PCA, PLS Theory . . . . .	15
2.2.1 Principal Component Analysis . . . . .	15
2.2.2 Partial Least Squares . . . . .	19
2.3 Statistical Modeling for Process Monitoring . . . . .	23
2.3.1 Model Development . . . . .	23

2.3.2	Model Deployment . . . . .	24
2.3.3	Fault Detection Metrics . . . . .	24
2.3.4	PCA Variations & Applications . . . . .	25
2.4	Industrial Software Evaluation . . . . .	26
2.4.1	Software Packages . . . . .	27
2.4.2	Package Comparison . . . . .	32
2.5	Batch Monitoring Procedure . . . . .	34
2.5.1	Batch Data Synchronization (A2, B2) . . . . .	35
2.5.2	Three-dimensional data unfolding (A3) [98] . . . . .	36
<b>Chapter 3 Batch Profile Synchronization</b>		<b>41</b>
3.1	Dynamic Time Warping and its derivative . . . . .	42
3.1.1	Dynamic Time Warping Algorithm . . . . .	43
3.1.2	Derivative Dynamic Time Warping Algorithm . . . . .	45
3.2	Robust Dynamic Time Warping . . . . .	47
3.2.1	SG Filter . . . . .	47
3.2.2	Robust DDTW Algorithm . . . . .	52
3.2.3	Computational Procedure . . . . .	54
3.3	Case Study . . . . .	56
3.3.1	Industrial NIR Data . . . . .	56
3.3.2	Dynamic Trajectory Synchronization . . . . .	57
3.3.3	Industrial Data Synchronization . . . . .	63
3.4	Summary . . . . .	69
<b>Chapter 4 A Combined EWMA-HMPCA Technique</b>		<b>70</b>
4.1	Related Techniques . . . . .	71
4.1.1	HMPCA . . . . .	71
4.1.2	Batch Dynamic PCA . . . . .	73

4.1.3	EWMA and Multivariate EWMA [59]	74
4.2	EWMA-HMPCA	75
4.3	Case Study	78
4.3.1	Polymerization	79
4.3.2	Bioreactor Fermentation	83
4.3.3	Two Phase Chemical Batch Reactor	86
4.4	Summary	88
<b>Chapter 5</b>	<b>P-optimal Design Of Experiments</b>	<b>100</b>
5.1	DOE background	102
5.2	Related Techniques	104
5.2.1	Parameter Estimation	104
5.2.2	Design Of Experiments	106
5.2.3	Principal Component Analysis	110
5.3	PCA Combined Criterion for DOE	112
5.3.1	Information Matrix and Parameter Covariance Matrix	112
5.3.2	Geometric Interpretation	113
5.3.3	PCA based optimal Criterion (P-optimal) and its Variation	114
5.4	Case Study	118
5.4.1	Yeast Fermentation Reactor Model	118
5.4.2	Polymerization Reaction Model	129
5.5	Summary	138
<b>Chapter 6</b>	<b>Summary and Recommendations</b>	<b>144</b>
6.1	Summary of Contributions	144
6.2	Recommendations for Future Work	147
<b>Appendix A</b>	<b>Nomenclature</b>	<b>150</b>

<b>Appendix B Polymerization Reactor Model</b>	<b>155</b>
<b>Bibliography</b>	<b>157</b>
<b>Vita</b>	<b>169</b>

# List of Tables

1.1	First Principles & Statistical Modeling Comparison . . . . .	10
2.1	PCA Variations and Applications . . . . .	26
2.2	MSPC Software Comparison . . . . .	32
3.1	Synchronization Methods Comparison . . . . .	41
3.2	CPU Time Comparison by Different Algorithms . . . . .	69
4.1	BDPCA and HMPCA Comparison . . . . .	75
4.2	Bioreactor Fermentation Process Monitoring Results . . . . .	85
4.3	Chemical Batch Process Monitoring Results . . . . .	87
5.1	P-optimal algorithm description . . . . .	117
5.2	DOE Results by Different Criteria for Case I . . . . .	124
5.3	Parameter Estimation Results by Carrying Out Designed Experiments for Case I . . . . .	125
5.4	DOE Results by Different Criteria for Case II . . . . .	128
5.5	Parameter Estimation Results by Carrying Out Designed Experiments for Case II . . . . .	128
5.6	Parameter Estimation Results by Sequential DOE for Case II . . . . .	129
5.7	Unknown Parameters in Polymerization Model . . . . .	131

5.8	Parameter Estimation Results for Polymerization Process . . . . .	137
-----	---	-----

# List of Figures

1.1	Types of modeling undertaken, Figure 11 in [16]	4
2.1	Flow chart of basic Multiway PCA	35
2.2	PV trajectory from different batches	36
2.3	PV profile of a polymerization process	37
2.4	Unfolding data matrix according to batch-wise $\bar{X} \rightarrow X(I \times JK)$	38
2.5	Unfolding data matrix according to variable-wise $\bar{X} \rightarrow X(IK \times J)$	39
2.6	Unfolding data matrix according to time-wise $\bar{X} \rightarrow X(K \times IJ)$	40
3.1	Warping path by DTW	45
3.2	Alignment of two trajectories.	48
3.3	DDTW alignment path of two trajectories.	49
3.4	Derivative estimation of a noise corrupted sinusoid.	49
3.5	SG filter and exponential filter comparison.	53
3.6	Raw NIR trajectory from IDRC dataset. Blue: reference trajectory. Red: trajectory to be synchronized.	57
3.7	Synchronized NIR trajectory. Black dash: DTW. Green dash: DDTW. Blue dash: RDDTW	58
3.8	Dynamic simulation example.	59
3.9	Reference and new trajectories	59



3.10	Alignment path for different methods. . . . .	61
3.11	Alignment results comparison. . . . .	62
3.12	Alignment results with different parameter setting. . . . .	63
3.13	Reference and New Trajectories for Lubrizol chemical specialty Process. . . . .	64
3.14	Alignment path with different bandwidth. . . . .	66
3.15	Alignment results comparison for industrial data case. . . . .	67
3.16	Alignment results with RDDTW and different $h$ . . . . .	68
4.1	E-HMPCA algorithm . . . . .	78
4.2	Polymerization process online monitoring results by HMPCA . . . . .	81
4.3	Polymerization process online monitoring results by BDPCA . . . . .	82
4.4	Polymerization process online monitoring results by E-HMPCA . . . . .	83
4.5	Bioreactor fermenter . . . . .	90
4.6	Bioreactor process online monitoring results for another normal batch . . . . .	92
4.7	Bioreactor process online monitoring results for pH sensor bias batch . . . . .	94
4.8	Chemical batch reactor with temperature controller, Figure 3.9 in [60] . . . . .	95
4.9	Temperature sensor failure of chemical batch process . . . . .	97
4.10	Activation energy change of chemical batch process . . . . .	99
5.1	Geometric interpretation of PCA combined DOE criteria . . . . .	114
5.2	Control inputs and sampling points calculated by different optimal design criterion. Red: control limits; Blue: designed control inputs; Dash: initial control guesses; Square: sampling points . . . . .	123
5.3	Control inputs and sampling points calculated by sequential P-optimal design criterion. Red: control limits; Blue: designed control inputs; Dash: initial control guesses; Square: sampling points . . . . .	126

5.4	Control inputs and sampling points calculated by different optimal design criterion. Red: control limits; Blue: designed control inputs; Dash: initial control guesses; Square: sampling points . . . . .	139
5.5	Control inputs and sampling points calculated by different optimal design criterion. Red: control limits; Blue: designed control inputs; Dash: initial control guesses; Square: sampling points . . . . .	140
5.6	Gas phase polymerization system . . . . .	141
5.7	Control inputs and sampling points calculated by different optimal design criterion. Red: control limits; Blue: designed control inputs; Dash: initial control guesses; Square: sampling points . . . . .	143
6.1	Interface for testing PCA model . . . . .	146
6.2	Interface for testing PLS model . . . . .	147

# Chapter 1

## Introduction

*All models are wrong, but some are useful - G.E.P. Box*

Mathematical modeling has broad applications and a long history as an analysis and decision making tool across many process industries [37]. Despite the long history, modeling activities experienced rapid growth during the last few decades due to time and cost constraints for reducing experimental effort and need to improve product quality and meet environmental regulations by applying model-based technologies [26, 29]. However, whenever model-based techniques are included (control, optimization, scheduling, monitoring, etc.), modeling is often the most time consuming step. In other words, lack of high quality models are the bottleneck for many model-based technology applications.

According to Cameron and Ingram [16], in a survey of 72 experienced industrial modelers worldwide, “There is a strong belief that modeling is a good investment. The return on investment is seen as high by 54 percent and very high by a further 16 percent of respondents... The data clearly illustrate that the respondents’ organizations have realized the ‘value adding’ capability that modeling can contribute to system insight

and directly affect the operational performance of those companies.”

To overcome these difficulties, one crucial way is to build up efficient and generic modeling tools and procedures for practitioner use, which is the main objective of this work. In Section 1.1, we summarize the prevailing modeling procedures, current status of industrial modeling, and general ideas in selecting modeling tools. The reason of focusing on batch process is also explained. Section 1.2 compares the two types of modeling techniques (statistical modeling, first principles modeling) in more detail and Section 1.3 lists the scope of this work and the structure of the thesis.

## 1.1 Industrial Process Modeling

With the rapid growth of modeling applications in process industries, there have been a number of surveys focusing on related topics from modeling to computing use and to programming skills (for a summary, refer to Table 1 in [16]). In this study, we extract some important information from these first hand knowledge with the least bias possible. By doing this, the deficiencies can be identified and further improvements can be proposed.

### 1.1.1 General Modeling Steps

According to Foss et al. [29], industrial modeling practices generally include following steps:

1. Problem statement and initial data collection. The problem statement differs for *market pull* and *technology push* projects, which are beyond the scope here. Initial data and information collection may greatly affect the result and always be filtered by the purpose of study.

2. Modeling environment selection. Generally speaking, there are three types of simulators: 1. flowsheet (steady state model), 2. equation-oriented system (non-steady state; dynamic model), 3. block diagram-oriented simulator for control applications. Figure 1.1 shows the most recent survey results where other types of models are included. It is obvious that first principles, empirical (data driven, statistical) and predefined models are the most commonly used models. Other models are infrequently employed due to applicability reasons and thus beyond the horizon of this work. However, some surveys point out Computational Fluid Dynamics (CFD) modeling is more prevalent for important unit operations (e.g., reactors) [29, 16]. Among these frequently used model types, predefined models are case specific and the procedures are hard to generalize. Therefore, our attention focuses on expanding available knowledge and support for statistical and first principles approaches. Meanwhile, nearly all the surveys agree that dynamic model analysis is insufficient in available packages or softwares. It is reasonable to increase dynamic analysis support for both first principles and statistical modeling approaches.
3. Model structure buildup and simplification. This step is usually based on literature search and initial data analysis which heavily relies on process understanding, experience and creativeness.
4. Implementation and verification. Coding and debugging compose the main part of implementation. Verification checks if the codes reproduce the expected features of the process. After the mathematical equation is built up, coding and debugging can be carried out smoothly by using commercial softwares [29].
5. Validation. Based on different modeling objectives, this step can be different. For continuous process PID controller design, step response test is good enough

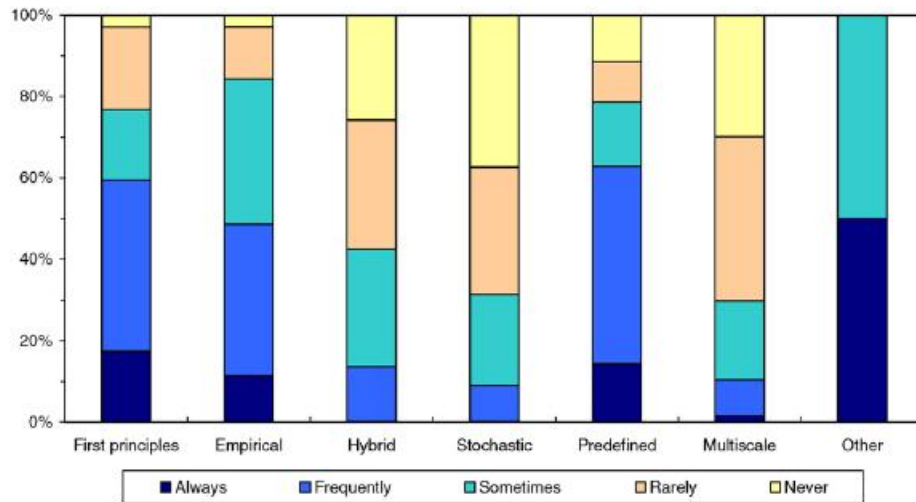


Figure 1.1: Types of modeling undertaken, Figure 11 in [16]

to test the validity of the model. For batch process MPC applications, multiple PRBS test with different initial batch conditions may be required. Also, most of the time, parameters are associated with a specific model structure defined a priori, validating the process verifies the correctness of model structure and parameter.

6. Documentation and life-cycle reuse. In industrial practice, models are only maintained if considered economically important [29]. This step is carried out more routinely among vendor companies than operating companies.

All the surveys agreed the steps listed above should be performed iteratively in a nonlinear fashion [29, 16]. Interestingly, more experienced modelers tend to carry out the task in a more unstructured and iterative way (Figure 1 in [29]). Following the above analysis, it is infeasible to regulate all the modeling steps and reasonable to focus our attention on providing and improving universal tools/methods for practitioner use. In this study, the two most widely used generic modeling ap-

proaches, statistical modeling and first principles modeling, are studied and detailed explanations are provided in Section 1.2 and 1.3.

### 1.1.2 Modeling Technique Selection

At the beginning of this chapter, the author quoted the famous saying by G.E.P. Box. With this in mind, the question is how to pick the suitable modeling procedure and technique that fits our goals. Pearson [71] provided four metrics in judging model utility for nonlinear control. With minor change, the rule can be used for most process model judgments:

1. approximation accuracy
2. physical interpretation
3. suitability for application goal
4. ease of development

Historically, low order linear models (e.g., first order plus time delay, second order plus time delay) enjoy great popularity because of criteria 3 and 4. Moreover, for many continuous process, linear models have acceptable accuracy around a specified steady state operation point. Furthermore, the different forms (state space, transfer function, discrete convolution, etc.) of linear models are equivalent and one can pick the favorable form according to specific application needs without losing information from other model form analysis. e.g., frequency analysis prefers transfer function; optimal control chooses state space model. Relatively mature tools and techniques have been developed for linear models across modeling, identification, control, and optimization applications [80, 57]. These linear models belong to the empirical model or statistical model family (all time series models are inherently statistically based).

However, for nonlinear processes (e.g. batch), there is no steady state and the system needs to be modeled by nonlinear structures. Generally speaking, nonlinear models can be divided into theoretical (first principles) based and statistically based ones. A detailed analysis on both model structures will be given in Section 1.2. In short, the equivalent characteristic disappears for most statistical based nonlinear dynamic models.

From above analysis, it can be concluded that modeling and related application tools are available for linear or approximately linear processes. Furthermore, many continuous processes can be treated as linear or linearized around certain operation conditions. Thus, continuous process modeling is in relatively good shape for many application purposes (e.g. classic control).

Batch is another important operation mode which is widely used to manufacture specialty chemicals, metals, electronic materials, ceramics, food, and agricultural materials, biochemicals and pharmaceuticals [80]. Besides numerous applications, theoretically, batch processes do not have steady states, usually operate over a broad set point range and have a very nonlinear behavior. Due to these uniqueness, linearization based approach is not sufficient and nonlinear modeling is required for many control, monitoring, and optimization purposes. From both application and theoretical points of view, efficient model development methods and procedures for batch process are highly desired.

## **1.2 First Principles & Statistical Modeling Comparison**

In Section 1.1, we reduce the scope of this work down to statistical and first principles modeling improvement due to their importance across process industries. In this section, the two modeling approaches are compared side by side from different



aspects including model assumptions, classifications, application areas, and ease to build model.

### ***Model Assumption and Output***

First principles modeling is always based on certain physical, chemical, or biological fundamentals such as reaction kinetics, mass balance, energy balance etc. As a result, based on different studying systems, the output of modeling would normally be Algebraic Equations (AE), Differential Algebraic Equations (DAE), Partial Differential Equations (PDE), SDE (Stochastic Differential Equations) or some combination of these. The unknown parameters of the mathematical systems are normally physical properties and reaction rate coefficients which are important and frequently not completely available [29]. To estimate these physically meaningful parameters, Maximum Likelihood Estimation (MLE) and Least Squares (LS) are always applied.

Statistical modeling is frequently data driven and there is no physical understanding included. When statistical rules (e.g., principal component analysis) are explicitly included, the resulting models can be Principal Component Analysis (PCA) models, Partial Least Squares (PLS) models, etc. When statistical rules are implicitly included, Auto-Regression with eXogenous (ARX) and Finite Impulse Response (FIR) models are normally resulted. The author used implicit here because the model structure itself does not based on statistical rules or assumptions but after certain manipulation, the model parameters actually rely on statistical analysis such as auto-correlation analysis, conditional distribution analysis, etc [57]. As described in Section 1.1.2, for linear system, ARX and FIR models are equal to each other but in nonlinear system, the corresponding derivatives Nonlinear ARX and Nonlinear FIR lose the equivalence property.

### *Model Applications*

First principles models are suitable for both interpolation and extrapolation applications because of their descriptive nature. As a result, first principles models can be applied to many application areas including control, optimization, monitoring, scheduling, and operator training. Furthermore, in most cases the model (e.g., DAEs) can be applied directly with minor or no further manipulations. Within all possible applications, first principles modeling is most suitable for nonlinear optimal control, large scale (plantwide) process optimization, and unit operation simulation with large nonlinearity and great importance.

Generally speaking, statistical models cannot be used in extrapolation cases. On the other hand, with proper treatments and manipulations, statistical modeling can be applied to nearly the same areas as first principles approach. However, based on different applications, the resulting models can be very different and the corresponding assumptions and techniques are quite different as well. For control applications, ARX and FIR based models are generally preferred; for monitoring purpose, PCA and PLS based models are prevailing; for optimization, neural networks and PLS models can be used. One can see that instead of getting DAE type of model for all applications, statistical modeling follows different procedures and results in various models for each application. Researchers have been putting great efforts in developing model structures and modeling procedures for each applications, e.g., [76, 71, 80, 57].

In general, a first principles model has much fewer model building assumptions, much broader applications and offers greater insights of the objective system than a statistical model. This is especially true for highly nonlinear continuous and batch processes.

### *Ease to Build Model*

First principles models are inherently hard to build due to following reasons: 1. lack of process knowledge, so the model structure is difficult to set up. 2. lack of measurements, so the model parameters are hard to identify with accuracy. Due to above reasons, many iterations and experiments are required to get a high fidelity first principles model for control related applications.

Statistical models are relatively easy to build when certain modeling goals are available in terms of time and cost. However, as argued previously, statistical models may fit for certain applications but fail to offer much process insights, especially for complex and large scale systems.

”One interviewee states, for example, that empirical models can be developed at about a tenth of the development cost for mechanistic models. It is, however, difficult to assess the overall benefit of an empirical model because much less new process knowledge is developed compared to first principles based modeling. Typically, such a model cannot be used as a basis for later applications.” - Foss et al., survey on 16 experienced industrial modelers in German and Norway [29].

As a summary, Table 1.1 lists the key points compared so far and it can be concluded that both first principles and statistical modeling have their own uniqueness and application advantages. Furthermore, for a control/process engineer, it is reasonable to start from the modeling objective to choose the best approach.

Model Type	First Principles	Statistical
Simulation Usage	Interpolation and extrapolation; wide set point changes	Interpolation only; around certain set point change
Modeling Assumptions	Physical/Chemical/Biological rule based	Data driven, statistical relation oriented
Suitable Application Areas	Nonlinear control; Plantwide optimization; Operator training	Process monitoring by PCA type of model; Process control by ARX type of model.
Ease to Build	Hard, experiment demanding and time consuming	Relatively easy, certain procedures are available
Current Difficulty	Model structure buildup; Model parameter estimation	Improve the modeling procedures for each type of model (specific application objectives)

Table 1.1: First Principles & Statistical Modeling Comparison

### 1.3 Outline of this work

In Sections 1.1 and 1.2, by analyzing current trends of industrial modeling, the importance and shortcomings of statistical and first principles modeling are discussed. The challenges in industrial modeling can be summarized as follows. At first, for all modeling applications, batch processes are difficult to tackle due to its nonlinear dynamics and non steady states. At the mean time, batch is an important operation mode across many process industries. As a result, most part of this work is focusing on batch process. Furthermore, for statistical modeling, although generic procedures are available, there is still plenty of room to improve and polish the procedure for each application area. For first principles modeling, reducing experimental time and cost in model structure buildup and parameter estimation are of special interest.

Chapters 2 to 4 focus on improving the methods for a specific statistical modeling application: batch process online monitoring. The reason for choosing this topic is explained in Chapter 2. Then, the key methodologies (PCA and PLS, originated first

in regression analysis) are introduced. Also, prevailing industrial batch monitoring procedures and commercial software packages are reviewed. Chapters 3 and 4 focus on improving two important steps in the whole online batch monitoring procedures, which are batch trajectory synchronization and batch dynamic consideration for monitoring. Some case studies are given after theoretical analysis in each chapter. The final objective of this part is to propose and demonstrate the effectiveness of an improved computational procedure (algorithm) for industrial batch process monitoring, and hopefully, the package can be commercialized.

For first principles modeling, a new criteria of DAE system Design of Experiments (DOE) is proposed by combining sensitivity analysis with PCA. After a concise introduction of DOE, Chapter 5 introduces the existing techniques and our new combined criteria in a systematic way. The new criteria includes as special cases the most widely used criteria, e.g., D-optimal and E-optimal. It is especially suitable for complex nonlinear system parameter estimation where dozens of parameters needs to be estimated. The whole DOE and parameter estimation algorithm is tested by two engineering case studies. Although dynamic DOE for parameter estimation is the main focus here, DOE for model structure buildup is briefly discussed as well. By employing DOE, the iteration process and related experimental costs in first principles modeling is expected to be reduced.

Chapter 6 serves as a conclusion of this work where the key results and future directions are pointed out.

## Chapter 2

# Statistical Modeling for Process Monitoring

In Chapter 1, the methodology, application areas and challenges of statistical modeling are reviewed by comparing with the first principles approach. This chapter focuses on introducing some important multivariate statistical process control (MSPC) techniques, e.g., PCA and PLS, and points out the general difficulties in real applications, which leads to the specific solutions given in Chapters 3 and 4. To provide working details on applying MSPC, this chapter is organized as follows. Section 2.1 provides introductory background information on MSPC. In Section 2.2, PCA and PLS theories, NIPALS algorithms, and their characteristics are discussed. Section 2.3 explains the main steps and important terminologies in applying PCA and PLS to process monitoring. Some commercialized software packages are reviewed in Section 2.4. Finally, Section 2.5 introduces the structured batch process online monitoring procedure and challenges for this application are also discussed.

## 2.1 Multivariate Statistical Process Control

Traditionally, statistical process control (SPC) charts such as Shewhart, CUSUM, and EWMA charts [22, 62] have been used to monitor industrial processes for the purpose of improving product quality. Such techniques are well developed for manufacturing processes and widely used in univariate systems in which a single process variable is monitored and normal distributed noise assumption applied. The SPC models are always based on mean and standard deviation calculations of each variable within a historical period [17]. However, industrial process usually includes more than one process variable, and univariate SPC charts do not perform as well for these multivariate systems [62, 54, 63]. For example, in chemical and biological systems, hundreds of variables can be recorded regularly in a single operating unit, resulting in a large data set that is hard to analyze with univariate methods. Furthermore, because many of the variables are correlated, they must be considered all together rather than individually. To meet these requirements, MSPC methods can be employed to reduce the dimension of the large raw data set while extracting useful information such as the existence of faults or abnormalities.

Since the late 1980s, industry has applied multivariate statistical projection methods to detect abnormal conditions and to diagnose the root cause of these situations [87, 76]. In nearly all applications, two basic techniques are applied: principal component analysis (*PCA*) and partial least squares (*PLS*). *PCA* is a dimensional reduction method that identifies a subset of uncorrelated vectors (principal components) so as to capture most variance in the data. *PLS* is a decomposition technique that maximizes the covariance between predictor and predicted variables for each component. In some cases, *PCA* can be applied to a data set before using *PLS* to reduce the variable dimension for *PLS* analysis. This step makes *PLS* much more efficient to use, Hoskuldsson [40] and MacGregor and Kourti [62] provide more details

on PLS.

During the past ten years, MSPC enjoys more and more popularity and has been applied across many industries including semiconductors [17, 20], chemicals [78], mining [49], and petrochemicals [2]. Miletic et al. [66] performed a comprehensive review of the applications of multivariate statistics in the areas of steel and pulp and paper. Kourti et al. [50] showed the power of MSPC in polymer processing.

In comparison with these industries, the application of MSPC in food, biological, and pharmaceutical industry is relatively immature. With online analyzers, plants are able to collect a tremendous amount of historical data. However, efficient tools are needed to manage these data and extract useful information. MSPC is a good candidate for this purpose as well as for achieving the manufacturing goals of high quality, efficiency and low cost. Increasing attention from both academic and industrial researchers has been focused on this promising area and it is also one important focus of this work. In 2001, Albert and Kinley [1] from Eli Lilly Company applied principal component analysis to an industrial batch tylosin biosynthesis process. In 2001, Lennox et al. [55] published results on different MSPC approaches for a fed-batch fermentation system operated by Biochemie GmbH. An MSPC application in a pharmaceutical fermentation process was studied by Lopes et al. [58] in 2002. Also, Chiang et al. [21] compared the performance of three MSPC methods in an industrial fermentation process at the San Diego biotech facility of the Dow Chemical Company. Gunther et al. [36] applied PCA to an industrial fed-batch cell culture process data which are gathered from Amgen Pilot Plant. Although industrial data or high fidelity simulation data are used, most of the analysis are carried out after the whole batch is finished (*offline*).



## 2.2 PCA, PLS Theory

### 2.2.1 Principal Component Analysis

In 1873-1874, Beltrami and Jordan independently derived the singular value decomposition (SVD) method that underlies PCA [45]. Meanwhile, the earliest work on PCA was done by Pearson in 1901 and Hotelling in 1933 [45, 43]. One of the most prevailing algorithm is NIPALS suggested by H. Wold in 1960s [91]. However, due to its heavy computational requirements, the spread of PCA was limited until personal computers became available in the 1980s. Since then PCA has been widely used in many areas, including agriculture, biology, chemistry, climatology, demography, ecology, economics, food research, genetics, geology, meteorology, oceanography, psychology, and process control [45]. According to the *ISI Web of Knowledge* (<http://isiknowledge.com>), nearly 14,000 technical articles used principal component analysis as keywords from 2000 to 2007.

#### PCA Objective

PCA focuses on finding the largest variance directions in an  $m$  dimensional ( $m$  usually large for industrial applications) data space. In order to do this, data matrix  $X(n \times m, n > m)$  is projected by [74]:

$$t = X \times p \quad (2.1)$$

where  $t(n \times 1)$  and  $p(m \times 1)$  are called leading score and loading vector, respectively.

In order to find the direction to maximize data covariance, Eq. 2.2 is built up:

$$\max J = t^T t = (Xp)^T \times (Xp) = p^T X^T X p \quad (2.2)$$

It can be seen that there are two variables ( $t$  and  $p$ ) and only one governing equation (Eq. 2.1). To solve the minimization problem, another constraint should be specified. In PCA, the constraint is chosen to be:

$$p^T \times p = 1 \quad (2.3)$$

Applying Lagrange multiplier, the appended index  $J'$  is:

$$J' = p^T X^T X p + \lambda (1 - p^T p) \quad (2.4)$$

At the minimum of Eq. 2.4, the first partial derivative of  $J'$  with respect to  $p$  should be equal to zero:

$$\frac{\partial J'}{\partial p^T} = X^T X p - \lambda p = 0 \quad (2.5)$$

which means:

$$X^T X p = \lambda p \quad (2.6)$$

Thus,  $\lambda$  and  $p$  are the leading eigenvalue and eigenvector of the covariance matrix ( $X^T X$ ), respectively. Similarly, the remaining loading vectors of PCA are equal to the corresponding eigenvectors of the covariance matrix ( $X^T X$ ). From linear algebra, it is known that the eigenvectors are orthogonal ( $p_i^T \times p_j = 0, i \neq j$ ) and Eq. 2.3 suggested  $p$  vectors are also normalized ( $p_i^T \times p_i = 1$ ).

After sequentially calculating  $t_i$ ,  $p_i$  and concatenating them into score and loading matrices:  $T = [t_1, t_2, \dots, t_m]$  and  $P = [p_1, p_2, \dots, p_m]$ , we obtain the widely accepted equation for PCA:

$$X = T \times P^T \quad (2.7)$$

## NIPALS for PCA

From Eq. 2.6, it can be seen that  $P$  and  $T$  matrices can be calculated by singular value decomposition (SVD) directly. However, in real applications, nonlinear iterative partial least squares (NIPALS) algorithm proposed by H. Wold [91] is widely used because its missing data handling ability. Furthermore, NIPALS uses iteration method to calculate one principal component at a time. Different from finding the largest variation of the covariance matrix (Eq. 2.2), NIPALS algorithm focuses on minimizing the following objective function:

$$\min \|X - tp^T\|_F \quad (2.8)$$

where

$$p^T p = 1$$

and  $F$  stands for Frobenius norm.

NIPALS objective function is different from the PCA objective function (Eq. 2.2). It is necessary to judge if the two objectives are the same. To do this, least squares solution to Eq. 2.8 is calculated:

$$p^T = (t^T t)^{-1} t^T X \quad \Rightarrow \quad p = X^T t (t^T t)^{-1} \quad (2.9)$$

Similarly, the least squares solution to the transpose of Eq. 2.8 ( $\min \|X^T - pt^T\|_F$ ) is

$$t^T = (p^T p)^{-1} p^T X^T \quad \Rightarrow \quad t = X p \quad (2.10)$$

Substitute 2.10 into 2.9, such that:

$$p = X^T (X p) (p^T X^T X p)^{-1} \quad (2.11)$$

We call the scalar  $(p^T X^T X p)^{-1}$  as  $\lambda$  and Eq. 2.11 changes to:

$$p = \lambda^{-1} X^T X p \quad \Rightarrow \quad \lambda p = X^T X p \quad (2.12)$$

which indicates the objective function of NIPALS (Eq. 2.8) is equivalent to extract largest variation introduced in PCA objective function (Eq. 2.2).

Computationally, NIPALS algorithm can be carried out by iteration and deflation as described in [34]:

1. select a column vector  $x_j$  from  $X$  as the starting point of  $t_i$ .
2. calculate  $p_i$  by Eq. 2.9:

$$p_i = X^T t_i (t_i^T t_i)^{-1}$$

3. calculate  $t_i$  by Eq. 2.10:

$$t_i = X p_i$$

4. compare  $t_i$  used in step 2 and calculated from step 3. If the difference is smaller than certain convergence rule, go to step 5, otherwise go to step 2.
5. deflate  $X$  by  $X = X - t_i \times p_i$ , set  $i = i + 1$  and go to step 1.

Instead of using raw measurement data, preprocessed data are required here. Preprocessing is crucial to PCA and different preprocessing criteria will lead to different results as further discussed in Section 2.5. Among many different preprocessing methods,  $X$  is scaled into zero mean and unit variance for each column in most process monitoring applications.

Due to the iterative nature of NIPALS, there might be convergence concerns when very similar eigenvalues exist in the covariance matrix. Luckily, in real applications,

the leading ones, which are important for process monitoring, are always different from each other (usually with different magnitude).

### 2.2.2 Partial Least Squares

As an important statistical regression technology, PLS was proposed by H. Wold [91] in 1966 where it was first applied to economic data analysis. Interestingly, PLS become popular in chemometrics society, which is partially due to his son S. Wold and H. Martens in the 80s [34]. After that, together with the rapid growth of computational speeds, PLS became a widely accepted tool in many fields including engineering, bioinformatics, food and medicine research, genetics, physiology, etc.

#### PLS Objective

In regression analysis, input and output data matrices  $X$  and  $Y$  can be related to model parameters  $\theta$  and structures  $f$  as described in Eq. 2.13:

$$Y = f(\theta, X) + E \quad (2.13)$$

If  $f$  is linear with respect to the parameters, Eq. 2.13 assumes a linear regression form:

$$Y = X\theta + E \quad (2.14)$$

Suppose the dimensions of  $X$  and  $Y$  are  $n \times m$  and  $n \times l$ , respectively. In process industry,  $n$ ,  $m$ , and  $l$  are the number of observations, inputs, and outputs. Normally,  $n > (m \text{ or } l)$  and Eq. 2.14 is over-determined. Ordinary least squares (OLS) is a common tool in calculating over-determined system and the objective function can be written as:

$$J = \min_{\theta} (\|Y - X\theta\|_F) \quad (2.15)$$

The well-known minimization result is:  $\hat{\theta} = (X^T X)^{-1} X^T Y$ . This is an unbiased estimation and the Frobenius norm of  $E$  in Eq. 2.13 is guaranteed to be a minimum with respect to  $\hat{\theta}$ . However, OLS fails when  $X$  has highly correlated columns because  $X^T X$  will be rank deficient so it cannot be inverted. In industrial processes, correlation may arise from physical relationships (e.g., mass balance, energy balance [76]).

PLS can solve the collinear problem in OLS which is a more robust and reliable algorithm and the PLS objective is framed as [73]:

$$X = tp^T + E \quad (2.16)$$

$$Y = uq^T + F \quad (2.17)$$

where  $t, p$  are  $n \times 1, m \times 1$ , and  $u, q$  are  $n \times 1, p \times 1$  vectors, respectively.

Instead of minimizing  $E$  and  $F$ , PLS chooses to maximize the correlation between  $t$  and  $u$ :

$$J = \max(t^T u) = \max(u^T t) \quad (2.18)$$

subject to the following constraints:

$$\begin{aligned} t &= Xw, & u &= Yq \\ \|w\|^2 &= 1, & \|q\|^2 &= 1 \end{aligned}$$

Substitute  $t = Xw$ ,  $u = Yq$ , and applying Lagrange multiplier to Eq. 2.18, we obtain:

$$J' = \max(w^T X^T Y q) + \frac{1}{2} \lambda_q (1 - q^T q) + \frac{1}{2} \lambda_w (1 - w^T w)$$

To find the minimum, the first partial derivative is taken with respect to  $w$  and  $q$  and set them to zero:

$$\frac{\partial J'}{\partial q} = w^T X^T Y - \lambda_q q^T = 0 \quad (2.19)$$

$$\frac{\partial J'}{\partial w} = q^T Y^T X - \lambda_w w^T = 0 \quad (2.20)$$

By rearranging Eq. 2.19 and 2.20, we have:

$$w^T X^T Y q = \lambda_q \quad \Rightarrow \quad X^T Y q = \lambda_q w \quad (2.21)$$

$$q^T Y^T X w = \lambda_q \quad \Rightarrow \quad Y^T X w = \lambda_w q \quad (2.22)$$

Substitute Eq. 2.21, 2.22 into original PLS objective function Eq. 2.18 and take consideration of the constraints ( $w^T w = 1$ ,  $q^T q = 1$ ):

$$J = \max(t^T u) = \max(w^T X^T Y q) = \max(w^T \lambda_q w) = \max(\lambda_q) \quad (2.23)$$

$$J = \max(u^T t) = \max(q^T Y^T X w) = \max(q^T \lambda_w q) = \max(\lambda_w) \quad (2.24)$$

There are some important features from above derivations including:  $\lambda_q = \lambda_w$ ;  $X^T Y Y^T X$  and  $Y^T X X^T Y$  shares the same eigenvalues ( $\lambda_q \times \lambda_w$ ) and the associated eigenvectors are  $w$  and  $q$ , respectively. Similar to PCA, after one pair of  $t_i$ ,  $p_i$ ,  $u_i$ ,  $q_i$ , and  $w_i$  is identified, the data matrices  $X$  and  $Y$  are deflated. Then the next pair will be calculated following the same routine. In this way, the big loading and score matrices are built up:

$$X = T P^T$$

$$Y = U Q^T$$

## NIPALS for PLS

Same as for PCA, NIPALS is an efficient, robust and the most widely used algorithm in PLS calculation. The wide popularity of NIPALS is partially due to the outstanding tutorial paper by Geladi and Kowalski [34]. The general steps of NIPALS include:

1. select column vectors  $x_j, y_j$  from  $X$  and  $Y$  as the starting point of  $t_i$  and  $u_i$
2. outer modeling by iteration through:

$$\begin{aligned}w_i &= \frac{X^T u_i}{\|X^T u_i\|} \\t_i &= X w_i \\q_i &= \frac{Y^T t_i}{\|Y^T t_i\|} \\u_i &= Y q_i\end{aligned}$$

3. inner modeling:

$$b_i = \frac{u_i^T t_i}{t_i^T t_i}$$

4. compare  $t_i$  used in step 2 and calculated from step 3. If the difference is smaller than certain convergence rule, go to step 5, otherwise go to step 2.
5. model deflation by:

$$\begin{aligned}p_i &= \frac{X^T t_i}{t_i^T t_i} \\X &= X - t_i p_i^T \\Y &= Y - b_i t_i q_i^T\end{aligned}$$

set  $i = i + 1$  and go to step 1.

This algorithm is named PLS-I and there is another variation called normalized inner model (PLS-II). In PLS-II,  $b_i$  is equal to one by carrying out certain nor-



malization steps in outer modeling. Based on this core structure, there are many variations focusing on speed, robustness, and accuracy improvements [23].

## 2.3 Statistical Modeling for Process Monitoring

Section 2.2 indicates PCA and PLS can extract and rank data correlations within (or between) a matrix according to their importance (size of eigenvalues). In industrial applications, when hundreds of process variables (PVs) are routinely measured, important correlations capture the features of the plant. In other words, instead of observing all PVs, monitoring the important correlations is more efficient and precise as process noise is marked as unimportant or uncorrelated terms by the algorithm automatically as shown below. In process monitoring applications, PCA and PLS are applied in a similar fashion so PCA is picked as an example here.

### 2.3.1 Model Development

Suppose process data are stored in a matrix  $X_{raw}$ ,  $n \times m$ , where  $n$  is the number of observations and  $m$  is the number of sensors [76, 17]. After scaling the matrix to zero mean and unit variance for each column, the data matrix ( $X$ ) is decomposed by PCA:

$$X = [T\tilde{T}] \times [P\tilde{P}] = T \times P + \tilde{T} \times \tilde{P} \quad (2.25)$$

where  $T$ ,  $P$  include score and loading vectors for process features part, corresponding to large eigenvalues of the covariance matrix of  $X$ ;  $\tilde{T}$  and  $\tilde{P}$  contain score and loading vectors dominated by noises (small eigenvalues). The space spanned by important loading vectors ( $P$ ) is named *principal subspace* while  $\tilde{P}$  spanned *residual subspace*.

It is crucial to decide the number of principal components ( $n_{cp}$ ) that goes to principal subspace ( $S_p$ ) and residual subspace ( $S_r$ ). For PCA, there are numerous methods and according to our experience and the published literature [98, 86, 39],

Cross Validation [92], Variance of the Reconstruction Error [86], and Parallel Analysis are proved to be robust and reliable. For PLS, Cross Validation [92] is the most widely used method.

### 2.3.2 Model Deployment

After a new observation  $x_{raw}$ ,  $(m \times 1)$  is collected online, it is normalized by  $x_{new} = \frac{x_{raw} - x_{mean}}{x_{std}}$  where  $x_{mean}$  and  $x_{std}$  are the column mean and standard deviation of  $X_{raw}$  calculated in model building step. This is called online data preprocessing.  $x_{new}$  is projected to principle and residual subspace by:

$$x_p = PP^T x_{new} \quad (2.26)$$

$$x_r = (I - PP^T)x_{new} \quad (2.27)$$

### 2.3.3 Fault Detection Metrics

Usually, squared prediction error (SPE, or  $Q$  statistic) and Hotellings  $T^2$  are calculated for each rescaled observation ( $x_{new}$ ) by:

$$SPE = \|x_r\| = \|(I - PP^T)x_{new}\| \quad (2.28)$$

$$T^2 = t_{new}\Lambda^{-1}t_{new}^T = x_{new}P\Lambda^{-1}P^T x_{new}^T \quad (2.29)$$

$T^2$  measures the variance of  $x_{new}$  within the principal subspace model while SPE projects  $x_{new}$  to the residual subspace. For more discussion on the asymmetric role of  $T^2$  and SPE in process monitoring, please consult [76]. If  $X$  follows multivariate normal distribution,  $T^2$  follows  $F$  distribution with  $n_{pc}$ ,  $n - n_{pc}$  degrees of freedom and the corresponding upper control limit with  $\alpha$  confidence level is:

$$T_\alpha^2 = \frac{n_{pc}(n^2 - 1)}{n(n - 1)} F_{n_{pc}, n - n_{pc}, \alpha} \quad (2.30)$$

Similarly, SPE upper control limit with a confidence level  $\alpha$  is [44]:

$$SPE_\alpha = \theta_1 \left( \frac{c_\alpha \sqrt{2\theta_2 h_0^2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right) \quad (2.31)$$

where

$$\begin{aligned} \theta_1 &= \sum_{j=n_{pc}+1}^m \lambda_j^i, \quad i = 1, 2, 3 \\ h_0 &= 1 - \frac{2\theta_1\theta_3}{3\theta_2^2} \end{aligned}$$

An alternative way to calculate upper control limit for SPE is derived in [70]:

$$SPE_\alpha = g\chi_{h,\alpha}^2 \quad (2.32)$$

where

$$g = \frac{\theta_2}{\theta_1}; h = \frac{\theta_1}{\theta_2}$$

The relationship between the two approaches are explained in [70].

Recently, Yue and Qin [96] proposed the so-called combined metric which combines  $T^2$  and SPE metrics by certain weighting matrices. In process monitoring applications, SPE is proved to be more sensitive to process faults and preferred over  $T^2$ .

If the calculated metric value is smaller than the corresponding upper control limit,  $x_{raw}$  is classified as good. Otherwise, if several observations (e.g., five) consistently indicate upper control limit violation, process fault is propagating in the system and an alarm is triggered.

### 2.3.4 PCA Variations & Applications

Based on these basic algorithms and metrics, there are many variations which have their specific applications and advantages. Table 2.1 listed some of these variations

and their usage in chemical and biochemical related applications [98].

Processes	Method	References	Method Features
Time invariant continuous process	PCA	Krestal et al. [51]	Reduce dimension
Time invariant continuous process	Recursive PCA	Li et al. [56]	Adaptive model updating
Dynamic process	Dynamic PCA	Ku et al. [53]	Time correlation between variables is considered
Nonlinear continuous process	Nonlinear PCA	Dong and McAvoy [25]	Nonlinearity is considered
Large-scale process	Multiblock PCA	MacGregor et al. [61]	Decompose large plant into small blocks
Multiscale process	Multiscale PCA	Bakshi [10]	Deal with hierarchy structures
Continuous or batch	Model-based PCA	Rotem et al. [78]	First principle model is used
Batch process	Multiway PCA	Nomikos and McGreggor [68, 70, 69]	Data unfolding is applied

Table 2.1: PCA Variations and Applications

## 2.4 Industrial Software Evaluation

One purpose of this work is to develop a generic software package for industrial (especially pharmaceutical or biological) batch process monitoring. From a practical point of view, it is important to evaluate current commercial software features including supported algorithm, application areas, and integration with other automation hardware and softwares. The evaluation is based on our direct experience with some packages, the vendors' product information, and user guides. Some of the software can be applied to areas beyond MSPC, however, in this report we will concentrate on evaluating the product features in process monitoring.

### 2.4.1 Software Packages

#### AspenTech

*Aspen Multivariate* is designed for developing and deploying sophisticated multivariate SPC models that are deployed online using the InfoPlus data historian and applications platform.

*Product Features:* Aspen Multivariate software suite requires online data from InfoPlus (data server of AspenTech) and analyzes them to provide condition monitoring, process characterization, instrument and unit fault detection and diagnosis, as well as process visualization, through a single graphical user interface. Five PCA algorithms can be chosen. Four trending control charts are available for Aspen Process Explorer to visualize and interpret the events, including Hotelling's  $T^2$  plot and  $Q$ -residual plot.

*Application Areas:* The five PCA algorithm is not clearly identified and it is impossible to judge if this package can be applied to batch processes. However, it appears that this suite can be applied to any continuous process.

#### Brooks Automation

Brooks Automation Inc. produces both hardware and software. In this work, only Statistical Process Control software products are analyzed. The name of the software is Brooks FDC. According to their website, FDC is a real-time advanced process control application package within the Brooks Sense Decide Respond real-time solution set.

*Product features:* The Brooks FDC (Fault Detection and Classification) is a software application that enables manufacturing person to monitor production and respond to equipment health issues. The module integrates into 200mm and 300mm

factories through the equipment automation interface and supports both standard sensor and custom data acquisition. Brooks FDC analyzes the data, classifies the fault and notifies operators to address the issue. FDC develops and models virtual health sensors or uses current available data to represent sensors where they currently do not exist. Users can troubleshoot and resolve process and equipment problems upon receiving fault notification from the Brooks FDC auto-notification module, bringing the benefit of Internet communications to plant-wide processes and equipment monitoring.

*Application Areas:* Brooks Automation mainly focuses on semiconductor industry plus some other discrete manufacturing areas so the software package is suitable for batch process but due to the uniqueness of the semiconductor manufacturing processes, the software may not be able to apply to other batch industries.

## **CAMO**

CAMO was founded by Arne Tysso in Norway in 1984. CAMO has headquarters in Oslo, Norway with offices in Woodbrige, New Jersey USA, and Bangalore, India. The flagship product of CAMO is the Unscrambler software.

*Product Feature:* Unscrambler is the major MSPC product from CAMO and its important features are,

- PCA modeling
- Multivariate Curve Resolution
- PLS regression
- 3-way PLS regression
- Clustering (K-Means)

- Automatic pretreatments in prediction and classification

*Application Areas:*

- Pharmaceutical & Biotechnology
- Chemical Manufacturing
- Food & Beverage
- Oil & Gas
- Pulp & Paper

**StatSoft**

StatSoft was founded in 1984 as a partnership of a group of university professors and scientists. It is used in mission critical manufacturing applications, in regulated FDA controlled industries [also to help achieve compliance with CFR Part 11 and Sarbanes-Oxley (SOX) regulations] and as a foundation of corporate-wide Six Sigma initiatives. The main product is Statistica, which provides four basic categories of product lines: Enterprise; Web-based analytic applications; Data mining solutions and Desktop.

*Product Feature:* The MSPC solution in Statistica belongs to Enterprise solutions. In all, an Enterprise-wide SPC system contains Statistical Enterprise-wide SPC System (SEWSS) and Monitoring and Alerting Server (MAS) with some multivariate statistical algorithm. The specific features include:

- Apply univariate and multivariate statistical methods for quality control, predictive modeling, and data reduction to complex manufacturing processes
- Determine the most critical process, raw materials, and environment factors and their optimal settings for delivering products of the highest quality

- Monitor the process characteristics interactively or automatically during production stages, rather than waiting for final testing
- Build, evaluate and deploy predictive models based on the known outcomes from historical data

*Analytical Capabilities and Algorithms:*

- Partial Least Squares
- Principal Components
- Neural Networks
- Independent Components Analysis (ICA)
- Support Vector Machines
- Cross-validation

*Application Areas:* Statistica can be applied to process industries, batch-oriented manufacturing such as pharmaceutical, chemical, petrochemical, pulp and paper, food manufacturing, semiconductor, and health care. Five out of the world's 10 largest pharmaceutical companies use Statistica, not necessarily MSPC application though.

## **Umetrics**

Umetrics is founded by Svante Wold and Rolf Carlsson with their two graduate students in 1987 and now it has more than 50 employees. Umetrics focuses on both statistical process control and design of experiments and have quality software packages available. In this work, we only examine MSPC related ones.



SIMCA-P and SIMCA-4000 are multivariate data analysis tools which can only be used in continuous processes. SIMCA-P+ is an extension of SIMCA-P which can analyze both continuous and batch processes. SIMCA-Batch On-Line execute batch models built with the SIMCA-P+. In other words, SIMCA-Batch can get data online from process through different supported interfaces (e.g., OPC, PI, FIX, MOPS, Oracle, SQL) or from a file and perform monitoring by deploying the model from SIMCA-P+.

*Product Feature:* SIMCA-P+ is analyzed in this section due to its batch handling ability. First of all, SIMCA-P+ has all features that SIMCA-P has. It can be used to develop models then transfer to SIMCA-Batch On-line for execution in real-time. In short, SIMCA-P+ can do:

- Batch model building (PCA or PLS)
- Control charts generation
- Contribution plots generation
- Various data filters
- Missing data treatment
- Report generation

*Application Areas:* In summary, Umetrics claims their software can be used in: Pharma R&D, Pharma-PAT, Semiconductor, Plastics, Chemicals, Pulp & Paper, Steel & Minerals, Food and Manufacturing. Pharma R&D focuses more on design of experiments while the others have close relationship to MSPC.

Company	Aspentech	Brooks Auto.	CAMO	Statsoft	Umetrics
Featured Software	Aspen Multivariate	FDC	Unscramber	Statistica	SIMPCA-P+
Specified Application area	No	Semiconductor	No	No	No
Operation Type	NA	Batch	Batch; Continuous	Batch; Continuous	Batch; Continuous
PCA	Yes	Yes	Yes	Yes	Yes
PLS	No	NA	Yes	Yes	Yes
Special Algorithms	Five PCA algorithm	NA	3-way PLS; K-means	ICA; Neural Network	Missing data; Various filters
Bioprocess Solution	Yes but not MSPC	No	No	Yes	Yes

Table 2.2: MSPC Software Comparison

### 2.4.2 Package Comparison

Table 2.2 compares several aspects of the software that are of interest. It can be seen that all software packages support PCA. Furthermore, Aspen Multivariate relies only on PCA. Also PLS is a widespread tool and most software package have that capability.

It seems that Aspen Multivariate is not very mature and has only an application in InfoPlus. Of course, one advantage is that the new software communicates easily with other Aspentech software and overall integrity can be excellent.

Among these software packages, Statistica shows powerful ability that includes many advanced algorithms and can be interfaced to other industrial software through their data server.

Umetrics has great potential in fault detection and diagnosis and it seems their products can be used for online process monitoring with their SIMCA-Batch On-Line or SIMCA-4000 package. Because monitoring is only part of the whole process control strategy, integrability with other control application softwares remains a doubt. It may not be a good idea to have more than one software packages to handle control and monitoring. In other words, if Umetrics combines their products together and make them easy to communicate with other control software (e.g., DeltaV), it may be a good alternative.

Brooks Automation is not analyzed because it mainly focuses on the semiconductor industry and their MSPC software is designed for their hardware. CAMO is similar to Umetrics and also has a trial version of their software.

Besides listed commercial oriented softwares, some large companies also have their own batch MSPC applications including Dow Chemical and AMD. Unfortunately we cannot get detailed information but they are all PCA, PLS based analysis with special data preprocessing rules for specific processes.

From above analysis, it can be seen that all the packages include PCA modeling and deployment features for continuous process monitoring. Continuous process is always regulated around a steady state and thus PV correlations remain the same once the system reaches a stable state. These features make PCA applications in continuous process relatively easy. In contrast, for batch processes, many of the packages are offline based and lack online calculation ability. Among those having online batch monitoring feature packages, there are few successful industrial stories. The challenges are two fold: 1. real time data management which allows data shared fluently between data collection server (e.g., OPC), data storage server, and algorithm part. 2. robust and reliable online batch monitoring algorithm that

can deal with real production challenges like batch nonlinearities, within batch PV correlation change at different batch phases, batch to batch deviations and so on. From a chemical engineering point of view, the main focus in this work is to develop a robust and reliable algorithm.

## 2.5 Batch Monitoring Procedure

In Section 2.3, some important definitions and algorithms for applying PCA to continuous processes are given. However, for batch and semi-continuous processes, there is no steady state and the historical trajectories usually contain considerable nonlinearity, so no effective online monitoring technique existed before 1990s. Nomikos and MacGregor [68, 70, 69] originally introduced the basic ideas of Multiway PCA and PLS methods for online batch monitoring. Because PCA and PLS are two dimensional decomposition tools, the three dimensional data collected from different batches are unfolded into two dimensional before further calculation. Another type of approach is based on parallel factor analysis (PARAFAC) or Tucker3 method [15, 82] in which the three dimensional data are analyzed directly without unfolding.

Multiway PCA is widely used in most known batch applications listed in Section 2.1 and thus it is chosen as the starting point of this work. In this section, the basic flow chart (Figure 2.1) and important steps in performing Multiway PCA is given. According to Figure 2.1, Multiway PCA contains offline model building and online deployment steps. For convenience, the five steps in model development part are named A1 to A5, and online deployment steps are called B1 to B5. NIPALS algorithm and principal component number choice ( $n_{pc}$ ) (A4) has been introduced in Section 2.2 and  $T^2$ , SPE control chart definitions (A5, B3 and B4) are given in Section 2.3. Thus, data synchronization (A2) and unfolding methods (A3) are

explained and challenges in the whole algorithm are discussed as well.

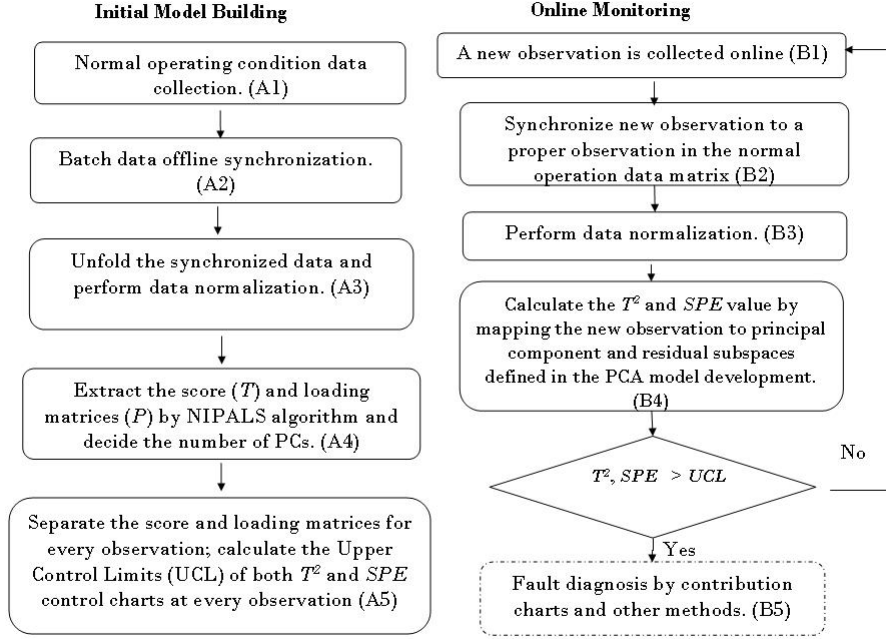


Figure 2.1: Flow chart of basic Multiway PCA

### 2.5.1 Batch Data Synchronization (A2, B2)

As pointed out earlier, the unequal length between batches and phases is a common feature for batch process (Figure 2.2). If PCA is used directly without synchronization, the results can be misleading. In other words, synchronization between batches is a necessary assumption to perform monitoring [70, 98]. However, relatively little attention has been paid to this area.

Besides traditional data interpolation, Kassidas et al. [46] migrated dynamic time warping (DTW) method from speech recognition and applied it to polymerization process data synchronization. Recently, a modified derivative-based DTW

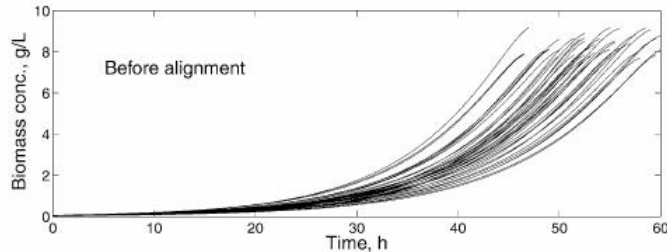


Figure 2.2: PV trajectory from different batches

algorithm is applied to time series data analysis [47]. Cherry [20] compared data interpolation and DTW in the application of etching process. According to his research, interpolation was good enough for many cases and DTW was suggested for poorly aligned cases. At the mean time, Cherry admitted data synchronization was far from a closed topic and needed new algorithms and developments. Moreover, most chemical and biochemical batches are not aligned well naturally compared with semiconductor processes. Thus, it is important to evaluate available results and propose new solutions. A detailed discussion is postponed until Chapter 3.

### 2.5.2 Three-dimensional data unfolding (A3) [98]

In this section, all the trajectories are assumed to have an equal length as a result of synchronization (A2). Different from two dimensional data in continuous process ( $X$  matrix in Section 2.3), historical data of batch are usually stored in an  $I \times J \times K$  matrix ( $\bar{X}$ ), where  $I$  is the number of batches,  $J$  is the variable number, and  $K$  is the sampling times (observations). A characteristic batch profile is shown in Figure 2.3 where large nonlinearities present and multiple phases exist. Recently, Kourti [48] has pointed out practical concerns that the measurements may not be equally

spaced and that some variables do not exist throughout the whole processes. These situations are variations of the general case ( $K$  is the same for each batch), but in this chapter  $J$  and  $K$  are assumed constants.

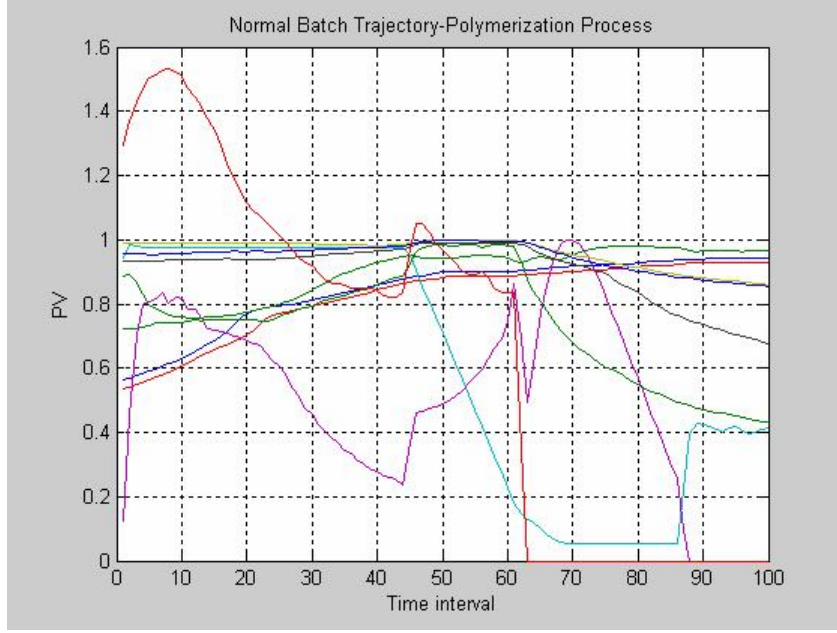


Figure 2.3: PV profile of a polymerization process

Sprang et al. [83] provided a critical evaluation for online batch processing and discussed different data unfolding methods. In summary, there are three ways to unfold the data: batch-wise, variable-wise, and time-wise (Figures 2.4 to 2.6). Besides Sprang et al., Lee et al. [54], Westerhuis et al. [89], and Kourti [48] have also compared the three different unfolding methods.

The way by which unfolding method uses leads directly to the variation information that PCA extract. Batch-wise unfolding focuses on analyzing the differences among batches, variable-wise unfolding attempts to discover the variability between variables, and time-wise unfolding is used to extract the correlation among observa-

tions at different times. The first two methods are widely used in batch monitoring, and we will discuss them in detail here.

After unfolding, the mean-centering and rescaling steps are carried out as described in Section 2.2 (preprocessing). Centering is performed so the PCA model represents the deviation from the mean, and the scaling step is necessary to give each process variable an equal weight prior further analysis. If different weights are needed for some variables, then a diagonal weighting function multiplies the raw data after mean centering [70].

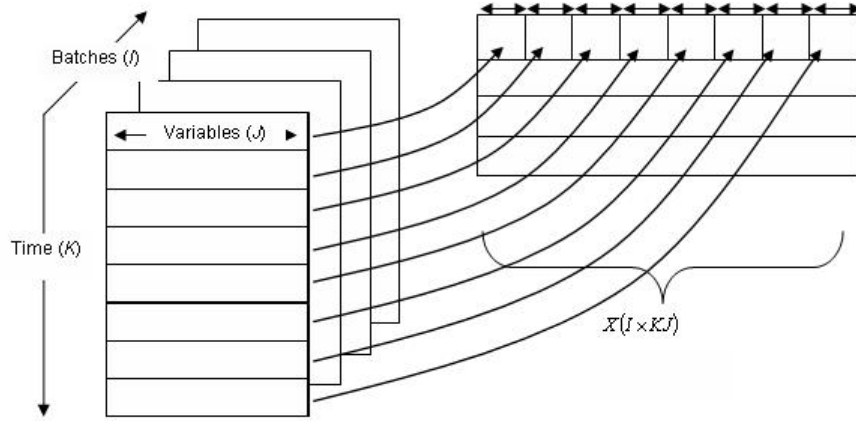


Figure 2.4: Unfolding data matrix according to batch-wise  $\bar{X} \rightarrow X(I \times JK)$

When batch-wise unfolding is used, the mean trajectory reflects the nonlinearities of the system. After preprocessing, the residual is the deviation of the specific trajectory from the average profile of the process. The nonlinear behavior of the batch is removed in this way, which is a big advantage, but the row vector data will not be complete until the end of a batch (Figure 2.4). In PCA online monitoring, the score and loading calculations need a complete data set. Thus, one has to predict the future values for the whole batch [70, 69] in online application, which is time



consuming and can add uncertainty, especially during a batch's initial period.

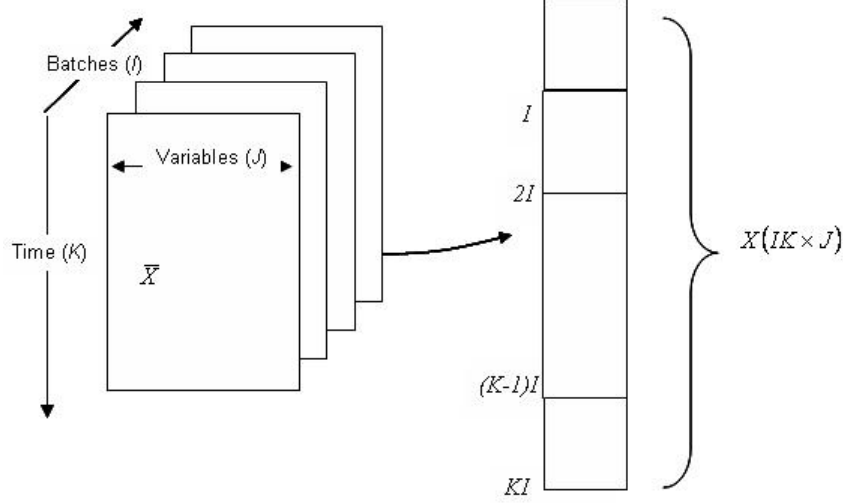


Figure 2.5: Unfolding data matrix according to variable-wise  $\bar{X} \rightarrow X(IK \times J)$

The variable-wise method discovered by Wold et al. [94] does not need future data prediction because only the current time data matrix ( $I \times K$ ) is needed for online applications (a small data matrix shown in Figure 2.5). The shortcoming of the variable-wise approach is that the system dynamics are remained in the data set after preprocessing. Thus, the calculated loadings will contain the correlations between variables, with large nonlinearity included [48, 54]. Furthermore, Westerhuis et al. [90] found that variable-wise analysis offers little benefit to monitoring, since it focuses on the wrong source of variations in the data.

Recently, Lee et al. [54] combined batch-wise and variable-wise methods, named *hybrid-wise unfolding*, and applied it successfully to a bioreactor fault detection case. Then, the data set is unfolded batch-wise, and the mean-centering and scaling steps are performed. After that, the data are rearranged to the variable-wise structure. In this way, future data prediction is avoided and data nonlinearity is removed at

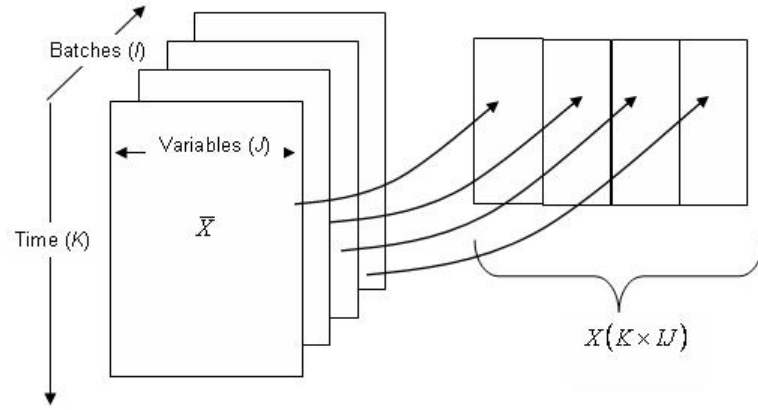


Figure 2.6: Unfolding data matrix according to time-wise  $\bar{X} \rightarrow X(K \times IJ)$

the same time.

Hybrid-wise unfolding Multiway PCA (HMPCA) is proved to be more sensitive to small process faults than other unfolding approaches. However, all Multiway PCA based algorithm neglected the dynamics (time dependent correlations) within the batch. Thus, HMPCA based algorithm is chosen as the starting point for further algorithm development study explained in Chapter 4.

Up to now, all the blocks in Figure 2.1 has been more or less covered. Considering the features of batch process, algorithm developments focus on data synchronization (Chapter 3) and batch dynamics treatment (Chapter 4).

## Chapter 3

# Batch Profile Synchronization

In Section 2.5, the importance of data preprocessing for batch monitoring is presented. In short, data synchronization is a foundation assumption for statistical modeling approaches. By far, data interpolation [20], indicator variable [68, 50], Dynamic Time Warping (DTW) [46], and Correlation Optimized Warping (COW)[67] are the best known methods. Important features of each method are listed in Table 3.1 and analyzed below.

Methods	Online Implementable	Guaranteed Synchronization	Computation Load
Interpolation	Synchronize every observation	No	Low
Indicator Variable	Synchronize every observation	No	Low
DTW	Synchronize every observation	Yes	High
COW	Cannot synchronize until a block of observation is available	Yes	High

Table 3.1: Synchronization Methods Comparison

In some cases the Indicator Variable approach does not work well [48] and simple Interpolation within the data can be employed but it cannot guarantee synchronization (the batches may still reach completion at different times). DTW and COW

are more complicated but they capture the features, match patterns for batches of different length and have broad applications. COW was proposed in 1998 to correct chromatograms for shifts in the time axis prior to multivariate modeling [67]. DTW was first used in speech recognition and introduced for batch data analysis by Kasidas et al. in 1998 [46]. Both COW and DTW translate the alignment problem into a dynamic optimization problem. The main difference is that COW is based on segment-wise data correlation analysis and DTW relies on a point to point Euclidean distance calculation. Recently, both methods are compared as preprocessing methods for chromatographic data analysis [72, 85]. While nearly identical results were suggested by both methods, COW was slightly preferred because a segment of data was considered together rather than one point at a time in DTW, such that the number of possible paths is reduced. Furthermore, DTW may need rigid constraints and sometimes raw distance calculations are not good enough [85]. However, for *on-line* process monitoring purpose, ideally data synchronization should be carried out in real time when a new observation is available, but COW needs to wait until a block of data is collected. As a result, COW is suitable for low frequency applications and DTW is preferred and selected as the starting point for batch monitoring applications.

In Section 3.1, DTW and Derivative DTW (DDTW) are reviewed. Next, a new Robust DDTW (RDDTW) algorithm is proposed and Section 3.3 compares different synchronization methods by using various datasets. A concise summary is given in Section 3.4.

### 3.1 Dynamic Time Warping and its derivative

Time series data are a commonly occurrence in physical, social, and economic systems. In order to compare one sequence with another, it is always desirable to

align the features (peak, valley, etc.). Dynamic Time Warping (DTW) can capture the dynamics and match patterns for different time series. After it was first used in speech recognition, the DTW method was successfully applied to industrial emulsion polymerization process monitoring [46], spectroscopic profile analysis [72], semiconductor production monitoring [20], and chromatographic data alignment [85]. Below, the key algorithm and important features for DTW and its variation are introduced.

### 3.1.1 Dynamic Time Warping Algorithm

Assume  $x_{ref}$  is a reference trajectory (previously defined) and  $x_{new}$  is a new trajectory to be synchronized. The first step of DTW is to define the Euclidean distance between each point of the two trajectories as:

$$d(i(\kappa), j(\kappa)) = (x_{ref}[i(\kappa), :] - x_{new}[j(\kappa), :]) \times W \times (x_{ref}[i(\kappa), :] - x_{new}[j(\kappa), :])^T \quad (3.1)$$

$d$  is the local distance,  $W$  is a positive definite weighting matrix that reflects the repeatability of each process variables for batch synchronization.  $\kappa$  is the number of grid points along the path (in Figure 3.1,  $\kappa = 1, 2, \dots, 7$ ). The lengths of  $x_{ref}$  and  $x_{new}$  are  $t$  and  $r$ , respectively. Then the total distance between the two trajectories are:

$$D(t, r) = \frac{\sum_{\kappa=1}^K d(i(\kappa), j(\kappa))}{N(w)} \quad (3.2)$$

where  $N(w)$  is a normalization factor. There are many possible point assignments and the goal is to find the optimal path  $\eta$  that minimizes  $D(t, r)$ , which can be denoted as (see the blue line in Figure 3.1):

$$\eta = \{c(1), c(2), \dots, c(K)\} \quad (3.3)$$

$c(\kappa)$  represents a grid point in Figure 3.1, where  $\kappa$  can be treated as another index used to connect reference ( $i$ ) and new trajectories ( $j$ ), e.g.  $c(1) = [1, 1]$ ,  $c(2) = [1, 2]$ , ...,  $c(7) = [5, 6]$  in Figure 3.1.  $K$  is the total number of points needed for the path ( $K = 7$  in Figure 3.1).

Before minimizing Eq. 3.2, there are some global and local constraints that should be addressed.

*Global constraints:* The two ends of both trajectories ( $c(1)$  and  $c(K)$ ) should be aligned together:

$$c(1) = [1, 1]$$

$$c(K) = [t, r]$$

where

$$c(\kappa) = [i(\kappa), j(\kappa)]$$

The constraint on aligned path lengths for  $x_{ref}$  and  $x_{new}$  is:

$$\max(t, r) \leq K \leq t + r$$

*Local constraints:* Local constraints are used to make the path continuous and monotonic. If  $(i, j)$  is the  $\kappa^{th}$  point on  $\eta(c(\kappa))$ , then the predecessor point on path  $\eta(c(\kappa - 1))$  can be any one of the three points (the three directions in Figure 3.1):

$$c(\kappa - 1) = (i - 1, j), (i - 1, j - 1), \text{ or } (i, j - 1) \quad (3.4)$$

In Figure 3.1 and all surface plots of warping path in this chapter, the darker one block is, the larger feature differences (raw distance for DTW) there is between the

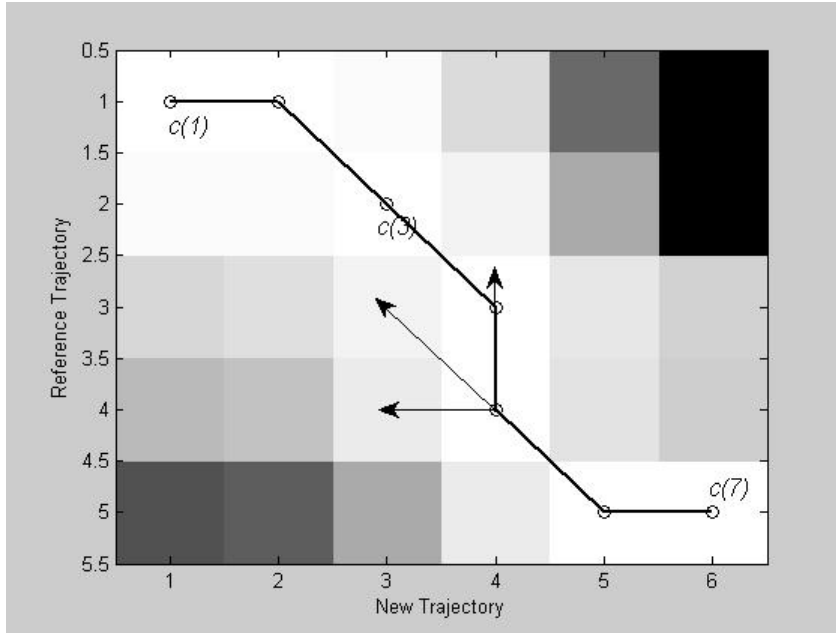


Figure 3.1: Warping path by DTW

corresponding observations in the reference and new trajectories. After minimizing Eq. 3.2 subject to the local and global constraints by dynamic programming, the optimal warping path can be found. Detailed algorithm descriptions can be found in [47, 46, 77].

### 3.1.2 Derivative Dynamic Time Warping Algorithm

When a feature (peak, valley, etc.) is slightly higher (or lower) from one trajectory to another, DTW may suggest aligning one single point on  $x_{ref}$  to a large number of points on  $x_{new}$  or the opposite way, as shown in Figure 3.2 (a). We call this undesirable behavior a *singularity*. To overcome these problems, Keogh and Pazzani [47] used the point derivative instead of Euclidian distance to measure the difference

between two trajectories:

$$d(i(\kappa), j(\kappa)) = (dx_{ref}[i(\kappa), :] - dx_{new}[j(\kappa), :]) \times W \times (dx_{ref}[i(\kappa), :] - dx_{new}[j(\kappa), :])^T \quad (3.5)$$

where  $dx_i = \frac{x_i - x_{i-1}}{2}$ . DDTW can be carried out by replacing Eq. 3.1 by Eq. 3.5 and then carrying out the minimization of Eq. 3.2. Instead of measuring Euclidean distance, DDTW considers the estimated local derivatives. According to Keogh and Pazzani [47], the time requirement for DDTW is  $O(tr)$ , which is the same as DTW, thus both methods take approximately the same time for computation. Detailed computation time comparison is made in Section 3.3.

Figures 3.2(a) and (b) compare the synchronization results of DTW and DDTW. The new trajectory follows  $x = \cos^2(t) + \sin(t)$  (in red) and the valley between time points 6 to 12 in the reference trajectory is moved downward by 2 (in blue). According to DTW (Figure 3.2(a)), point 8 in the new trajectory is aligned to five points (7 to 11) in the reference trajectory. This result is consistent with the shortcomings of DTW described earlier. After replacing point distance by point derivative, DDTW results are shown in Figure 3.2(b). It can be seen that point 8 in the new trajectory is only aligned to point 8 in the reference. Comparing Figures 3.2(a) and (b), one can see that the new trajectory is aligned much better by DDTW than DTW. Figure 3.3 shows the alignment path by DDTW. As expected, a perfect alignment between these two trajectories should be a diagonal line and Figure 3.3 is very close to this.

From this noise-free case, we can see DDTW seems to correct the *singularity point* problem with DTW. However, DDTW may not be as robust as DTW as a result of numerical derivative estimation. When left point estimation ( $dx = \frac{x_i - x_{i-1}}{2}$ ) is used to estimate point derivatives of the process:  $x(t) = \sin(t) + N(0, 0.04)$ , the quality



of estimation becomes unacceptable (Figure 3.4). Thus, a robust derivative-based time warping algorithm is desirable for industrial applications.

## 3.2 Robust Dynamic Time Warping

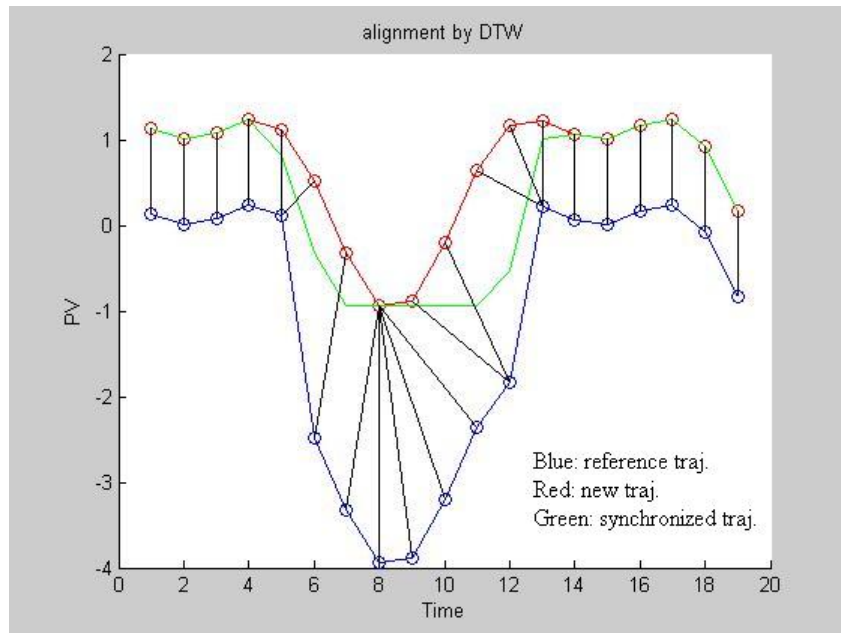
In 1964, Savitzky and Golay [79] combined the idea of least squares and moving window together, which can be used to smooth raw data and predict derivatives simultaneously. In 1972, errors contained in [79] were corrected by Steinier et al. [84]. There are more than 1000 citations according to *web of science* and most of the applications are in signal processing and analytical chemistry data analysis. The main advantage of this approach is that it tends to preserve features of the distribution such as relative maxim, minim and width, which are usually 'flattened' by other adjacent averaging techniques (like moving averages). However, most applications are offline data processing and real-time synchronization application are unavailable to our best knowledge. In this Section, the SG filter algorithm is discussed then RDDTW is proposed by combining DDTW with SG filter.

### 3.2.1 SG Filter

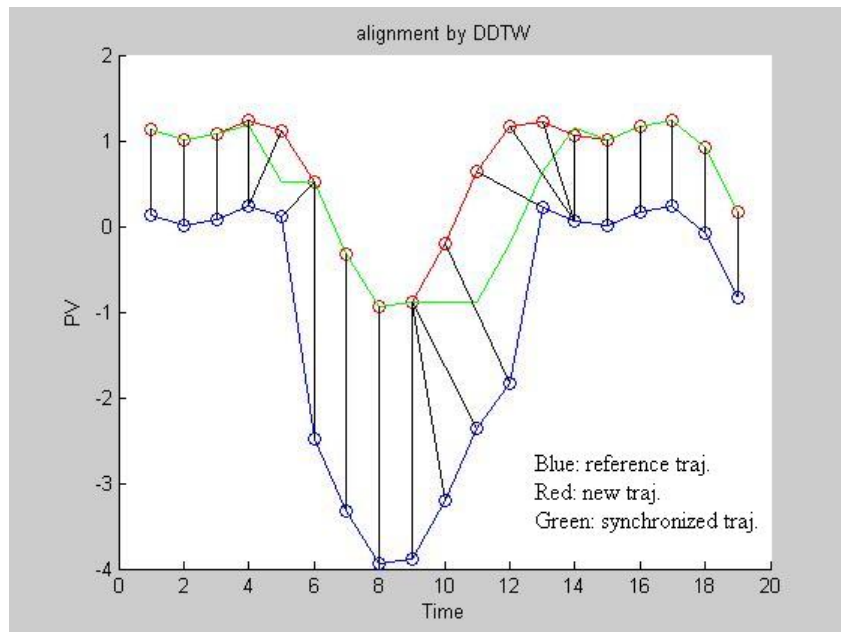
The smoothing problem with a moving window can be reformulated as follows:  $2h + 1$  equally separated points are to be fit into an  $m^{th}$  order polynomial ( $m < 2h + 1$ ):

$$f_i = \sum_{k=0}^m a_{mk} i^k; i = -h, -h + 1, \dots, 0, \dots, h - 1, h \quad (3.6)$$

From the  $2h + 1$  raw measurements, we are interested in estimating the noise free value of the middle point ( $i = 0$ ) and its derivatives. There are methods to estimate the left ( $i = -h$ ) and right ( $i = h$ ) point values but this will sacrifice estimation



(a) DTW alignment



(b) DDTW alignment

Figure 3.2: Alignment of two trajectories.

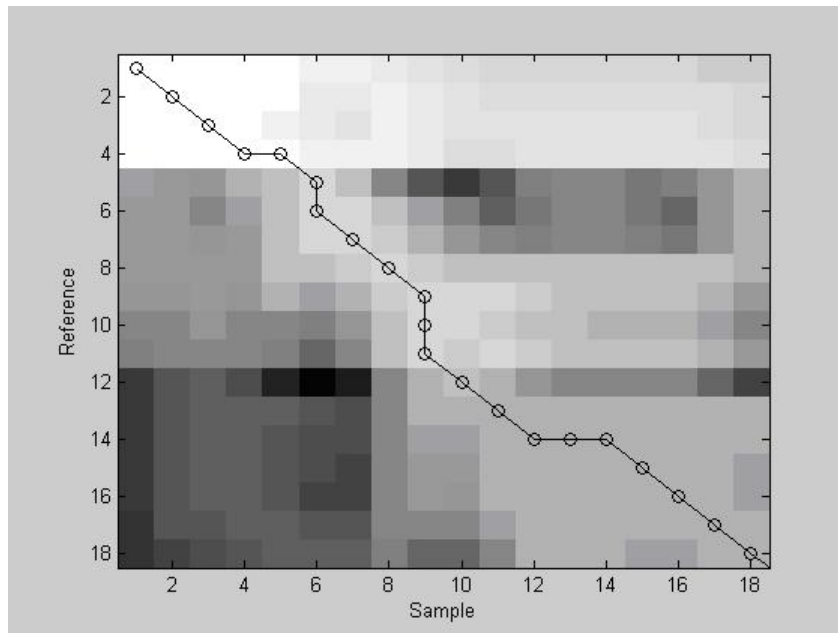


Figure 3.3: DDTW alignment path of two trajectories.

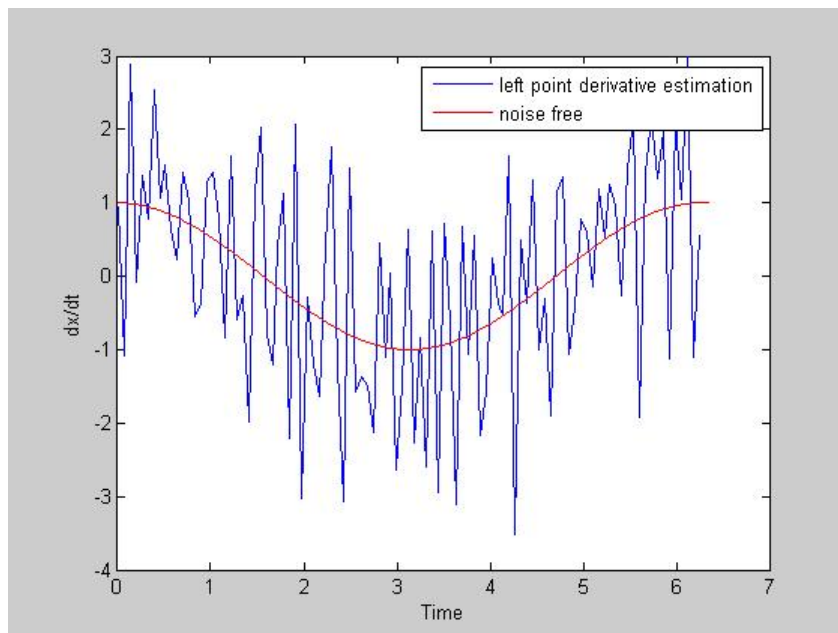


Figure 3.4: Derivative estimation of a noise corrupted sinusoid.

accuracy. The  $j^{th}$  order derivatives of Eq. 3.6 at  $i = 0$  can be expressed as:

$$\left(\frac{d^j f_i}{di^j}\right)_{i=0} = j!a_{mj} \quad (3.7)$$

Suppose the noise free value of the  $2h + 1$  points are  $X = [x_{-h}, x_{-h+1}, \dots, x_h]$  such that the objective function of smoothing is:

$$J = \min \{(X - F)^T \times (X - F)\} \quad (3.8)$$

The unknowns in Eq. 3.8 are the polynomial coefficients and in order to find the best fit, the first derivative should be zero (Note: there is no constraint on polynomial coefficients):

$$\frac{\partial J}{\partial a_{m\gamma}} = \sum_{i=-h}^h \left[ \left( \sum_{k=0}^m a_{mk} \times i^k \right) - y_i \right] \times i^\gamma + a_{m0} - x_0 = 0 \quad (3.9)$$

By solving Eq. 3.9, we have:

$$\sum_{k=0}^m a_{mk} S_{\gamma+k} = F_k \quad (3.10)$$

where

$$S_{\gamma+k} = \sum_{i=-h}^h i^{\gamma+k}$$

$$F_k = \sum_{i=-h}^h i^k \times x_i$$

and  $\gamma$  is the equation number that runs from 0 to  $m$ . Eq. 3.10 can be used to determine the value of polynomial coefficients  $a_{mk}$  derived in [79, 84]. If the polynomial order or the number of observations used in the estimation is so large that it is not listed in [79, 84], Eq. 3.10 can be used as follows. The first step calculates  $S_{r+k}$  and substitutes it into Eq. 3.10. Together with  $F_k$ , all parameters ( $a_{mk}$ ) can be

expressed by a weighting matrix in terms of observations.

As an example, seven observations are used to determine the first derivative by fitting a quadratic polynomial with following parameter settings,  $m = 2$ ;  $h = (7 - 1)/2 = 3$  and  $k = 1$ :

$$dx_0 = a_{21} = \frac{-3x_{-3} - 2x_{-2} - 1x_{-1} + 0x_0 + 1x_1 + 2x_2 + 3x_3}{28}$$

Taking Eq. 3.7 into account,  $j = 1$  so that  $j! = 1$ , thus  $a_{21}$  is the robust quadratic derivative estimation function parameter with seven observations.

By combining the Savitzky-Golay (SG) filter with DDTW, a robust DDTW algorithm can be proposed. The noisy sinusoid process described in Section 3.1.2 (Figure 3.4) is tested by the SG filter method. A piecewise average (every ten observations) is taken first as the process dynamics change much slower than the sampling rate. For comparison purpose, widely used exponential filter is used as the preprocessing step for left point derivative estimation:

$$x_{filter}(i) = x(i) \times a + x_{filter}(i - 1) \times (1 - a), a = 0.5$$

As shown in Figure 3.5(a), both SG and exponential filter showed an excellent *state tracking* ability with negligible noise level. However, for *derivative estimation* (Figure 3.5(b), SG filter showed significant advantages over exponential filter based left point method. Though the noise level for raw measurements is reduced by using exponential filter, left point estimation magnifies the remaining noise significantly. SG filter fits blocked wise data into polynomials then use polynomial to calculate derivative which guarantees a better precision. Left point estimation can be treated as a special case of SG filter with  $m = 1$ , and  $h = 2$ .

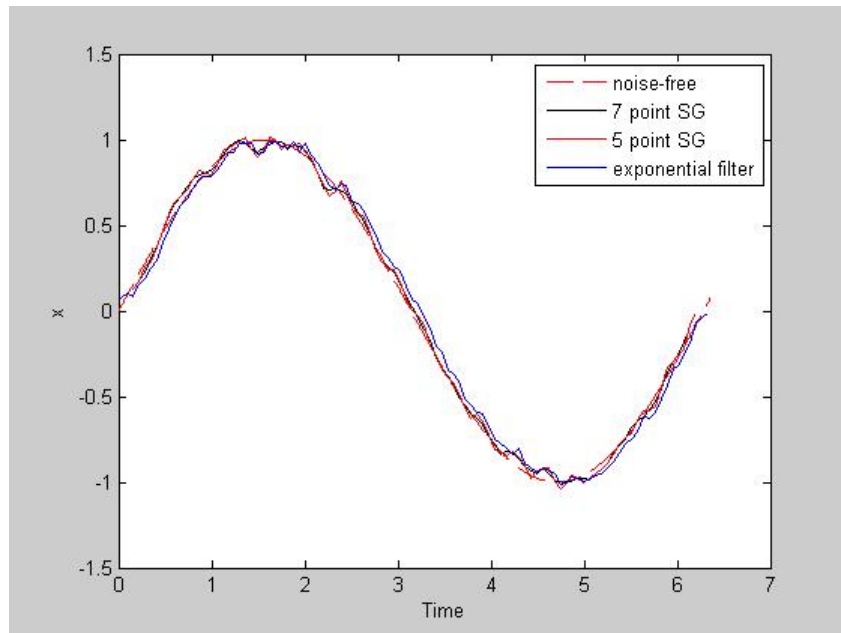
Two parameters are required for SG filter: polynomial order ( $m$ ) and number of observations in a block ( $2h + 1$ ). In this problem (Figure 3.5(b)), both five point and seven point second order polynomial SG filter give satisfactory results, thus the SG filter is not sensitive on the two filter parameters. Filter parameter selection is further discussed in Section 3.3.

If the process noise does not follow normal distribution, Eq. 3.8 should be reformulated into a Maximum Likelihood Estimation rather than a Least Squares problem. In that case, the solution will depend on noise probability density function ( $pdf$ ) and the solution may not be trivial. However, normal distributed noise is a commonly used assumption in both industry and academia.

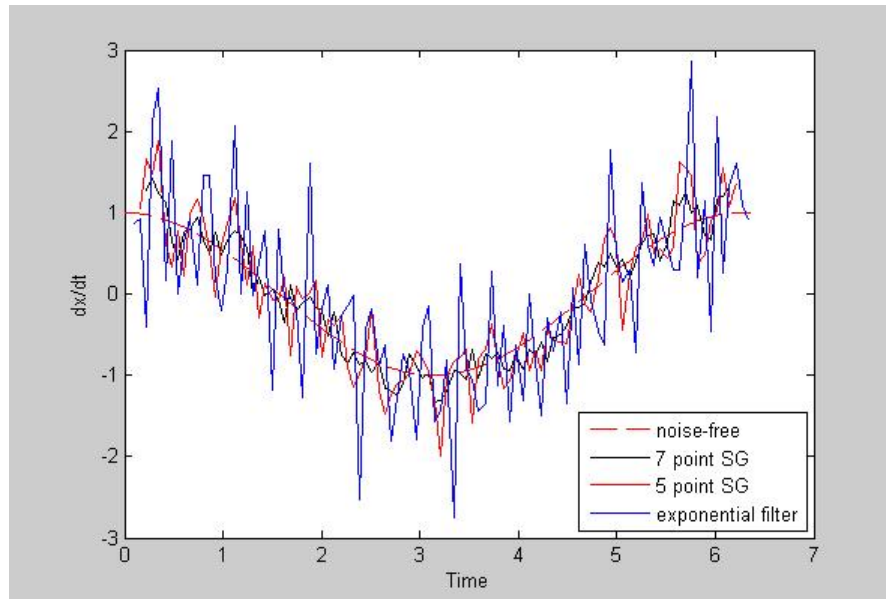
### 3.2.2 Robust DDTW Algorithm

RDDTW can be treated as a combination of DDTW and robust numerical derivative estimation. Generally speaking, RDDTW algorithm can be summarized as:

1. Evaluate the raw trajectory. If the raw trajectory has a high frequency noise level, it is necessary to use piecewise averaging before performing further calculations.
2. Use SG filter to estimate numerical derivatives. In the above example, only a quadratic polynomial is used. In real applications, one can choose higher order polynomial together with a larger number of observations. Usually the number of observations should be at least twice as many as the order of the polynomial in order to get good results [79]. The number of observations indicates the rate of change in the dynamics. For example, a second order polynomial is chosen and if four observations are used to fit the polynomial, it indicates the four observations follow the polynomial. If ten observations are used to fit, then it indicates the ten points follow the trend of a second order polynomial.



(a) original state estimation



(b) derivative estimation

Figure 3.5: SG filter and exponential filter comparison.

Polynomial order hardly goes beyond fifth for chemical processes due to the relatively slow dynamics.

3. Use the estimated derivative and the algorithm described in Sections 3.1 to perform the alignment robustly.

Above steps are suitable for all offline signal processing applications. For batch online synchronization, once an observation ( $i$ ) is collected, numerical derivative estimation for the  $i - h$  point will be carried out ( $2h + 1$  is the number of points used to fit an  $m^{th}$  order polynomial), as discussed in Section 3.2.3.

### 3.2.3 Computational Procedure

In Figure 2.1, the process monitoring applications are divided into *offline model building* and *online monitoring*. This chapter mainly deals with the *Data Synchronization* block in both cases.

*PCA model development:*

- A Scaling: There is a serial of good batches:  $(X1)_{raw}, (X2)_{raw}, \dots, (XI)_{raw}$  and each batch has  $J$  variables. For each variable, calculate its range in every batch and then take the average value. Finally, divide each variable in all batches with the average range of the variable. This step ensures every variable has the same magnitude

$$Xi(j, :) = \frac{Xi(j, :)_{raw} \times i}{\sum_{i=1}^I range(Xi(j, :)_{raw})}$$

- B Synchronization: There are  $I$  good batch data sets and the one  $(Xi)$  that is closest to the average length of all trajectories is selected as the reference batch. The weighting matrix  $W$  is set equal to an identity matrix at the beginning of the iteration.



*Iterative process:*

- a Apply the SG filter to estimate local derivatives and calculate the optimal synchronization path by RDDTW algorithm described in Section 3.2.1 and 3.2.2.
- b Apply backward tracking as described in [46].
- c Asymmetric synchronization: After backward tracking, the optimal path will have a different length for both reference and sample. In order to make the synchronized path length equal to reference, the value of consecutive horizontal steps (Figure 3.3) are averaged and aligned with related reference points. In this way, the sample trajectory will have the same data length as the reference.
- d The synchronized batch profile ( $X_i$ ) have the same length now, so define the average trajectory as:

$$\bar{X} = \sum_{i=1}^I \frac{X_i}{I}$$

- e Define the weighting matrix  $W$  as follows:

$$W(j, j) = \left( \sum_{i=1}^I \sum_{k=1}^K [X_i(k, j) - (\bar{X})(k, j)]^2 \right)^{-1}$$

Normalization:

$$W = \frac{W \times J}{\sum_{j=1}^J W(j, j)}$$

Note,  $W$  is a diagonal matrix.

- f Go back to step *a* until  $W$  converges. When  $W$  converges, store  $W$  and average trajectory ( $\bar{X}$ ) for future use.

C Offline PCA model building. This belongs to the next block in Figure 2.1.

*PCA Online Monitoring:*

- A Scaling: Divide each variable with the corresponding average range in PCA building step.
- B Synchronization: Set the average trajectory of the good batches as the reference and perform synchronization (same as item  $c$ , asymmetric synchronization, in model building step) using the weighting matrix ( $W$ ) that was stored from PCA development.
- C Online PCA calculation which belongs to the next block in Figure 2.1.

### 3.3 Case Study

Three examples are used to cover lab experimental data, dynamic simulation data, and industrial manufacturing data. The purpose is to test the robustness of the algorithm under different circumstances (signal/noise ratio, sampling frequency, process dynamic frequency, etc.), and also algorithm performance with different parameter settings.

#### 3.3.1 Industrial NIR Data

In 2002, the International Diffuse Reflectance Conference (IDRC) published a shootout data set including spectra from 654 pharmaceutical tablets of two spectrometers. Here, two spectra were selected to test our algorithm (Figure 3.6). One can see the raw trajectories are fairly well-synchronized and only a minor alignment is needed. The main differences between  $x_{ref}$  and  $x_{new}$  are the magnitudes of the trajectory features.

Figure 3.7 compares the alignment results. One can see that DTW shifted the raw trajectory between wavelength 30 to 80, which is unnecessary. Furthermore,

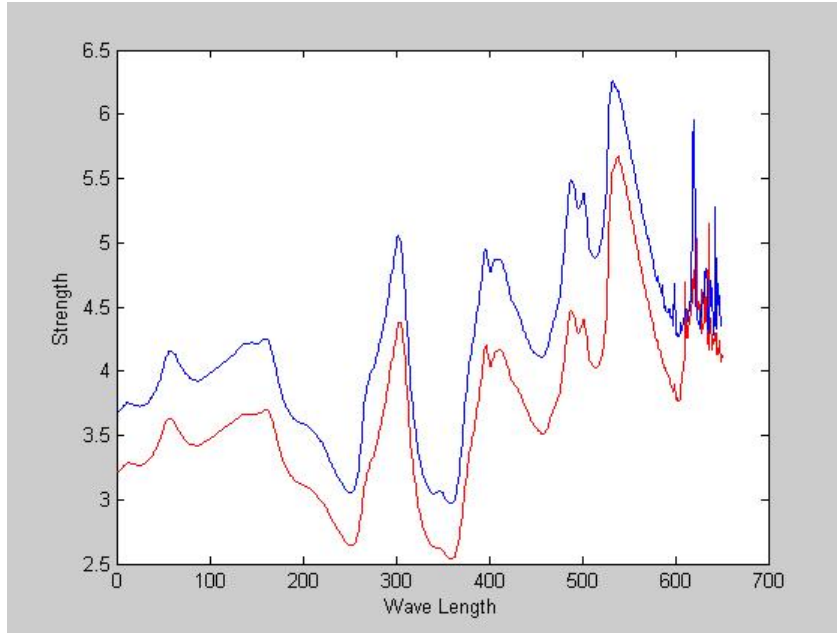


Figure 3.6: Raw NIR trajectory from IDRC dataset. Blue: reference trajectory. Red: trajectory to be synchronized.

flat periods (*singularity points*) appear at nearly every peak. For this low noise level case, DDTW and RDDTW give similar results and only minor differences exist around wavelength 550, which indicates RDDTW and DDTW are equivalent in this case. Thus, we can conclude that DTW causes *singularity points* due to feature differences, and for low noise level cases, both DDTW and RDDTW give satisfactory results that avoid *singularity points*.

### 3.3.2 Dynamic Trajectory Synchronization

Section 3.3.1 gives an offline analytical data synchronization problem and this section focuses on a dynamic trajectory synchronization problem. Figure 3.8 shows a process developed by SIMULINK. The system input is composed of a pulse sequence and a random number generator. After a one second transport delay, the input signal passes through a second order and a first order system, sequentially.

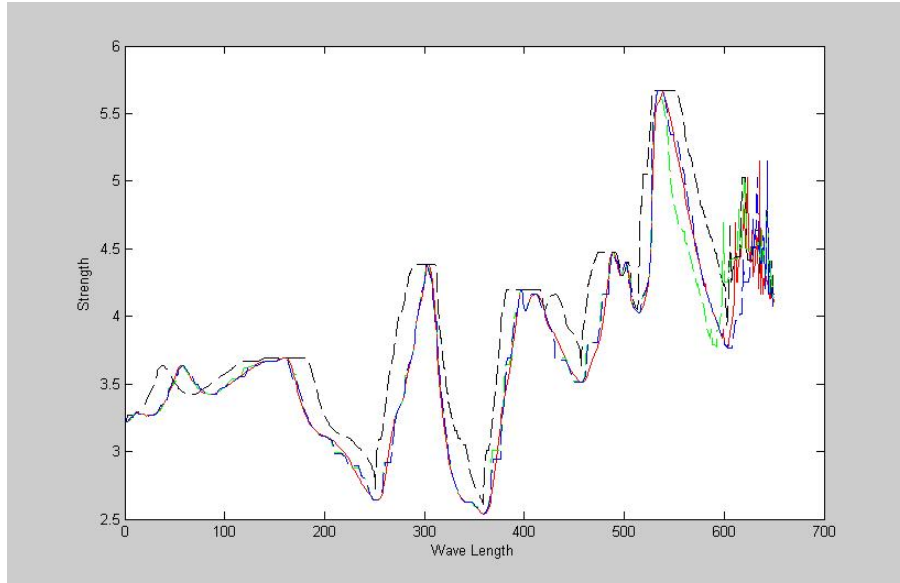


Figure 3.7: Synchronized NIR trajectory. Black dash: DTW. Green dash: DDTW. Blue dash: RDDTW

Finally, the output from first order system contains random noise. Reference and new trajectories are generated with different pulse magnitudes (0.7 for reference and 0.5 for new trajectory, respectively). Normal distributed random numbers were added to the input ( $N(0, 0.05)$ ) and output ( $N(0, 0.01)$ ) variables, respectively.

The simulated trajectories are shown in Figure 3.9 with a sampling rate of seconds. Both trajectories behave similarly but the new trajectory (in blue) is neither synchronized nor has the same magnitude as the reference (in red). It can be seen that the noise frequency is not very high and the piecewise average is taken for every two observations. For all three methods, the local constraint is set to 40. For RDDTW, seven points with a 3<sup>rd</sup> order polynomial is used.

Figure 3.10 shows the alignment paths for each method. It is clear that Figure 3.10(a) has many perpendicular and horizontal moves indicating *singularity points*.

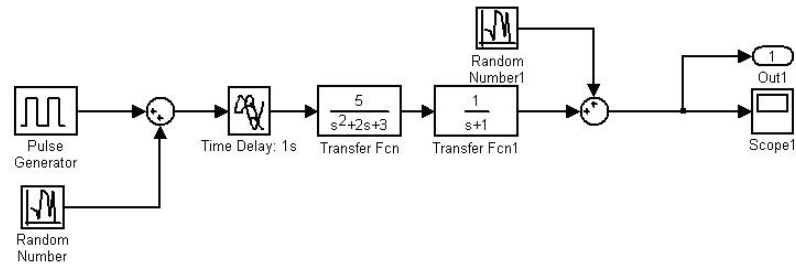


Figure 3.8: Dynamic simulation example.

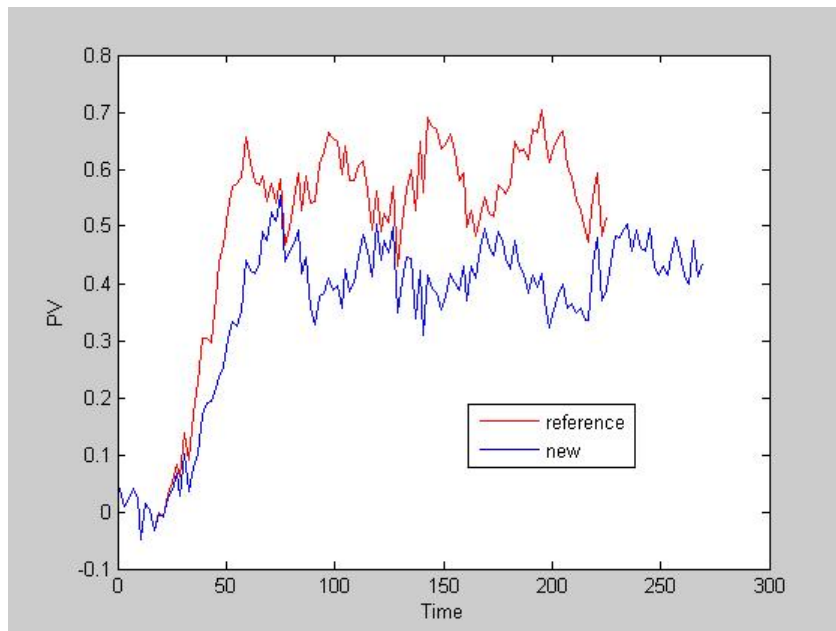


Figure 3.9: Reference and new trajectories

Furthermore, the alignment path reaches the boundary at times which means synchronization failure. In comparison, DDTW and RDDTW have fewer *singularity points* and do not reach the boundary set by bandwidth constraints. After a closer look at Figures 3.10(b) and (c), DDTW and RDDTW give different alignment paths. DDTW follows a straight line for the first 80 time periods with a ratio equal to one. This means few data folding or extracting happens during this phase. From period 80 to approximately 100, there is a significant horizontal dominated line. In comparison, RDDTW roughly go diagonal all through the batch which justifies no *singularity point* and the new trajectory is homogeneously shortened to match the pattern of the reference which is in accordance with intuition.

Aligned trajectories are shown in Figure 3.11. DTW (in blue) shows many flat periods caused by singularity points. Figure 3.11(a) compared the reference (in red) trajectory with both DDTW (in green) and RDDTW (in black) results, neither method has many singularity points. DDTW results still have a significant phase delay for the first 85 time period compared with reference trajectory which indicates failure. On the contrary, RDDTW shows well-synchronized results all along the batch. In Figure 3.11(b), warped trajectories are compared with original one. DTW distorts the original one as shown by the *singularity points*. As discussed above, DDTW nearly coincides with the original trajectory for the first 85 time periods. At the mean time, RDDTW keeps the important features of the original data (peak height, valley depth, etc) and synchronized well with the reference trajectory throughout the whole batch.

From above analysis, one can see that DTW fails due to *singularity points*; DDTW overcomes this difficulty but the noise level leads to unacceptable alignment results; RDDTW with seven points and  $3^{rd}$  order polynomial works best.

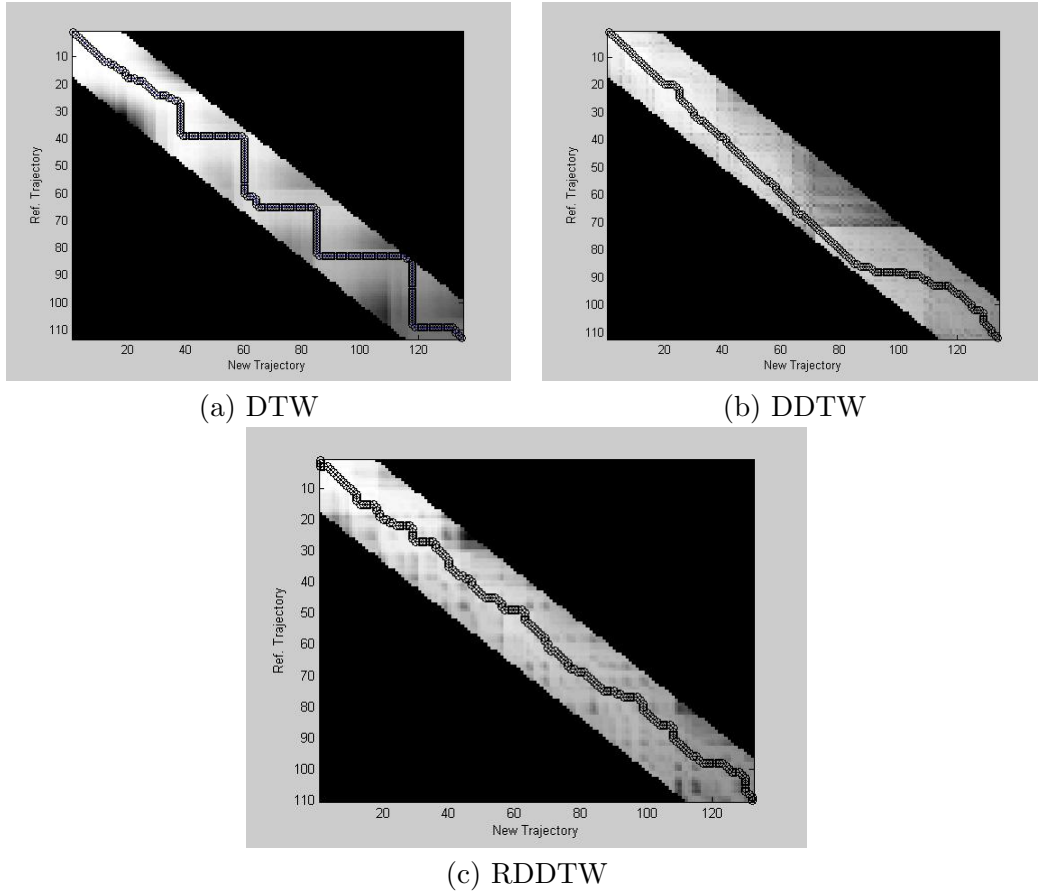
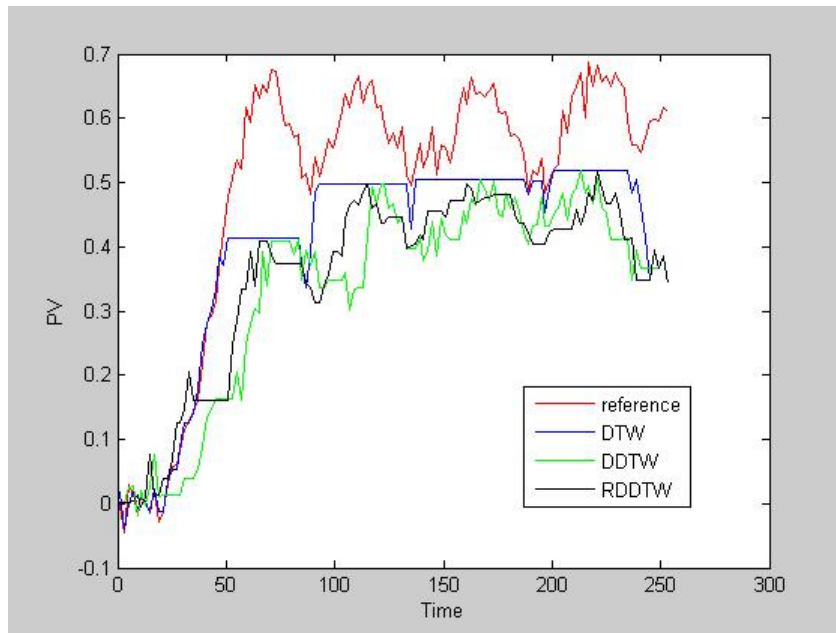
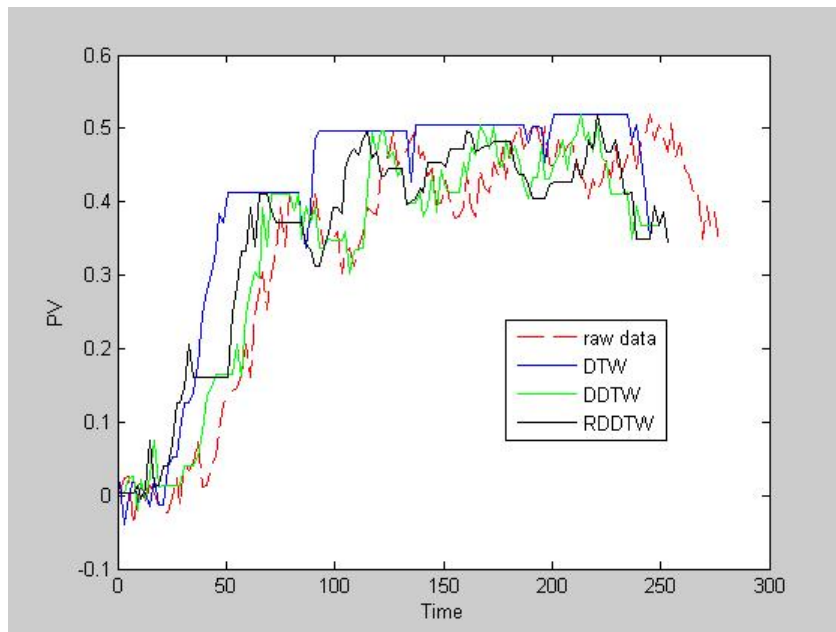


Figure 3.10: Alignment path for different methods.

The next question is how filter parameters affect the performance of RDDTW. Due to the slow dynamics of most chemical processes, high order polynomial for a small number of observations is unnecessary. Here, polynomial order is set to  $h$  and the number of points used to fit the polynomial is  $2h + 1$ . Thus, the only unknown parameter left is  $h$  and testing results are shown in Figure 3.12. One can see that, for  $h = 3, 4$ , and  $5$ , the synchronization results are all acceptable despite the flat period for  $h = 4$  around  $t = 50$ . The resulting trajectories are well-aligned with the reference and most features in original trajectories are retained as well. Thus, RDDTW method is robust to both process noise and filter parameter setting.



(a) Reference



(b) Original

Figure 3.11: Alignment results comparison.



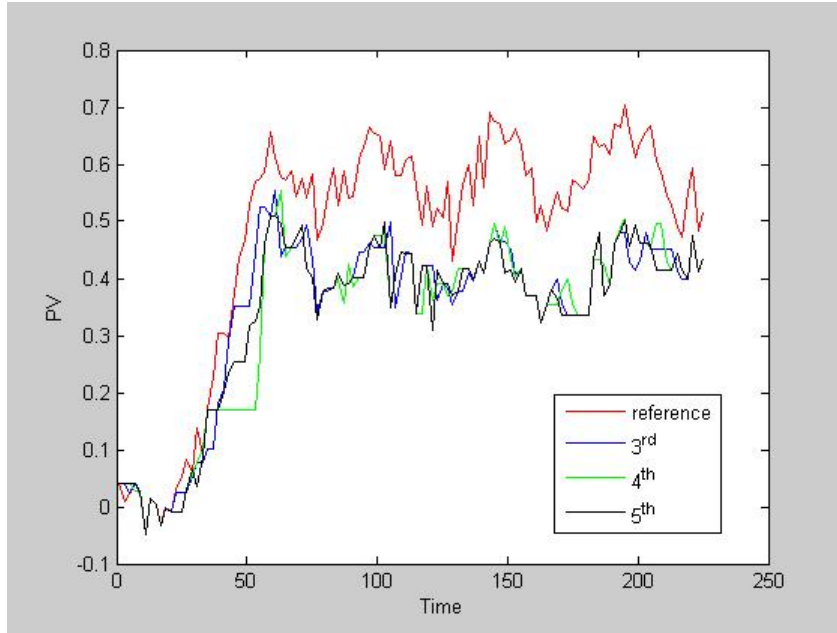


Figure 3.12: Alignment results with different parameter setting.

### 3.3.3 Industrial Data Synchronization

Process data provided by Lubrizol and Emerson Process Management are tested here (the values are masked). There are more than twenty process variables measured and for demonstration purpose, one with most interesting features is shown here (Figure 3.13). The whole trajectory can be roughly divided into four parts and each part has different lengths, heights, and slopes. The total length of  $x_{ref}$  and  $x_{new}$  are different as well. Furthermore, different from previous examples, the reference and new trajectories crossover each other throughout the batch. These commonly faced features challenge both distance and derivative based methods in many ways. For DTW (distance based), the crossings between two trajectories indicate small distance differences, therefore the algorithm may have difficulty in discriminant  $x_{ref}$  and  $x_{new}$ , and results will bias toward the reference trajectory. For DDTW (derivative based), the process noise may lead to unsatisfactory results.

For RDDTW (derivative based), the polynomial order needs to capture the important feature changes (transient periods between phases) as fast as possible with noise filtered out. Also, for all methods, the bandwidth constraint is another important parameter to discuss.

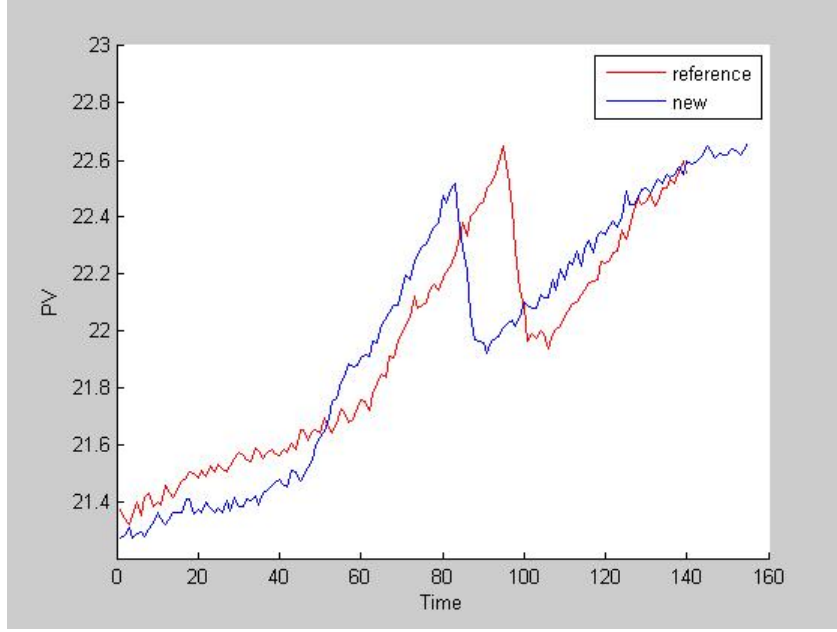
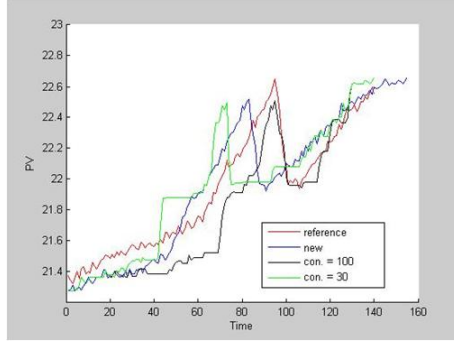


Figure 3.13: Reference and New Trajectories for Lubrizol chemical specialty Process.

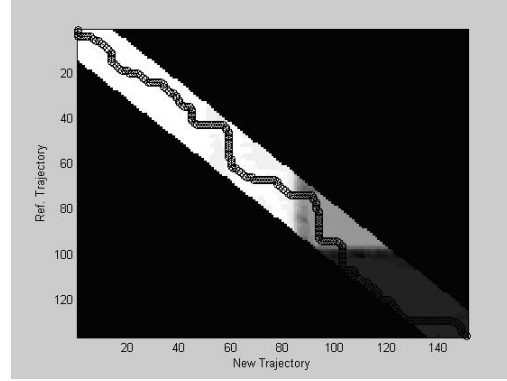
Figure 3.14(a) compares the effect of the bandwidth constraints on RDDTW. When the bandwidth constraint is equal to 30, the synchronized trajectory (in green) deviated significantly from both reference and original ones. Meanwhile, if the bandwidth constraint is equal to 100 (in black), the resulting trajectory retained most original data features and was synchronized well with the reference. The small bandwidth constraint leads to a problem because the important alignment feature points are not included in the allowed synchronization region. By looking at Figure 3.13, the most significant features are the peak and the valley between time period 80 and 105. We name these *key points* for synchronization. It

is important to align the *key points* with each other. In other words, include *key points* in the allowed synchronization region. For this problem, the *key points* are: 1,  $\langle 81(\text{original}), 95(\text{reference}) \rangle$  and 2,  $\langle 91(\text{original}), 105(\text{reference}) \rangle$ , representing the peak and valley. This is illustrated by the surface plots in Figure 3.14(b) to (d), where the lighter part indicates the allowed region for synchronization. Figure 3.14(b) indicates when the *key points* are not included, the synchronized path (-o-) cannot pass through those points that *must be aligned*. If the *key points* are included (Figure 3.14(c) and (d)), RDDTW is able to go through those points and obtain the correct alignment results. Because the main purpose of setting the bandwidth constraint is to save computational resources, it is reasonable to set it to a relatively large value when memory storage is not an issue. Though RDDTW is used here, DTW and DDTW follow the same rule for the bandwidth constraint choice as *key points* are a feature of trajectories.

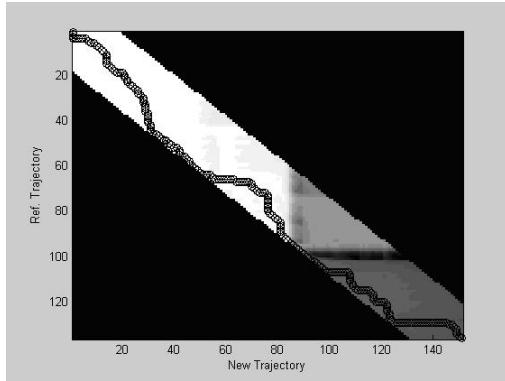
After setting the bandwidth constraint to a suitable number (100 in this case), DTW, DDTW and RDDTW are tested. For DTW and DDTW, there is no other parameter to be specified. For RDDTW, the polynomial order ( $m$ ) and number of points to fit the polynomial ( $2h + 1$ ) need to be specified. In Section 3.3.2, it is suggested to set the polynomial order to  $h$ . This rule is tested here with  $h = 2, 3$ , and 4. First, DTW, DDTW, and RDDTW ( $h = 2$ ) results are compared in Figure 3.15. In Figure 3.15(a), DTW results mostly overlap with the reference trajectory, which means the results are heavily biased toward the reference trajectory and original features such as peak height, rising slope, etc. are lost. For DDTW, the rising period from time 50 to 90 is not synchronized with the reference and RDDTW gives a well-synchronized trajectory with less bias toward the reference trajectory. In Figure 3.15(b), DDTW and RDDTW are plotted against the original trajectory. It is apparent that DDTW did not shift the original data until time 90, which implies that under industrial noise conditions, DDTW may have difficulty in accurate derivative



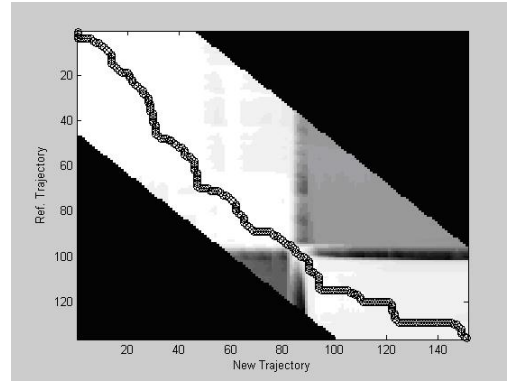
(a) synchronized traj.



(b) bandwidth constraint = 30



(c) bandwidth constraint = 40

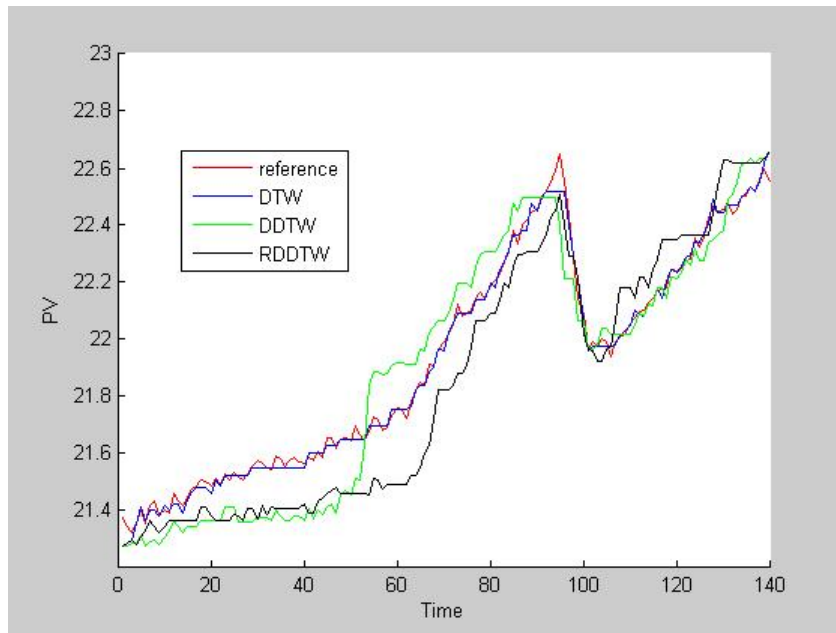


(d) bandwidth constraint = 100

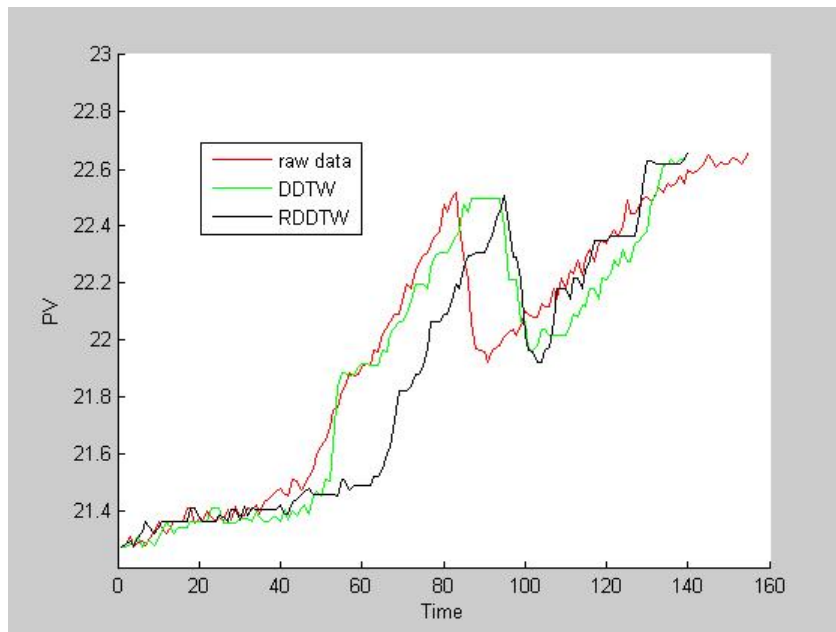
Figure 3.14: Alignment path with different bandwidth.

estimation. RDDTW warped the data and retained the important features including rising ratio, peak height and valley depth.

The last parameter to discuss in RDDTW is  $h$ . Figure 3.16 shows the results for different  $h$  and it is clear that different values of  $h$  give very similar results, which is consistent with the results of the dynamic simulation example (Figure 3.12). These results suggest when the number of points used to fit the polynomial is set to  $2h + 1$ , the algorithm is robust to different polynomial orders ( $h$ ).



(a) Reference



(b) Original

Figure 3.15: Alignment results comparison for industrial data case.

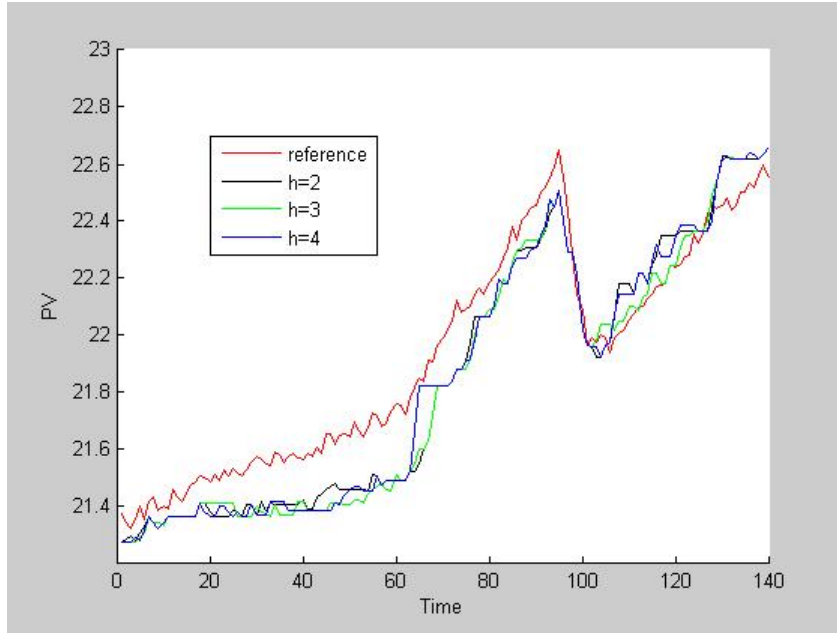


Figure 3.16: Alignment results with RDDTW and different  $h$

Table 4.2 compares the CPU time spent for different algorithms. To highlight the differences, each algorithm was called 100 times by an Intel 1.7GHz computer. With increment of bandwidth constraint, all three algorithms take more time as more point-by-point calculations are required, but all three methods take similar times ( $\pm 0.5$  second at most) to perform the calculation. DTW as the foundation algorithm takes the lowest amount of time. DDTW, as indicated in Section 3.1.2, takes nearly the same time as DTW, which means the left point estimation step is computationally cheap. For RDDTW, because the polynomial parameter values for SG filter were stored in a file ahead of time, the online computation step only reads those values and the extra time taken is negligible. Moreover, different polynomial orders did not affect the speed too much because the vector multiplication of a 3<sup>rd</sup> and a 4<sup>th</sup> order polynomial is nearly the same computationally. Therefore, by using RDDTW, the online calculation burden is only slightly increased and the benefit can be significant.

Bandwidth Constraint	Times (s)			
	DTW	DDTW	RDDTW ( $m = 4$ )	RDDTW ( $m = 3$ )
100	27.65	27.80	28.72	27.75
80	24.35	24.52	24.98	24.79
60	20.97	20.96	20.99	21.28

Table 3.2: CPU Time Comparison by Different Algorithms

### 3.4 Summary

In this chapter, a new algorithm (RDDTW) is proposed to perform data alignment for batch processes that can be summarized as three steps:

1. Piecewise averaging according to sampling frequency and process dynamics;
2. SG filter to estimate local numerical derivatives;
3. Dynamic optimization to calculate the alignment path.

Three examples are used and synchronization results show that RDDTW can significantly reduce the number of *singularity points*, retain the most important features of the original trajectory and avoid bias toward the reference profile. In addition, choosing three algorithm parameters (bandwidth constraint, polynomial order, and number of points used to fit the polynomial) are discussed and suggestions are made.

Although the examples shown in this research are mainly chemically related, RDDTW can be used in other time series analysis such as gesture recognition, manufacturing, speech processing and medicine.

## Chapter 4

# A Combined EWMA-HMPCA Technique

Section 2.5 and related references [97, 54] proved that Hybrid-wise unfolding Multiway PCA (HMPCA) is effective for small batch fault cases. However, no matter which unfolding method is applied, available Multiway PCA (MPCA) methods treat the monitored processes in a static fashion and do not take process dynamics into account. These methods assumed that one variable at a specific time is only correlated to other process variables at that time and independent of all earlier and future process variable measurements (time independent). This assumption can be detrimental to monitoring processes with a small amount of propagating variation or drift. These faults cannot be identified until their magnitudes reach the static control limit. Wold [93] utilized the Exponentially Weighted Moving Average (EWMA) method to detect small levels of variation, and Yoo and Lee [95] combined EWMA with kernel PCA to consider drifting effects in the process. However, these efforts are limited to continuous processes.



Unlike a continuous process, a batch process has a strong time dependency (dynamics). Thus, it is necessary to incorporate dynamic effects into modeling to capture more information of the batch. To do this, Chen and Liu proposed Batch Dynamic PCA (BDPCA) [18] based on an earlier Dynamic PCA (DPCA) framework for continuous processes [53]. BDPCA has been shown to be sensitive to small process variations and drifts but it requires much more computational resources compared with traditional MPCA [99], which will be discussed in Section 4.1.2.

In this chapter, a new method incorporating batch process dynamics into PCA is proposed by incorporating EWMA with HMPCA (E-HMPCA). In Section 4.1, HMPCA, Batch Dynamic PCA (BDPCA) and EWMA methods are reviewed. The EWMA combined HMPCA algorithm is proposed in Section 4.2. Results from three well-known batch process examples are compared in Section 4.3. Finally, a summary is given in Section 4.4.

## 4.1 Related Techniques

### 4.1.1 HMPCA

In Section 2.5, HMPCA is introduced as the best Multiway PCA approach because it combines the advantages of both batch-wise unfolding and variable-wise unfolding approaches. In short, by using HMPCA, batch nonlinearity is reduced while the future observation prediction step is avoided. After unfolding, the resulting matrix  $X$ ,  $(IK \times J)$  is decomposed by PCA:

$$X = T \times P \quad (4.1)$$

where  $T$  and  $P$  are  $IK \times n_{pc}$  and  $J \times n_{pc}$  matrix, respectively. Hence, the scores ( $T$ ) preserve the major statistical relations with respect to time and batch. Next, the

large score matrix is time-sliced into  $t_k(I \times n_{pc})$  components and the corresponding covariance matrix at each time is  $S_k = \frac{t_k^T t_k}{I-1}$ .

For online monitoring, when a new observation ( $x_{new,k}$ ) is taken,  $T^2$  and SPE control charts are used for analysis:

$$t_{new,k} = x_{new,k} \times P \quad (4.2)$$

$$T_{new,k}^2 = t_{new,k} S_k^{-1} t_{new,k} \leftrightarrow \frac{n_{pc}(I^2 - 1)}{I(I - n_{pc})} F_{n_{pc}, I - n_{pc}} \quad (4.3)$$

$$SPE_{new,k} = ((I - PP^T)x_{new,k})^T (I - PP^T)x_{new,k} \leftrightarrow \frac{v_k}{2m_k} \chi_{2m_k/v_k}^2 \quad (4.4)$$

where  $m_k$  and  $v_k$  are mean and variance of the training data at time  $k$ . Different from Eq. 2.31, Eq. 4.4 is used here to calculate the Upper Control Limit (UCL) for SPE and a related derivation can be found in [70]. It is obvious that hybrid-wise unfolding approach only requires the current observation ( $x_{new,k}$ ) for online monitoring and no future data estimation is needed. Basically,  $T^2$  considers the variations in the principal component subspace and SPE measures the magnitude of the sample projection on the residual subspace. Because the residual subspace is more sensitive to process abnormality (Section 2.3) and insignificant faults are tested in this work, the SPE control chart is generally used.

HMPCA has been shown to be more sensitive to small process faults compared to batch-wise MPCA because the covariance matrix  $S_k$  is updated at each time step instead of being static all along the batch. However, as stated above,  $S_k$  in HMPCA only reflects the process variable correlation at current time ( $k$ ), thus process dynamics are not taken into account.

### 4.1.2 Batch Dynamic PCA

To incorporate batch process dynamics into PCA, batch-wise unfolding and data preprocessing (Figure 2.4) are applied and a time-lagged data window for a complete batch is constructed as in Eq. 4.5:

$$X_d^i = \begin{bmatrix} X_d^i(d+1) \\ X_d^i(d+2) \\ \vdots \\ X_d^i(K) \end{bmatrix} = \begin{bmatrix} x^i(d+1)^T & x^i(d)^T & \dots & x^i(1)^T \\ x^i(d+2)^T & x^i(d+1)^T & \dots & x^i(2)^T \\ \vdots & \vdots & \ddots & \vdots \\ x^i(K)^T & x^i(K-1)^T & \dots & x^i(d+1)^T \end{bmatrix} \quad (4.5)$$

In Eq. 4.5,  $i$  is the number of batch,  $d$  is the window length and  $K$  is the total observation number.  $X^i(d)$  is an auto-regressive model structure for batch  $i$  at a specific time. With a proper value for  $d$  (Eq. 31 in ref [53]),  $X_d^i$  can capture the dynamic behavior of a batch process and the corresponding covariance matrix is:

$$S_{X_d X_d}^i = \frac{(X_d^i)^T \times (X_d^i)}{K - d + 1} \quad (4.6)$$

Each term ( $S_{mn}$ ) in Eq. 4.6 represents the dynamic relationship between two variables ( $m$  and  $n$ ) in batch  $i$ . Equation 4.6 only considers one batch, thus in order to consider all batches, the covariances of all  $I$  batches are averaged.

$$S_{X_d X_d}^{avg} = \frac{(K - d + 1) \sum_{i=1}^I S_{X_d X_d}^i}{J(K - d)} \quad (4.7)$$

After the covariance matrix is calculated, the same procedures as batch-wise MPCA are followed. The calculation of the UCL for  $T^2$  and SPE control charts can be found in Section 2.3 and related references [18].

The advantages of BDPCA are that it successfully takes batch dynamics into account and enhances the fault detection ability for small process variations. However,

it has some disadvantages as well:

1. In BDPCA covariance matrix calculation, one has to perform large data matrix calculations. For example, the three-dimensional data matrix is  $I \times J \times K$  and in BDPCA, the dimension of the  $X^i$  matrix would be  $(K - d) \times (J \times (d + 1))$ . Therefore, the covariance  $(S_{X_d X_d}^i)$  is a  $(J \times (d + 1)) \times (J \times (d + 1))$  dimension matrix.
2. In comparison with batch-wise unfolding MPCA, BDPCA calculates the covariance matrix of each batch separately first and then takes the average of all the batches. The shortcoming of this sequence is that the batch to batch variation is lost with only the mean trajectory value retained.

#### 4.1.3 EWMA and Multivariate EWMA [59]

First introduced in 1959, EWMA is a widely used tool in process control and monitoring due to its simple theory and intuitive results. EWMA and Multivariate EWMA control charts can be defined by the following equation:

$$y_i = \lambda x_i + (1 - \lambda)y_{i-1} = \lambda \sum_{j=1}^i (1 - \lambda)^{(i-j)} x_j \quad (4.8)$$

where  $x$  is a time series and  $\lambda$  ( $0 \leq \lambda \leq 1$ ) determines the rate at which older data enter into the calculation of the EWMA statistics. A large value of  $\lambda$  implies more weight is given to recent data and less weight to older data (Eq. 4.8). The value of  $\lambda$  is usually set between 0.05 and 0.3. Generally speaking, a small  $\lambda$  is helpful in detecting small process variations but increases the delay time of detection, while a large  $\lambda$  has the opposite effect. Recently, Yoo and Lee [95] applied EWMA combined PCA to a continuous biological wastewater treatment process with  $\lambda = 0.05$  and the algorithm successfully detected small process drifts that were unobservable to a standard static PCA algorithm.

It is well-known that EWMA is able to take process dynamics into account and is sensitive to process drifts and small variations due to its memory effect compared with traditional statistical control charts [19]. Besides the memory effect, EWMA reduces the variance of the PV by a factor of  $\lambda/(2-\lambda)$ , which also makes the drifting more noticeable. However, EWMA cannot be applied directly to batch processes due to the large nonlinearity of the typical process. For a detailed discussion of EWMA and its applications, see [59, 19, 80].

## 4.2 EWMA-HMPCA

Table 4.1 lists the main differences between BDPCA and HMPCA methods. Although BDPCA takes process dynamics into account by using a lag time window, it fails to consider batch to batch correlation, which is also important for online monitoring. Furthermore, as discussed in Section 4.1, BDPCA requires large data matrix calculations compared to HMPCA.

Method	Correlation Captured	Computational Resources	Process Dynamics Considered
BDPCA	Time to Time	High	Yes
HMPCA	Batch to Batch; Time to Time	Low	No

Table 4.1: BDPCA and HMPCA Comparison

Thus, HMPCA is a promising method and its only shortcoming is that process dynamics are not considered, which means the correlation of process variables is considered in a static way:  $S_k = t_k^T \times t_k / (n_{batch} - 1)$ . Define  $T(IK \times R)$  as a time series that captures the correlations with respect to both time and batch. Thus, our proposed method uses EWMA algorithm to incorporate process dynamics into the

score matrix ( $T$ ):

$$t_{E,k} = \lambda t_k + (1 - \lambda)t_{E,k-1} = \lambda \sum_{j=1}^k (1 - \lambda)^{k-j} t_j \quad (4.9)$$

By doing this,  $t_{E,k}$  is a new time series which depends not only on the current observation but also previous ones, thus batch dynamics is taken into account. The new covariance matrix ( $S_{E,k}$ ) is calculated by the scores at each time:

$$S_E = \frac{t_{E,k}^T \times t_{E,k}}{I - 1} \quad (4.10)$$

Note that the loading matrix  $P$  is kept constant in both HMPCA and E-HMPCA because it is used to capture the relationship between  $J$  variables all along the batch. However, according to Eq. 4.4, SPE statistics only rely on  $P$  because SPE is proportional to  $I - PP^T$ . For SPE to consider process dynamics, EWMA is used to filter the residual subspace projection:

$$e_{E,k} = \lambda e_k + (1 - \lambda)e_{E,k-1} = \lambda \sum_{j=1}^k (1 - \lambda)^{k-j} e_j \quad (4.11)$$

where

$$e_k = (I - PP^T)x_k$$

After introducing EWMA to filter score ( $T$ ) and error ( $e$ ) matrices, one can perform monitoring after a new observation ( $x_{new,k}$ ) is obtained by projecting it to newly defined principal component and residual subspaces:

$$t_{E,new,k} = \lambda t_{new,k} + (1 - \lambda)t_{E,new,k-1} = \lambda \sum_{j=1}^k (1 - \lambda)^{k-j} t_{new,j} \quad (4.12)$$

where

$$\begin{aligned}
t_{new,k} &= P^T x_{new,k} \\
e_{E,new,k} &= \lambda e_{new,k} + (1 - \lambda) e_{E,new,k-1} = \lambda \sum_{j=1}^k (1 - \lambda)^{k-j} e_{new,j}
\end{aligned} \tag{4.13}$$

where

$$e_{new,k} = (I - PP^T)x_{new,k}$$

As a result, the filtered  $T^2$  and SPE control charts can be generated by:

$$\begin{aligned}
T_{E,new,k}^2 &= t_{E,new,k} S_{E,k}^{-1} t_{E,new,k} \\
&\leftrightarrow \frac{n_{pc}(I^2 - 1)}{I(I - n_{pc})} F_{n_{pc}, I - n_{pc}}
\end{aligned} \tag{4.14}$$

$$\begin{aligned}
SPE_{E,new,k} &= e_{E,new,k}^T e_{E,new,k} \\
&\leftrightarrow (v_{E,k}/2m_{E,k}) \chi_{2m_{E,k}^2/v_{E,k}}^2
\end{aligned} \tag{4.15}$$

All the definitions here are consistent with those given in Section 4.1.1. Based on the MPCA online batch monitoring procedure shown in Figure 2.1, a more detailed E-HMPCA algorithm flow chart is generated in Figure 4.1. The main differences between MPCA (Figure 2.1) and E-HMPCA are: the EWMA filtering step is introduced after calculating score ( $T$ ) and loading ( $P$ ) matrices by NIPALS; RDDTW methods (Chapter 3) are used for data synchronization.

In our previous work [99], instead of filtering  $t_k$  and  $e_k$  matrices,  $S_k$  is filtered. However, filtering  $e_k$  and  $t_k$  is more direct as they are time series. Furthermore,  $T^2$  and SPE control charts are quadratic in  $t_k$  and  $e_k$ , respectively (Eqs. 4.14 and 4.15). The advantages of E-HMPCA include:

1. Compared to BDPCA, large matrix calculations are avoided
2. Compared to BDPCA, batch to batch variation is also included

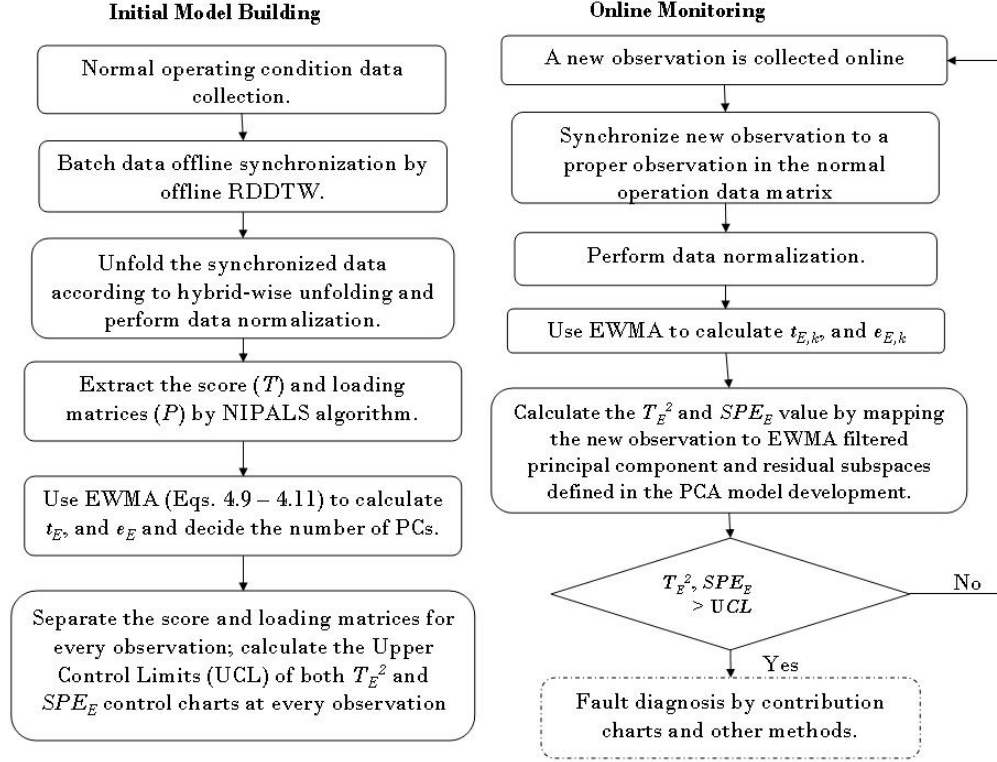


Figure 4.1: E-HMPCA algorithm

- Compared to traditional HMPCA, process dynamics are included
- Similar to HMPCA, this method is easy to implement online because the score matrix  $T$  ( $KI \times R$ ) can be divided into  $t_k$  ( $I \times R$ ) sub-matrices according to time

### 4.3 Case Study

In order to demonstrate the effectiveness of E-MHPCA, three examples are presented. The first one uses industrial data from a Dupont polymerization process; the second one is a bioreactor fermentation process and the third one is an exothermic two phase batch reactor. HMPCA, BDPCA and E-HMPCA are used to analyze the



examples in an online fashion and comparisons are made from the results. Examples with small sensor bias and process drifts are tested here and it is known that the SPE control chart is more sensitive to small faults (Section 4.1), hence SPE control charts and contribution plots prevail here. To demonstrate the E-HMPCA algorithm,  $T^2$  charts are used in the polymerization case where obvious fault exists.

#### 4.3.1 Polymerization

This process was first analyzed by Nomikos and MacGregor [70] in developing batch-wise unfolding MPCA. One batch consists of two phases and each phase lasted approximately two hours. The first phase included ingredients loading, pre-heating, and solvent removal and the second phase was mainly polymerization reaction. The controlled variables are vessel pressure and rate of temperature change. Ten process variables are measured including: temperature, pressure, and flow rates. For more information regarding process description, please consult [70]. All three PCA models (HMPCA, BDPCA and E-HMPCA) are built with 36 normal batches and one abnormal batch is tested. The faulty batch tested (batch 49 in original database) produced marginally acceptable product and worse than normal ones.

#### HMPCA

The raw data are preprocessed by RDDTW, hybrid-wise unfolding, and normalized into zero mean and unit variance, which are covered in previous sections. Parallel Analysis (PA) method [41] indicates three PCs should be retained with 53.07% process variation. Figures 4.2(a) and (b) show the  $T^2$  and SPE calculation results by HMPCA, which indicate some abnormal situations happening between time period 57 and 65. This result is the same as a batch-wise unfolding PCA result [70]. In order to find the root cause that leads to this fault, SPE contribution plots for each variable are presented in Figure 4.2(c). Once an operator finds a fault, an online

contribution plot can be generated to help analyze which variables have the largest deviation from normal operating conditions. In the SPE contribution plot, it can be seen that process variables 6, 7 and 10 exhibit large contributions to the deviation. It is known that the trajectories of variables 6 to 10 have obvious deviations from normal situations [70], which demonstrates the effectiveness of the SPE contribution chart. The monitored process went back to normal after time 65, which indicates the operational problem was corrected. However, due to the faulty period, the resulting product quality was worse than normal. Based on these observations, it can be concluded that hybrid-wise unfolding can be applied for this obvious fault case satisfactorily.

## BDPCA

For the BDPCA method, the time-lagged window ( $d$ ) is set to five. Both R ratio [92] and Predicted Error Sum of Squares (PRESS) suggest 17 PCs should be retained in the BDPCA model, which is a very large number. Figure 4.3(a,b) indicates the fault is detected right after time interval 57. Furthermore, it can be seen that the SPE chart in BDPCA has a more significant violation of the corresponding UCL than HMPCA, because process dynamics are taken into account. However, when looking at the SPE contribution plot, because there are ten process variables and a five lag window, the total number of bars is  $(5 + 1) \times 10 = 60$  (Figure 4.3(c)), which is hard to interpret. Moreover, the contribution plot is not meaningful as the bars are nearly at the same height for most of the variables. The contribution plots failed to explain variable deviations one at a time because the time-dependent process variable correlations are shown explicitly in BDPCA.

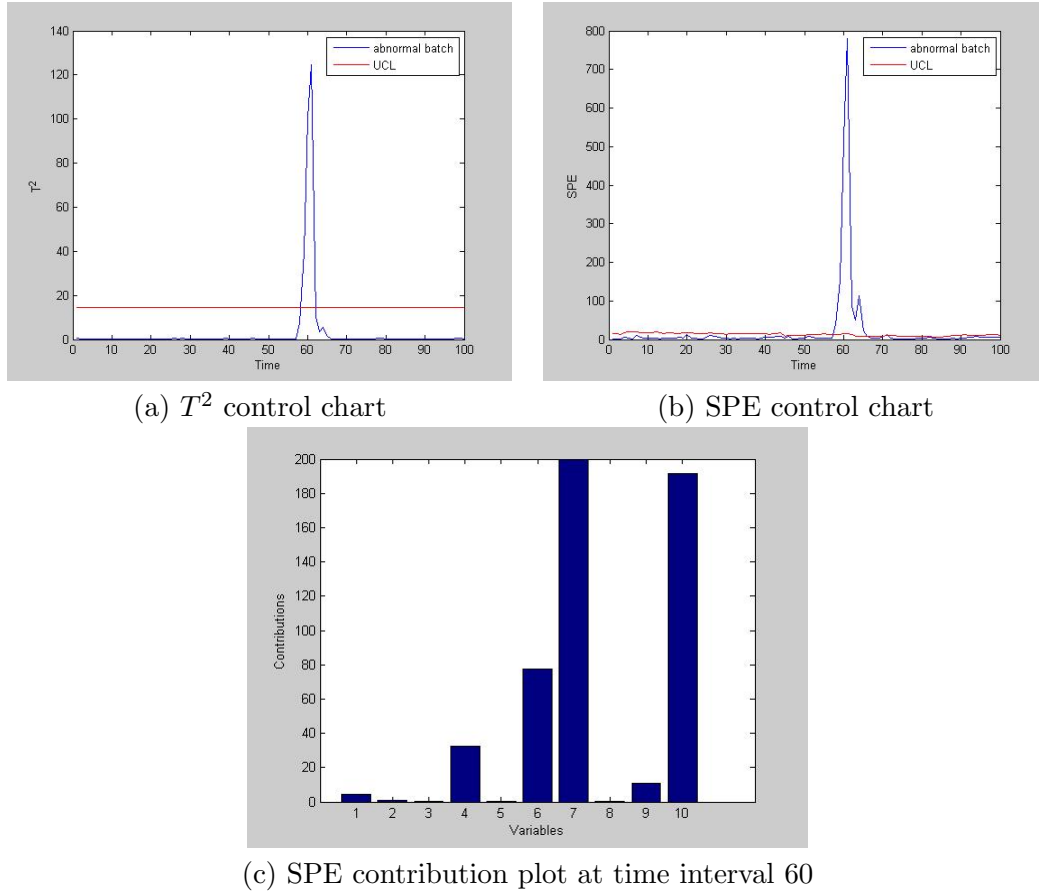


Figure 4.2: Polymerization process online monitoring results by HMPCA

### E-HMPCA

Similarly, Figure 4.4 shows the monitoring result of proposed method. Three PCs are enough to build the statistical model which is significantly smaller than that for BDPCA (17 PCs). E-HMPCA has nearly the same modeling steps as HMPCA, regardless of the EWMA step (in Figure 4.1).  $\lambda$  was set to 0.2 for this case. Both SPE and  $T^2$  control chart shows that a very sharp peak starts at time interval 57. Meanwhile, the SPE contribution plot also suggests variables 4, 6, 7, and 10 have the largest contribution to the fault, which is consistent with HMPCA. Compared with Figure 4.3(c), the contribution plot is easy to read as the time dependent inter-

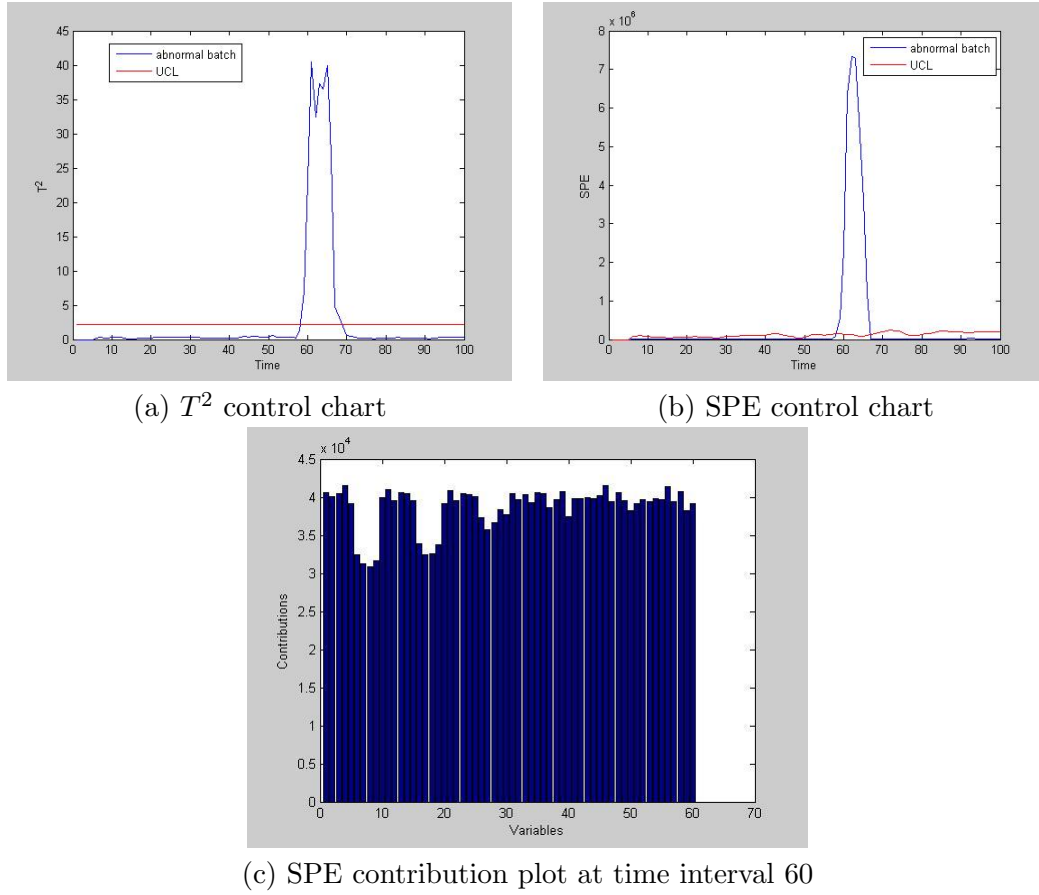


Figure 4.3: Polymerization process online monitoring results by BDPCA

correlations between PVs are implicitly included in the algorithm by the EWMA filter.

Overall, the SPE and  $T^2$  charts demonstrate the correctness and effectiveness of the E-HMPCA algorithm. The three methods indicate there is a fault between time interval 57 and 65. Because this industrial example has an obvious abnormality at *time* = 57, it can be concluded that all three methods give consistent results; among the three, BDPCA and E-HMPCA give better results in terms of fast detection. In addition, the E-HMPCA contribution plot (Figure 4.4(c)) is much easier to interpret

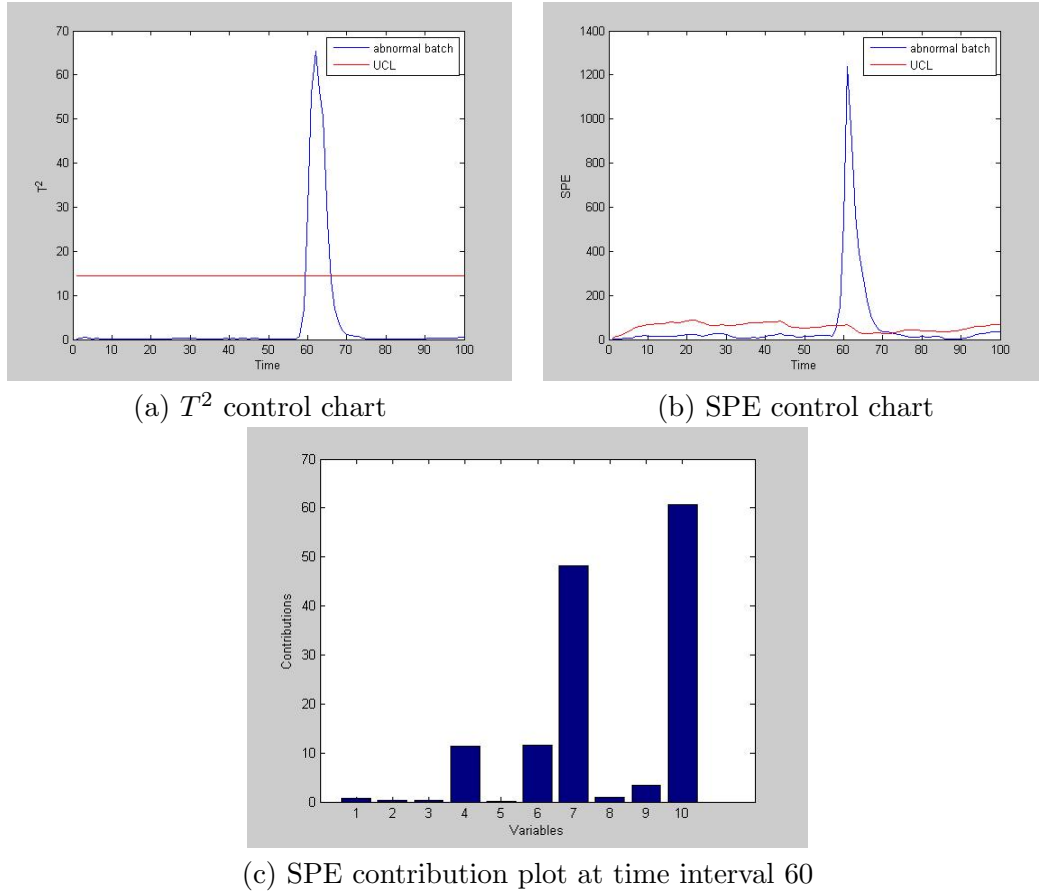


Figure 4.4: Polymerization process online monitoring results by E-HMPCA

and more meaningful compared with BDPCA (Figure 4.3(c)).

#### 4.3.2 Bioreactor Fermentation

A detailed bioreactor model was developed by G. Birol et al. [12] and simulation is performed on a DeltaV computer control system emulator (<http://www.emersonprocess.com>). By using DeltaV, controllers can be tuned/changed easily and various faults can be generated. The simulated bioreactor volume is shown in Figure 4.5(a) and the time scale is one thousand times faster than real time [13]. Variations are added to the process variables to mimic a real industrial plant, so all normal batches will fluctu-

ate around a mean trajectory. Process variable values are sampled every 2 seconds. Twelve process variables are monitored regularly, which are batch time, base reagent flow rate, head pressure, vent flow rate, substrate concentration, biomass concentration, product concentration, broth pH, dissolved oxygen concentration, current product yield, coolant water valve and broth temperature. In Figure 4.5(b), the PVs are rescaled to fit in the same figure and it can be seen that this is a one phase batch with large nonlinearity. PID controllers are used to control dissolved oxygen, substrate concentration and broth pH. Two types of batch profiles are used to test the algorithms: 1). Normal batch; 2). Bioreactor pH sensor has small bias during the entire batch.

Data from eleven normal batches are used for batch-wise MPCA model building. Recently, Gunther et al. [36] used ten normal batch profiles for PCA modeling and obtained satisfactory monitoring results. In this work, similarly, a small number of normal trajectories are used to test not only MPCA-based algorithm but BDPCA and E-HMPCA. Biological processes usually take a long time to finish a batch so it is time consuming in collecting data for offline model building. A small number required for statistical modeling can relief the burden.

HMPCA, BDPCA, and E-HMPCA are used here to address the importance of introducing dynamics to detect small faults. According to Parallel Analysis (PA) [41], four PCs with 70% of the variance captured is suggested in both HMPCA and E-HMPCA model building. The EWMA factor  $\lambda$  is equal to 0.2 in this case. For BDPCA, four lag window and ten PCs are used with 73% variance captured. Again, BDPCA requires more computational resources.

Figures 4.6 and 4.7 show the online monitoring results by applying the three methods. It can be seen that for the normal batch case (Figure 4.6), E-HMPCA

Case	Methods	Type 1 (%)	Type 2 (%)	Delay
Normal Batch	HMPCA	NA	7.12	NA
	BDPCA	NA	6.9	NA
	E-HMPCA	NA	3.88	NA
pH Sensor Bias	HMPCA	52.4	NA	20
	BDPCA	24.9	NA	NA
	E-HMPCA	15.5	NA	NA

Table 4.2: Bioreactor Fermentation Process Monitoring Results

has a larger proportion of monitored points under UCL and the curves in Figure 4.6(c) is smoother than in Figure 4.6(a). The reason is that EWMA reduces the process variance by a factor of  $\lambda/(2 - \lambda) = 9$  so the high frequency noise is partially canceled out. Generally speaking, few points violate the UCL in all the figures and we can conclude the process is in control.

Figure 4.7 gives the monitoring results of the pH sensor bias case. One can see clearly BDPCA and E-HMPCA detect the fault faster than HMPCA. Not only do more points violate the UCL compared with HMPCA but also the magnitude of violation is much greater. Among the three, in BDPCA the points violate the UCL with the largest magnitude. However, E-HMPCA shows more consistent detection results than BDPCA. Related monitoring statistics are listed in Table 4.2. Quantitatively, for the normal batch case, E-HMPCA also shows the smallest amount of Type 2 (false alarm) rate and the other two have similar results. For the small pH sensor bias case, E-HMPCA gives less Type 1 (misdetection) errors than do HMPCA and BDPCA. Furthermore, E-HMPCA and BDPCA have no delay in detecting the fault while traditional HMPCA has a 20 time interval delay (when four consecutive points violate the corresponding UCL) (Figure 4.7). Among the three, E-HMPCA shows the best performance with the smallest detection error and no detection time delay. Also, a small number of normal batch profiles (eleven) gives satisfactory modeling and monitoring results.

### 4.3.3 Two Phase Chemical Batch Reactor

A first order consecutive reaction sequence,  $A \rightarrow B \rightarrow C$ , is carried out in a jacketed batch reactor as shown in Figure 4.8. The total batch duration can be divided into two phases. During the first phase, steam is fed into reactor jacket in order to make the system reach the desired reaction temperature. In the second phase, cooling water is used to remove the reaction heat. Reactor temperature is controlled by manipulating coolant water flow rate. The process is described in detail by Luyben [60]. Five process variables are measured: component A concentration; reactor temperature; wall temperature; jacket temperature; and temperature controller output. The total simulation time is 300 and the sampling interval is one.

Fifty normal batch datasets are used to build a PCA model. In HMPCA, the PC number is set to two. In BDPCA, the size of the lag window ( $d$ ) is seven and the number of retained PCs is chosen to be 12, which is again a large number. In comparison, E-HMPCA only needs two PCs to capture the process variation with  $\lambda = 0.2$ . All the PC number selections are based on the PA method.

Two types of faults are generated for testing as described below:

1. Metal wall sensor failure at time interval 100.
2. Activation energy for reaction step 2 ( $B \xrightarrow{k_2} C$ ) changes from 20000 to 22000 at time zero.

Before testing the two faulty datasets, another normal batch is tested. All three methods indicate the process is in control throughout the batch. In the following sections, the two specific faults are tested by all three methods and results are compared.



Case	Methods	Type 1 (%)	Type 2 (%)	Delay
Temp. Sensor Failure	HMPCA	2.0	1.0	5
	BDPCA	2.5	0	2
	E-HMPCA	0.5	0	1
Kinetics Change	HMPCA	71.3	NA	$\approx 210$
	BDPCA	44.0	NA	41
	E-HMPCA	24.3	NA	36

Table 4.3: Chemical Batch Process Monitoring Results

In the first case, the metal wall temperature sensor fails at time interval 100. SPE control charts are shown in Figure 4.9. It is clear that all three methods detect the fault somewhere before time 150 and throughout the following batch. Meanwhile, BDPCA and E-HMPCA detect the fault earlier than MPCA. According to Figures 4.9(b) and (c), the fault is detected right after  $t = 100$  (before 110). In Figure 4.9(a), the obvious spike appears at  $t = 120$ . As a result, it can be concluded that BDPCA and E-HMPCA methods are more sensitive to this fault at the initial period. Related monitoring statistics are listed in Table 4.3. According to Table 4.3, E-HMPCA again has the shortest detection delay and the smallest percentage of Type 1 and 2 errors. Basically, for this obvious fault case, all three methods detect the fault at an early stage and can be used in online monitoring.

The abnormality of the activation energy change case is small, which makes the fault detection difficult in comparison with the metal sensor failure case. This is confirmed by Figure 4.10(a), which shows the HMPCA SPE control chart. One can see that very few points exceed the UCL and according to this chart, the process is in control until time 210 (Table 4.3). In Figures 4.10(b) and (c), BDPCA and E-HMPCA both indicate there is an occurring fault during the batch. Furthermore, E-HMPCA has the fastest detection ( $t = 36$ ) and smallest number of Type 1 errors, according to Table 4.3 and Figures 4.10(b) and (c).

From above results, it can be concluded that BDPCA and E-HMPCA can capture the dynamic relationship between process variables and detect the occurrence of small disturbances, but E-HMPCA has a better fault detection ability for the activation energy change. Furthermore, E-HMPCA does not need the time-lagged window, and thus the covariance matrix calculation is easier to perform. Also the retained PC number is much smaller for E-HMPCA.

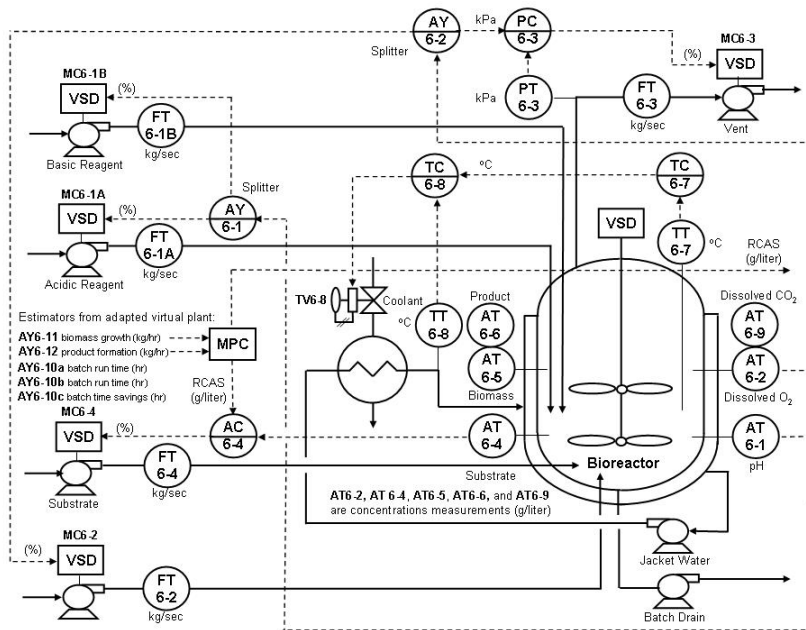
#### 4.4 Summary

Some online batch process monitoring methods (HMPCA) only focus on static batch to batch variations while others take batch dynamics into account by appending older observations to the current time series (BDPCA), which results in a large computational load. In this work, a new method (E-HMPCA) is proposed to include process dynamics and keep the covariance matrix calculation and further online application simple. The new method introduces an EWMA filter to account for batch process dynamics by weighting past score ( $t_k$ ) and error ( $e_k$ ) matrices. In general, E-HMPCA has the following advantages:

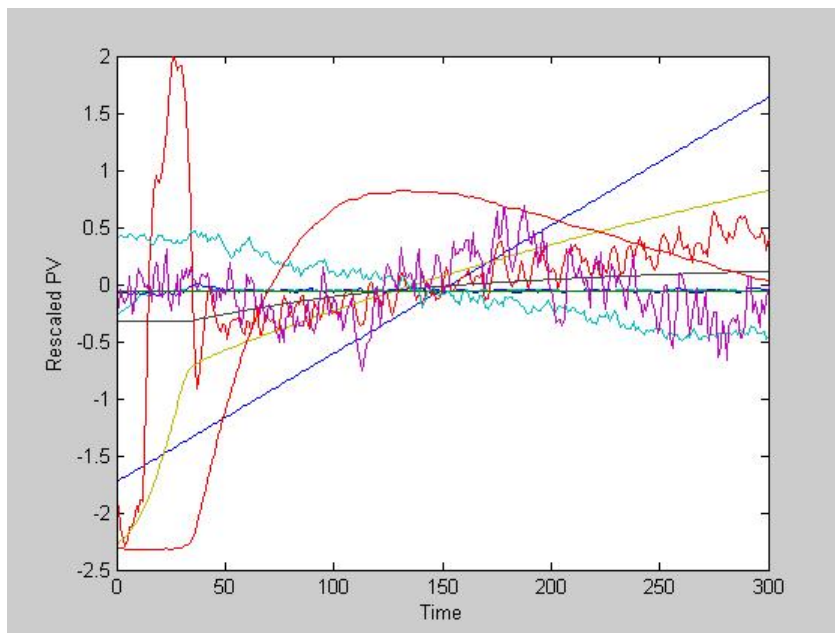
1. easy to implement online (same as HMPCA);
2. retains both batch to batch and time to time variance together with process dynamics;
3. less computationally demanding compared with the lag window approach;
4. needs much fewer principal components and a more useful contribution plot is available compared with the lag window approach.

Three batch process examples are used to demonstrate the effectiveness of the new method. E-HMPCA and BDPCA methods detect process faults faster than

HMPCA, and E-HMPCA performs the best when small process variations are propagated. Furthermore, E-HMPCA enjoys the fewest Type 1 and Type 2 errors for all tested cases among the three methods.

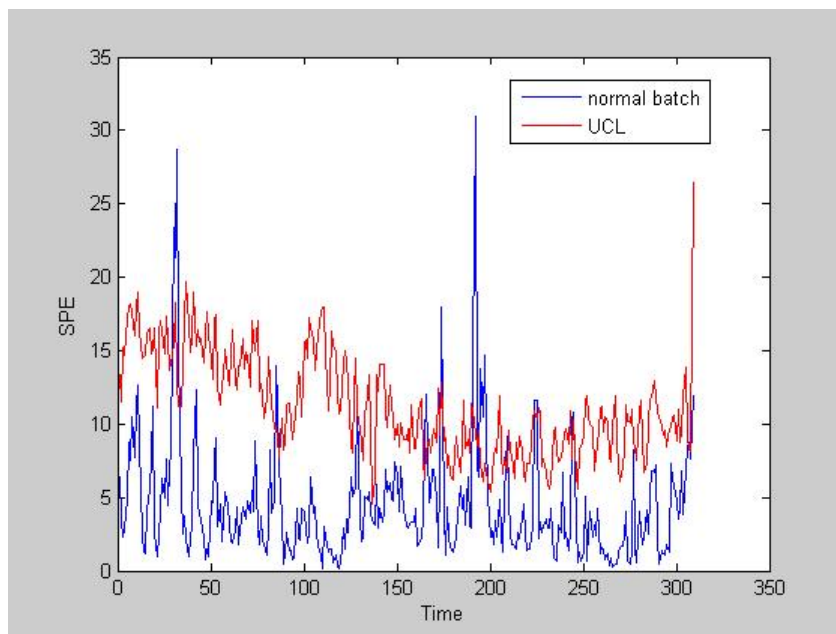


(a) Flow sheet of bioreactor fermenter, Fig 6-3e in [13]

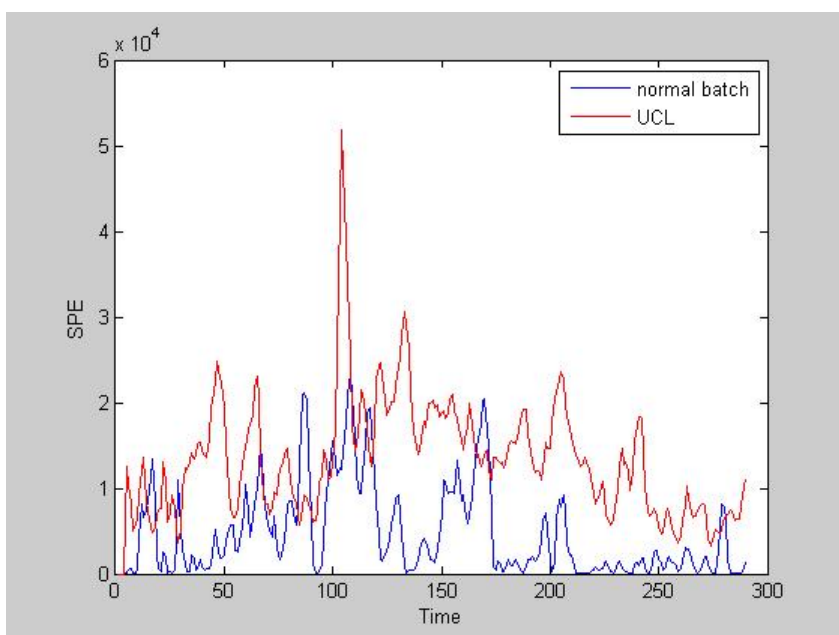


(b) Rescaled PVs for bioreactor fermenter

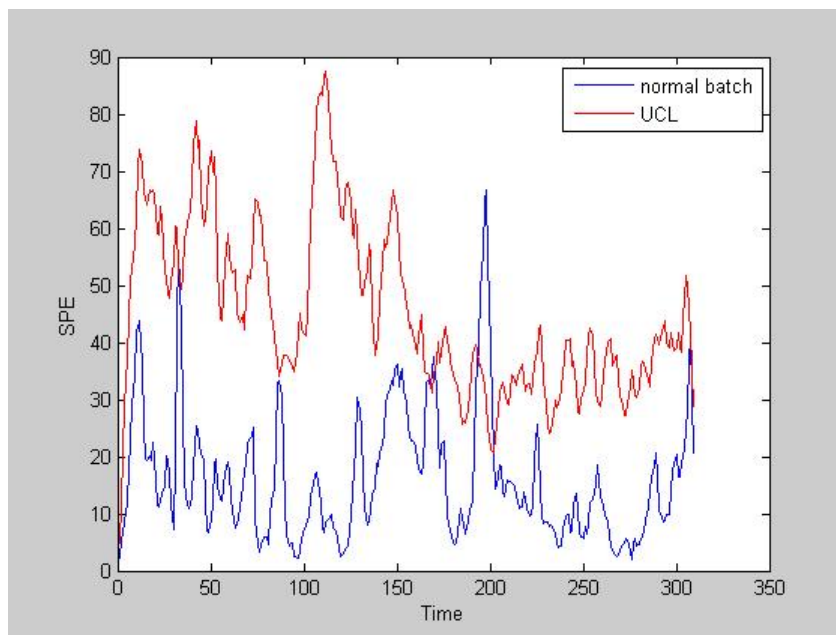
Figure 4.5: Bioreactor fermenter



(a) SPE control chart for HMPCA

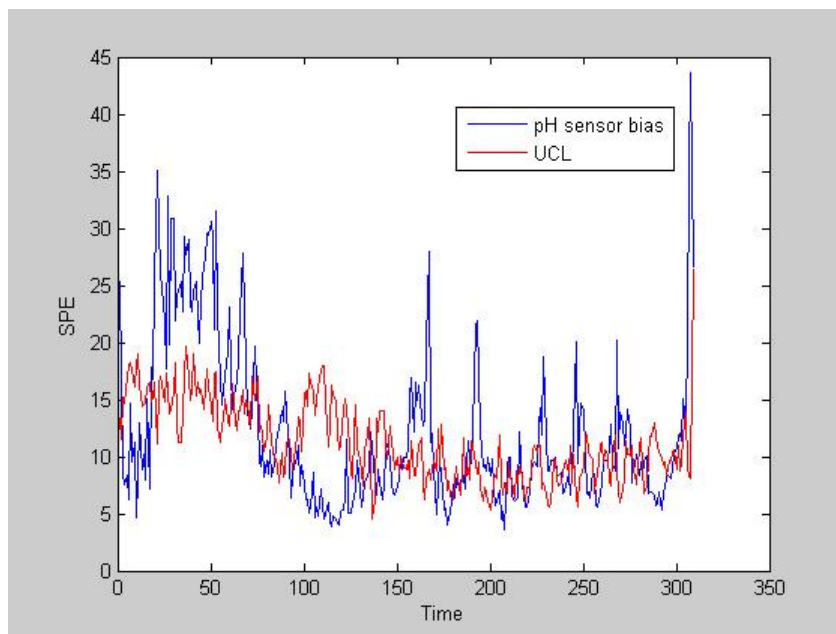


(b) SPE control chart for BDPCA

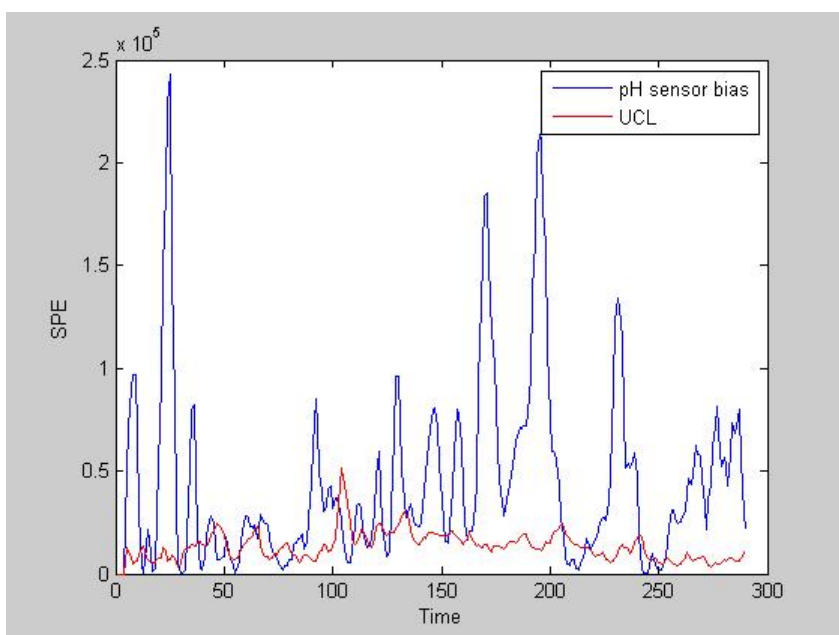


(c) SPE control chart for E-HMPCA

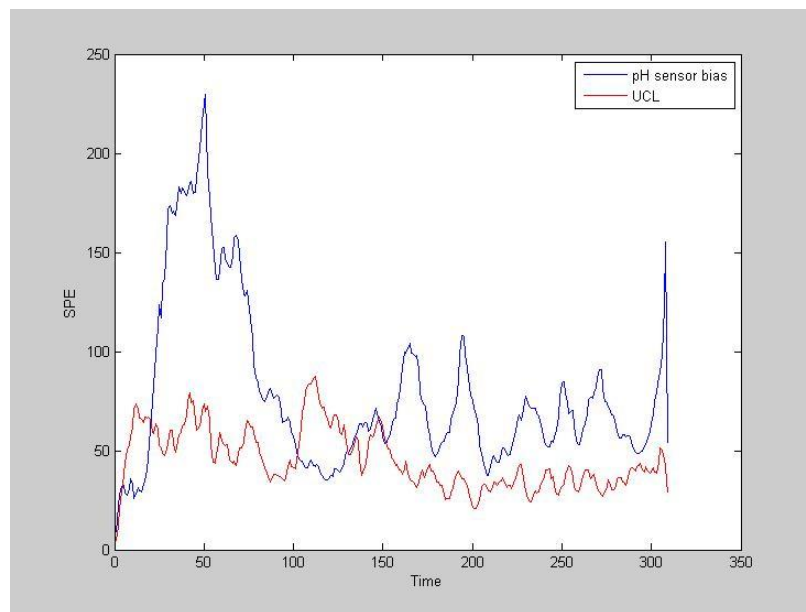
Figure 4.6: Bioreactor process online monitoring results for another normal batch



(a) SPE control chart for HMPA



(b) SPE control chart for BDPCA



(c) SPE control chart for E-HMPCA

Figure 4.7: Bioreactor process online monitoring results for pH sensor bias batch



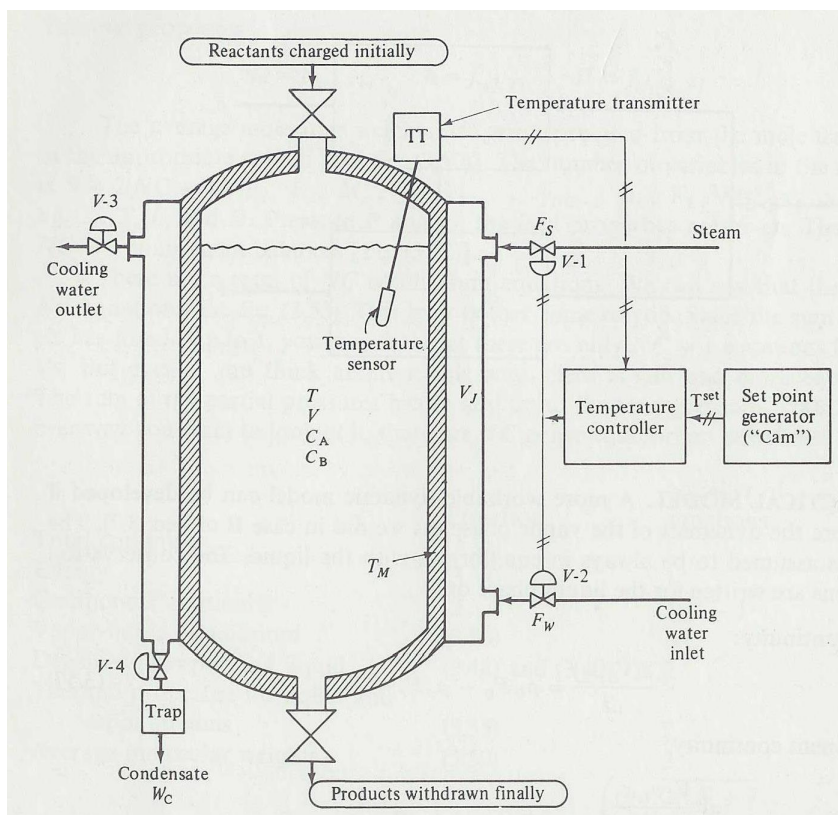
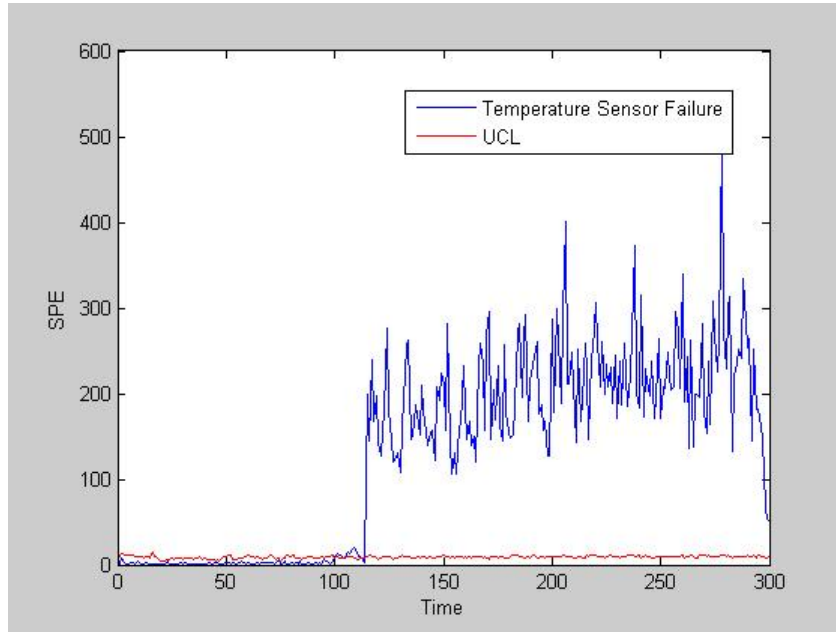
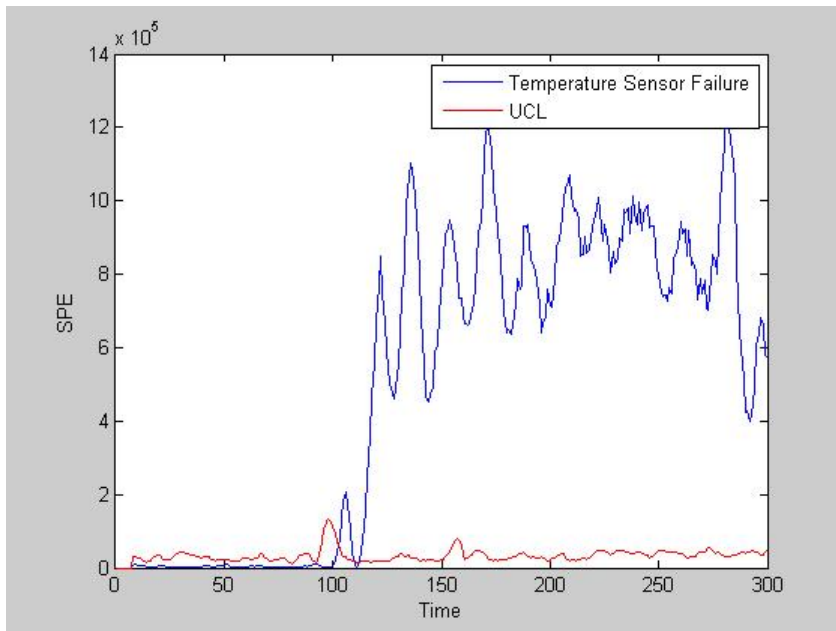


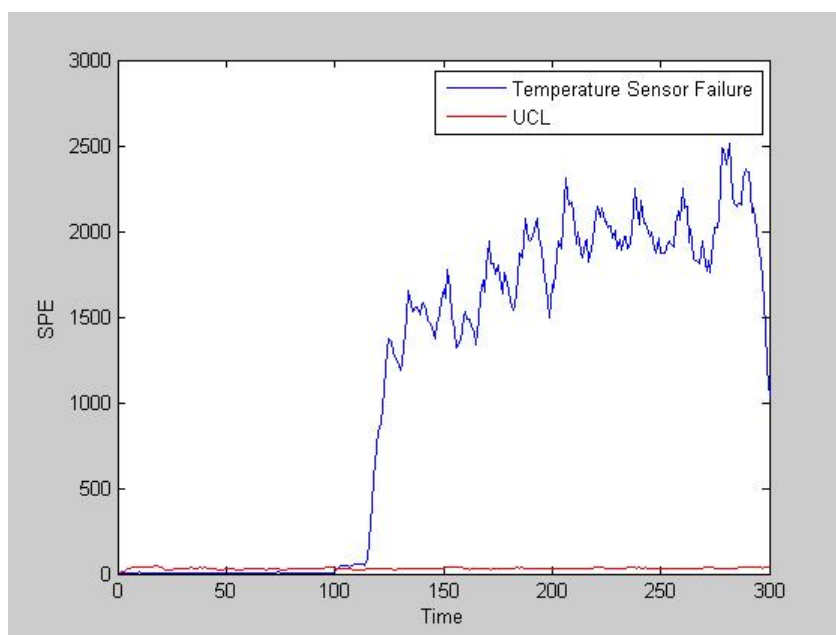
Figure 4.8: Chemical batch reactor with temperature controller, Figure3.9 in [60]



(a) SPE control chart for HMPCA

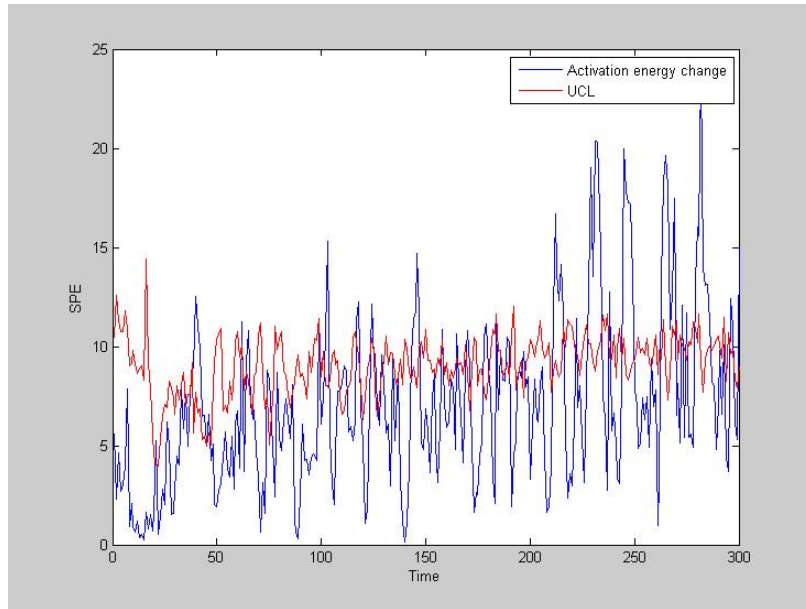


(b) SPE control chart for BDPCA

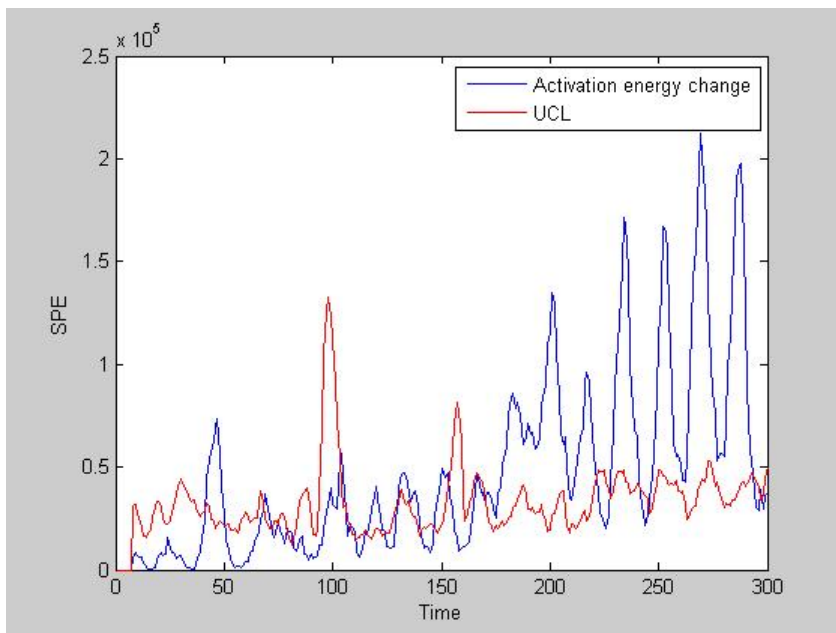


(c) SPE control chart for E-HMPCA

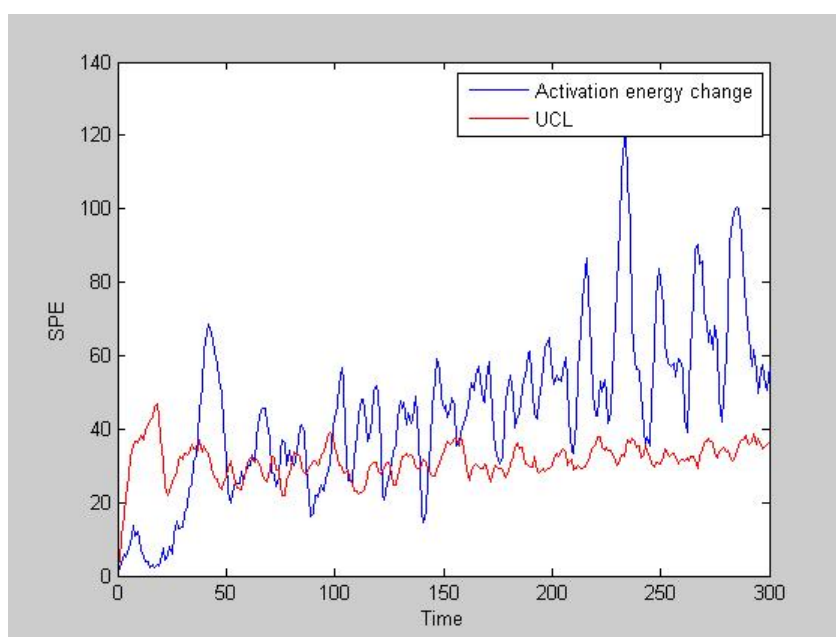
Figure 4.9: Temperature sensor failure of chemical batch process



(a) SPE control chart for HMPCA



(b) SPE control chart for BDPCA



(c) SPE control chart for E-HMPCA

Figure 4.10: Activation energy change of chemical batch process

# Chapter 5

## P-optimal Design Of Experiments

*A mathematical model is only as good as its estimated parameters - Witkowski and Allen*

In Chapter 1, the first principles modeling focuses on theoretical rule based mathematic equations that can reproduce the real-world behavior accurately over a broad range. As widely accepted, first principles modeling is highly creative and innovative work so it is impractical to describe and regulate every modeling procedure. However, no matter which modeling routine we apply, the resulting mathematical models are frequently nonlinear algebraic equations (AE), dynamic algebraic equations (DAE), or partial differential equations (PDE). In chemical engineering applications, PDEs are seldom used due to two reasons: 1. Computationally expensive, not suitable for online oriented control and monitoring purposes; 2. Chemical processes are normally well-mixed so the three dimensional location dependency is not necessary. As a result, AE and DAE systems are the most frequently used first principles modeling techniques. The model parameters are typically used to describe

special properties such as reaction orders, adsorption kinetics, etc.

For industrial scale problems, the model parameters normally are not known a priori and have to be determined from measurements - *parameter estimation*. However, it is always expensive to get important variables (e.g., concentrations) measured. Furthermore, the plant should not be perturbed too much from the steady state operation conditions, which frequently leads to repetitive measurement results and does not produce new information. This leads to the problem of designing experiments wisely to get the maximum information for specific modeling purposes (e.g., model structure selection, parameter estimation) - *model-based Design Of Experiments (DOE)*.

In this work, a new Principal Component Analysis (PCA)-based optimal (P-optimal) criterion for model-based DOE is proposed that combines PCA with information matrix analysis. The P-optimal criterion is a general form that encompasses most widely-used optimal design criteria such as D-, E- and SV-optimal, and it can automatically choose the optimal objective function (criterion) to use for a specific DAE system. Two engineering examples are used to validate the algorithms and assumptions. The main advantages of P-optimal DOE include ease of reducing the scale of optimization problem by choosing parameter subsets to increase estimation accuracy of specific parameters and avoid an ill-conditioned information matrix.

The chapter is organized as follows: In Section 5.1, DOE related background information are reviewed. Section 5.2 introduced the basic idea of parameter estimation, popular optimal design criteria of DOE, and related PCA knowledge. The new criterion is proposed in Section 5.3 and case studies are used to compare the performance of P-optimal criterion and traditional ones in Section 5.4. Section 5.5 serves as a summary.

## 5.1 DOE background

As detailed mathematical models are highly desirable in process control, design, monitoring and optimization applications, first principles modeling is becoming more and more important for process engineers. However, a reliable model for real systems under certain conditions (operation constraints) is usually hard to develop and the procedure is not trivial. Asprey and Macchietto proposed a general modeling procedure [3]:

- Preliminary system analysis and model structure building based on process information
- Design optimal experiments according to model structure
- Carry out experiments as designed
- Use experimental data to estimate model parameters and perform model validation by analyzing estimated parameters and available data

Before discussing the detail, DOE here is designed for a specific DAE or AE system, so called *model-based DOE*. There exists another large family of DOE, which is basically *factorial analysis* based DOE techniques using empirical models. By analyzing the procedure, DOE is performed before real experiments are carried out. The objectives of model-based DOE can be model structure selection, model parameter estimation, etc. From an application point of view, they differ mainly on setting up the DOE objective function. Because of the similarity between these applications, this chapter will mainly focus on parameter estimation applications and model structure selection will be covered briefly.

From the listed procedure, it is obvious that experimental data play a key role not only for parameter regression but also for model validation. Thus, how to design



experiments to get good parameter estimation results with a limited experimental effort is an important issue. Design Of Experiments (DOE) is a statistical tool to choose the optimal experimental conditions for deterministic models. Following the pioneering work of Box and Lucas [14], Atkinson and Hunter [9], the advantages from the use of DOE are well-known, with applications to many areas, e.g., linear discrete time model frequency domain analysis [57, 35], adsorption kinetics [5], reaction kinetics [42, 30, 8], biological networks [31, 81], and fermentation processes [3].

From an algorithm development perspective, DOE has been utilized to AE systems for a long time and was applied to DAE systems in the early 1990s [100, 4]. Many optimal design criteria have been proposed (D-optimal, E-optimal, etc.) and compared by different case studies; Walter and Pronzato [88] gave a detailed discussion of available optimal design criteria and their geometrical interpretations. Recently, Atkinson [6] used DOE for non-constant measurement variance cases and Galvanin et al. [32] extended the DOE territory to parallel experiment designs.

Despite its many applications and attractive results, DOE still has some shortcomings. DOE may not be applicable to large/medium scale DAE systems due to numerical difficulties and calculations usually take a long time to converge. No clear rule set is available in choosing an optimal design criterion for a specific case. More discussion on the advantages and shortcomings of available DOE methods is provided in Section 5.2.

This work focuses on a generalized methodology that introduces Principal Component Analysis (PCA) into DOE. By combining the two methods, one can elegantly divide the large system into small pieces and design a series of experiments to avoid the numerical problems discussed above. Furthermore, the new proposed criterion reduces to well-known optimal design criteria (D- E-, SV- optimal) under certain

assumptions and one can choose a subset of model parameters to increase estimation accuracy without changing the form of the objective function (Ds-optimal [7]). Galvani [32] proposed SV-optimal, which also used Singular Value Decomposition (SVD). However, further analysis of the relationship of SVD results with model structure and physical background was not performed.

## 5.2 Related Techniques

### 5.2.1 Parameter Estimation

Parameter estimation is a classical problem for many science and engineering disciplines which can be generalized into the following optimization problem in many chemical engineering applications:

$$\min_{\theta} \sum_{i=1}^n \sum_{j=1}^q (y_{m,i,j} - f_j(t, x_i, \theta, u))^2 \quad (5.1)$$

subject to:

$$H\dot{x}_i = f_j(t, x_i, \theta, u)$$

$$u_{min} \leq u \leq u_{max}$$

$$t_0 \leq t \leq t_f$$

$$x_{min} \leq x \leq x_{max}$$

$$\theta_{min} \leq \theta \leq \theta_{max}$$

where  $n$  and  $q$  are the number of experiments and equations, respectively.  $y$  stands for measured variables and subscript  $m$  indicates measured value.  $x$  is the state variables of the DAE system. For simplicity, the state variables ( $x$ ) are assumed to be measurable, thus  $y = x$ .  $f$  represents the DAE equations and  $H$  is used to discriminate algebraic and dynamic equations (the corresponding rows for AEs are zero).  $\theta$  represents the model parameters and  $u$  is the controlled variables. Assume

the control profile  $u$  is known over a predefined time horizon ( $t_f$ ). In parameter estimation, the only unknown in integrating  $f$  is  $\theta$  and normally the boundary of  $\theta$  is defined according to physical, chemical laws or process insight. One more assumption is the measurement noises follow a multivariate normal distribution ( $N(0, V_m)$ ), otherwise Eq. 5.1 needs to be rebuilt from MLE according to the specific noise distribution function. In most cases, however, normally distributed noise is a safe assumption.

Eq. 5.1 is similar to the classical optimal control problem in which the objective function usually is:

$$\min_u x(t_f)$$

The techniques to solve this dynamic system optimization problem can be classified into *sequential* and *simultaneous*. In sequential approaches, only the unknown variables (e.g.,  $\theta$  for parameter estimation,  $u$  for optimal control) are discretized and manipulated directly by the NonLinear Programming (NLP) solver. After the unknown variables are updated, the DAE is integrated given the initial condition ( $x_0$ ) and integration time ( $t_0, t_f$ ). For simultaneous methods, all the state variables ( $x$ ) are discretized along  $t$  according to orthogonal collocation and approximated by polynomials between two neighboring discretization grids. Thus, the integration step is avoided and both state and unknown variables are changed by NLP directly with certain constraints. A review of these methods can be found in [27].

After the NLP solver converges, the corresponding  $\theta$  is our best estimate ( $\hat{\theta}$ ) based on the available measurements. To evaluate the accuracy of the estimation, the posterior covariance matrix is defined by:

$$V(\hat{\theta}, \phi) = \left[ \sum_{r=1}^q \sum_{s=1}^q v_{m,rs}^{-1} J_r J_s + V_0^{-1} \right]^{-1} \quad (5.2)$$

where  $\phi$  is the design vector which typically contains the measurement time, initial state condition, control variables, etc.  $v_{m,rs}$  is the  $r^{th}$  term in the measurement covariance matrix that can be estimated by:

$$v_{m,rs} = \frac{\sum_{i=1}^n (y_{ri} - f_r(x_i, \hat{\theta})) \times (y_{ri} - f_r(x_i, \hat{\theta}))}{n - 1} \quad (5.3)$$

For AEs, the sensitivity matrix is defined by:  $J_r = \partial f_r / \partial \hat{\theta}$  evaluated at  $n$  experimental points (sampling times). For DAEs,  $V$  matrix can be treated as a sequential experimental design result according to Zullo [100]. With Eq. 5.2 kept the same,  $J_r$  contains the sensitivity coefficient of output  $y_r$  with respect to parameter  $\hat{\theta}$  evaluated at different sampling times ( $t_s$ ):

$$J_r = \begin{bmatrix} \frac{\partial y_r}{\partial \hat{\theta}_1} & \frac{\partial y_r}{\partial \hat{\theta}_2} & \dots & \frac{\partial y_r}{\partial \hat{\theta}_m} & \leftarrow & t_1 \\ \frac{\partial y_r}{\partial \hat{\theta}_1} & \frac{\partial y_r}{\partial \hat{\theta}_2} & \dots & \frac{\partial y_r}{\partial \hat{\theta}_m} & \leftarrow & t_2 \\ \vdots & \vdots & \ddots & \vdots & & \\ \frac{\partial y_r}{\partial \hat{\theta}_1} & \frac{\partial y_r}{\partial \hat{\theta}_2} & \dots & \frac{\partial y_r}{\partial \hat{\theta}_m} & \leftarrow & t_n \end{bmatrix} \quad (5.4)$$

The diagonal terms of  $V$  lead to the estimation confidence region:

$$\sigma_{\hat{\theta}} = F(\alpha, n, m) \times \text{diag}(V^{1/2}) \quad (5.5)$$

where  $F(\alpha, n, m)$  represents  $F$  distribution with confidence level  $\alpha$  and degrees of freedom  $n$  and  $m$ . Intuitively, the smaller the confidence region is, the better  $\hat{\theta}$  is. More discussion on the estimation result evaluation will be given in Section 5.4.

### 5.2.2 Design Of Experiments

As stated in parameter estimation, the smaller  $\sigma$  is, the better estimation results are. Moreover,  $\sigma$  is closely related to parameter covariance matrix  $V$  (Eq. 5.5).

Thus, in model-based DOE, Fisher information matrix ( $M$ ) is defined as the inverse of the estimated parameter covariance matrix  $V$ :

$$M(\theta, \phi) = \sum_{r=1}^q \sum_{s=1}^q v_{rs}^{-1} J_r J_s \quad (5.6)$$

For both static (AE) and dynamic (DAE) systems, the Fisher information matrix plays a central role in model-based Design Of Experiments (DOE) with the goal to design a series of experiments based on the model structure. By carrying out these experiments, the model parameters can be estimated with the best accuracy [11]. In Section 5.1, DOE is performed before parameter estimation, so the best estimation ( $\hat{\theta}$ ) should be replaced by some initial guess or best knowledge so far ( $\theta$ ). The remaining unknown in Eq. 5.6 is the design vector  $\phi$ , which normally contains measurement time, initial state condition, control variables, etc.

After taking inverse (Eq. 5.6), the smaller is the estimation covariance matrix ( $V$ ), the larger the information matrix ( $M$ ) will be. Remember in parameter estimation, a small  $V$  is desired, so in DOE, our goal is to maximize  $M$  by manipulating the design vector ( $\phi$ ). If there is only one model parameter ( $J$  is  $n \times 1$ ),  $M$  should be a scalar and so is  $V$ . As a result, maximizing the absolute value of  $M$  is the only criteria. However, usually  $M$  is an  $m \times m$  matrix ( $m$  is the number of parameters in the model), thus there are several ways to define the minimization criteria, as outlined in previous work on DOE:

*D-optimality:*

$$F = \min_{\phi \in \Phi} (|V|) = \max_{\phi \in \Phi} (|M|) \quad (5.7)$$

Minimize the determinant of the prediction error covariance matrix and thus the volume of the covariance matrix ellipsoid.

*E-optimality:*

$$\begin{aligned} F &= \min_{\phi \in \Phi} (b_1), b_1 \geq b_2 \geq \dots \geq b_m \\ &= \max_{\phi \in \Phi} (\lambda_m), \lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m \end{aligned} \quad (5.8)$$

Minimize  $b_1$ , which is the largest eigenvalue of  $V$ . This is equivalent to reducing the maximum diameter of the covariance ellipsoid of  $\theta$ .

*Ds-optimality:*

$$F = \min_{\phi \in \Phi} \frac{|V|}{|V_s|} \quad (5.9)$$

where

$$V_s = \left( \sum_{r=1}^q \sum_{s=1}^q v_{m,rs}^{-1} J_{r,e} J_{s,e} \right)$$

$V_s$  is the parameter covariance matrix for the parameters of less interest.  $V$  and  $V_s$  are  $m \times m$  and  $e \times e$  matrices, respectively. The only difference between  $V$  and  $V_s$  is that  $J_{s,e}$  is composed of the parameters that are of less interest.

From the above analysis, parameter estimation and DOE are both rely on the information matrix ( $M(\theta, \phi)$ ) analysis with different unknown variables. In parameter estimation, the unknown is  $\theta$  while in DOE  $\phi$  is unknown. The dual relationship between the two is similar to how the Linear Quadratic Regulator (LQR) is related to the Kalman Filter. Based on the same process model ( $\zeta(t, x, u, \theta)$ ), LQR focuses on generating optimal control signals ( $u$ ) while Kalman Filter estimates the state variables ( $x$ ).

In Section 5.1, we know model-based DOE has other applications beyond parameter estimation. In first principles modeling area, model structure selection (model discrimination) is another important topic. Based on available information, one may end up with multiple process model structure candidates. Model discrimination means: by running the fewest number of experiments, one can select the

candidate that matches the process best. To do this, Espie and Macchietto [28] proposed the corresponding objective function for dynamic system as:

$$F = \max_{\phi \in \Phi} \int_{t_0}^{t_f} \sum_{i=1}^{N_M} \sum_{j=i+1}^{N_M} T_{i,j}(\phi, t) dt \quad (5.10)$$

where

$$T_{i,j} = (f_i(\phi, \theta_I, t) - f_j(\phi, \theta_J, t))^T \times (f_i(\phi, \theta_I, t) - f_j(\phi, \theta_J, t))$$

Here,  $N_M$  is the number of candidate model structures and other definitions are the same as in parameter estimation. As continuous sampling is not feasible, the integration is replaced by  $\sum_{t_k}$  in real applications.  $\phi$  from Eq. 5.10 results in the biggest differences among different model structures ( $f_i$ ). Thus, after getting the real experiment profile ( $y_m$ ), the best candidate model structure predicts  $y_m$  most accurately. As indicated in Section 5.1, the objective function is the main difference between model discrimination and parameter estimation. Therefore this work is focusing on parameter estimation and model discrimination can be carried out in a similar way.

As discussed in Section 5.1, model-based DOE has seen numerous applications in the past few decades, however, there are still some drawbacks that require further study:

1. DOE sometimes fails to find out the optimal experimental condition for medium and large scale DAE systems and it usually takes a long time even for small scale systems. Franceschini and Macchietto [30] applied E-optimal to a biodiesel production dynamic model identification problem. According to their research, designing a single set of experiments aimed at improving the estimation of all parameters together failed as the numerical optimization calculations did not converge. It is reasonable to design different experiments for a specific group

of parameters. However, there is no trivial/automatic way to separate the model parameters into different groups wisely.

2. There are many criteria to choose from (D-optimal; E-optimal; A-optimal; Ds-optimal etc.) and there is no clear way to tell which one to use for a specific case.
3. All criteria depend on minimizing the prediction error variance-covariance matrix  $V$  in some sense. Because  $V$  is calculated from  $M$ ,  $M$  may not be full rank or it may be ill-conditioned (condition number is greater than  $10^{10}$ ). In this case  $M$  may not be invertible and  $V$  cannot be numerically calculated. As a result, working on  $M$  instead of  $V$  is suggested.
4. It is hard to focus on improving specific parameters in the DAE system. Ds-optimal suggested above leads to a different form of the objective function and a more complex optimization problem. The different magnitudes of  $|V|$  and  $|V_s|$  can lead to scaling problems.

### 5.2.3 Principal Component Analysis

The algorithm and theory of Principal Component Analysis (PCA) is discussed in detail in Chapter 2. Here, only the relevant properties are reviewed. PCA decomposes the data matrix from experiments ( $X$ ) by the following expression:

$$X = T^T P + E \quad (5.11)$$

$X$  is an  $n \times m$  data matrix, which can be decomposed into score ( $T$ ,  $n \times n_{pc}$ ) and loading ( $P$ ,  $m \times n_{pc}$ ) matrices, while  $E$  is the residual. PCA has very attractive features including:



1.  $t_i$  (the  $i^{th}$  column of  $T$ ) are orthogonal:

$$B = \frac{T^T T}{n-1} = \text{diag}(b_1, b_2, \dots, b_m) \quad (5.12)$$

where  $b_i$  is the  $i^{th}$  eigenvalue of the covariance matrix  $(X^T X / n - 1)$  in descending order.  $t_i$  explains the relationship between each row.

2.  $p_i$  (the  $i^{th}$  column of  $P$ ) are orthonormal:

$$I = P^T P = \text{diag}(1, 1, \dots, 1) \quad (5.13)$$

$p_i$  focuses on capturing the relationship between each column of  $X$ . Because  $X^T X$  is symmetric, its eigenvalues and eigenvectors are real.

The first few columns of  $T$  and  $P$  capture most of the variance existing in  $X$  and when  $n_{pc}$  (the number of principal components) is equal to  $\min(n, m)$ ,  $E$  is a zero matrix. There have been some studies on choosing the optimal number of principal components ( $n_{pc}$ ) to separate useful information and noise in process monitoring [75, 99]. The Cumulative Percent Variance (CPV) is one such method and the threshold for this method can be set to 90%.

$$CPV(n_{pc}) = \left( \frac{\sum_{j=1}^{n_{pc}} b_j}{\sum_{j=1}^m b_j} \right) \quad (5.14)$$

The relationship between PCA and Singular Value Decomposition (SVD) can be explained by manipulating the equations for PCA (Eqs. 5.15b-5.15c) and comparing

them with the SVD Eq. 5.15a.

$$SVD : \quad \frac{1}{n-1} X^T X = W L C^T \quad (5.15a)$$

$$PCA : \quad = \frac{1}{n-1} (TP^T)^T \times (TP^T) = \frac{1}{n-1} P T^T \times T P^T \quad (5.15b)$$

$$= \frac{1}{n-1} P \times (T^T T) \times P^T = P B P^T \quad (5.15c)$$

$X^T X$  is a real symmetric matrix such that the left eigenvector ( $W(m \times m)$ ) and right eigenvector matrix ( $C(m \times m)$ ) are exactly the same and the corresponding eigenvalues are stored in  $L(m \times m)$ . As a result:

$$P = C = W$$

$$L = B$$

## 5.3 PCA Combined Criterion for DOE

### 5.3.1 Information Matrix and Parameter Covariance Matrix

For purpose of simplicity, assume there is only one measured output ( $q = 1$ ) and the measurement error is identity ( $v_{m,rs} = 1$ ), such that Eq. 5.2 becomes:

$$V(\theta, \phi) = [J^T J]^{-1} = [M]^{-1} \quad (5.16)$$

The sensitivity matrix  $J$  can be viewed as  $X$  in the above PCA equations, and  $M$  is proportional to the covariance matrix  $V$  (the scaling factor  $(1/n - 1)$  in the covariance is contained in  $v_{m,rs}$ ). Assume the eigenvalue and eigenvector matrices of  $M$  are  $\Lambda$  and  $P$ , respectively. Substituting Eqs. 5.11 and 5.15c into Eq. 5.16, we obtain:

$$\begin{aligned} M &= J^T J = (TP^T)^T \times (TP^T) = P \cdot \Lambda \cdot P^T \\ V &= M^{-1} = (P \cdot \Lambda \cdot P^T)^{-1} = P^{-T} \cdot \Lambda^{-1} \cdot P^{-1} \end{aligned} \quad (5.17)$$

Because  $P$  is orthogonal:  $P^T \cdot P = I; P^{-1} \cdot P = I \Rightarrow P^T = P^{-1}$

$$V(\theta, \phi) = P^{-T} \cdot \Lambda^{-1} \cdot P^{-1} = P \cdot \Lambda^{-1} \cdot P^T \quad (5.18)$$

From PCA analysis, in order to get  $V$  from  $M$ , one can efficiently decompose  $M$  by SVD or NIPALS as described in Chapter 2 and replace  $\Lambda$  in Eq. 5.17 with  $\Lambda^{-1}$ . In Section 5.2.3, it has been shown the columns of  $P$  explain the relationship between different columns in  $J$ . For example, the smallest eigenvalue in  $M$  is  $\lambda_m$  and its inverse  $\lambda_m^{-1}$  will be the largest eigenvalue for  $V$ , which indicates the largest variance in prediction error covariance matrix. The corresponding eigenvector ( $p_m$ ) gives the direction of the largest variance in the  $m$  dimensional parameter space.

### 5.3.2 Geometric Interpretation

A geometric interpretation of DOE criteria is shown in Figure 5.1, which shows the covariance matrix of a two parameter system. Two eigenvectors  $p_2, p_1$  indicate the direction of largest and second largest direction of variance. The projection of long axis ( $p_2$  direction) on  $\theta_1$  and short axis ( $p_1$  direction) on  $\theta_2$  is proportional to the confidence region of  $\theta_1$  and  $\theta_2$ , respectively. Thus, in order to shrink the estimated confidence region, traditionally D-optimal focuses on shrinking the size of the ellipsoid but it fails when  $\lambda_1$  becomes very large and  $\lambda_2$  is small (the ellipsoid degenerates into a line which also minimizes the size criterion). E-optimal focuses on minimize the longest the axis of the ellipsoid (maximize  $\lambda_2$ ). However, this method may not be efficient when several axes have a similar length (multi-parameter systems). In Figure 5.1, when  $\lambda_1$  is much larger than  $\lambda_2$ , the ellipsoid will degenerate into a line (D-optimal fails) and it is reasonable to look at  $\lambda_2$  alone, which is E-optimal. Instead, when  $\theta_2$  is well known, one can only focus on shrinking the projection of both ellipsoid axes on  $\theta_1$  direction:  $\min \left( \left| \frac{p_1(1)}{\lambda_1} \right| \times \left| \frac{p_2(1)}{\lambda_2} \right| \right)$ . In order to eliminate the absolute value and take advantage of the unit length of  $p_i$ , we use

the following expression:

$$\min \left( \frac{p_1(1)^2}{\lambda_1} \times \frac{p_2(1)^2}{\lambda_2} \right) = \max \left( \frac{\lambda_1}{p_1(1)^2} \times \frac{\lambda_2}{p_2(1)^2} \right)$$

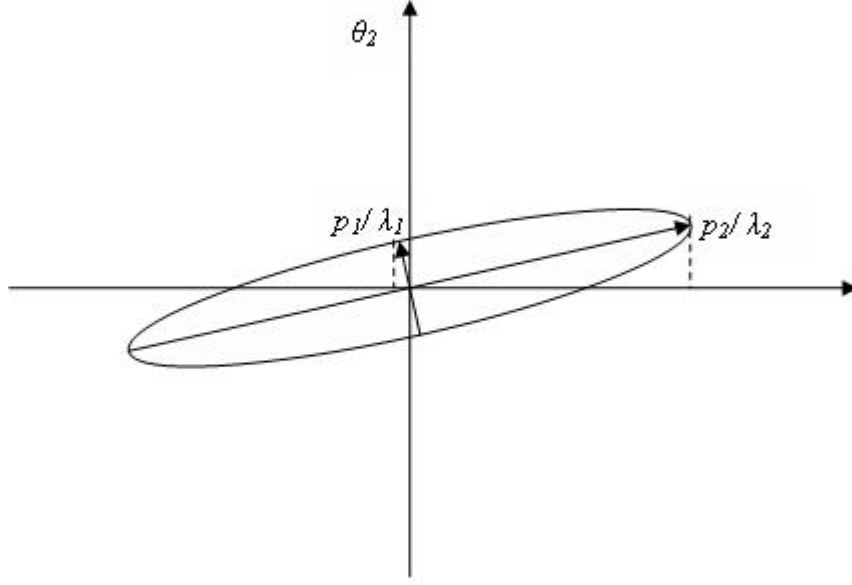


Figure 5.1: Geometric interpretation of PCA combined DOE criteria

### 5.3.3 PCA based optimal Criterion (P-optimal) and its Variation

From Section 5.3.1 and 5.3.2, it is reasonable to formulate the new objective function as follows:

$$F = \min_{\phi \in \Phi} \left( \prod_{i=m-n_{pc}+1}^m \left( b_i \cdot \sum_j P_{ji}^2 \right) \right) \quad (5.19)$$

where  $b_i$  are eigenvalues of  $V$  in ascending order ( $b_i = 1/\lambda_i$  and  $\lambda_i$  is in descending order) and  $P$  is the corresponding eigenvector matrix. The advantage of storing

eigenvalues of  $V$  in ascending order is that  $P$  can be used directly without transformation, otherwise  $P$  for  $V$  needs to be transformed by:  $P_V = P_M \cdot \begin{bmatrix} & & 1 \\ & \ddots & \\ 1 & & \end{bmatrix}$ .  $n_{pc}$  is the number of optimal number of PCs and  $j$  corresponds to the parameters that are selected to increase estimation accuracy.

Suppose we want to improve the precision of all parameters ( $j = 1 : m$ ) and all the PCs are retained by PCA. Then Eq. 5.19 becomes:

$$F = \min_{\phi \in \Phi} \left( \prod_{i=1}^m \left( b_i \cdot \sum_j P_{ji}^2 \right) \right)$$

where  $p_i$  is orthogonal  $\Rightarrow \sum_{j=1}^m P_{ji}^2 = 1$

$$F = \min_{\phi \in \Phi} \left( \prod_{i=1}^m (b_i) \right) = \min_{\phi \in \Phi} |V| \Rightarrow D - optimal \quad (5.20)$$

When only the largest eigenvalue of  $V$  is suggested by PCA ( $i = m$ ) and all parameters are to be estimated, Eq. 5.19 is:

$$F = \min_{\phi \in \Phi} \left( b_m \cdot \sum_j P_{jm}^2 \right)$$

where  $p_i$  is orthogonal  $\Rightarrow \sum_{j=1}^m P_{jm}^2 = 1$

$$F = \min_{\phi \in \Phi} (b_m) \Rightarrow E - optimal \quad (5.21)$$

When all the PCs are to be used with specific parameters to be estimated (e.g., the first  $s$ ), Eq. 5.19 becomes:

$$F = \min_{\phi \in \Phi} \left( \prod_{i=1}^m \left( b_i \cdot \sum_{j=1}^s P_{ji}^2 \right) \right) \quad (5.22)$$

SV-optimal was used for parallel and sequential process DOE by Galvanin et al. [32], which is similar to our approach. After obtaining the eigenvalues of the  $V$  matrix ( $b_i$ ), a series of experiments are designed to minimize  $b_1, b_2, \dots, b_m$  respectively. Furthermore, they noticed that by minimizing some eigenvalues, the estimation of certain parameters will improve, but no analysis was provided. The SV-optimal can be treated as a special case in our generalized approach by setting  $i = k$  and  $j = 1 : m$ .

$$F = \min_{\phi \in \Phi} \left( b_k \cdot \sum_j P_{jk}^2 \right)$$

$$\text{where } p_i \text{ is orthogonal} \Rightarrow \sum_{j=1}^m P_{jk}^2 = 1$$

$$F = \min_{\phi \in \Phi} (b_k) \Rightarrow SV - optimal \quad (5.23)$$

Comparing Eq. 5.23 with Eq. 5.19, SV-optimal cannot be used in designing an experiment that increases the estimation accuracy of specific parameters that is explained in  $b_k$ .

When calculating  $n_{pc}$  by the method described by Eq. 5.14, one can choose the eigenvalues of either  $M$  or  $V$ . When using  $V$ , the last  $n_{pc}$  eigenvalues (kept in ascending order such that  $P$  does not need to be transformed) and the corresponding eigenvectors should be used to define the objective function. When using  $M$ , if the first  $k$  eigenvalues sum to 90%, then the remaining  $m - k$  eigenvalues ( $n_{pc} = m - k$ ) and eigenvectors are used in Eq. 5.18. In real applications, using  $M$  to do PC number selection will prefer D-optimal results while using  $V$  will bias the results toward E-optimal. This is because the first few (most of the time one or two) PCs is enough to capture most of the variance for either  $V$  or  $M$ . Normally DOE tries to increase the estimation precision for most model parameters and usually a single eigenvalue cannot include information for most parameters (some elements in  $p_i$  are close to zero), thus retaining more eigenvalues in the objective function for the first

few runs is preferred. As a result, in later examples,  $M$  is used for calculations and the computational steps for P-optimal algorithm are summarized in Table 5.1.

1	Initialize model parameters ( $\theta_0$ ) and design vector ( $\phi_0$ ) ( $i = 0$ )
2	Calculate sensitivity matrix ( $J$ ) (Eq. 5.4) and information matrix ( $M$ ) (Eq. 5.6)
3	Choose the number of PCs using CPV (Eq. 5.14) and group all parameters into subsets according to $P$ if necessary
4	Optimize the objective function (Eq. 5.19) and find the optimal experimental design ( $\phi_i$ ). If the parameters are divided into groups in step 3, multiple experiments can be designed in parallel (as discussed in [32])
5	Carry out real experiments and use the data to regress model parameters ( $\theta_i$ )
6	Calculate $t$ and $\chi^2$ values to justify the quality of the estimated parameter (model validation). When the quality is not satisfied, set $i = i + 1$ and return to step 2

Table 5.1: P-optimal algorithm description

Generally speaking, the new criterion enjoys the following advantages:

1. For medium and large scale DAE system cases, it is always easy to reduce the scale of the DOE problem by choosing certain parameters out of the entire set to be the focus.
2. The proposed P-optimal criterion is a general form of most widely-used criteria such as D- and E- optimality.
3. By introducing PCA to carry out both eigenvalue calculation and selecting the optimal number of eigenvalues to evaluate, the ill-conditioned  $M$  is avoided. PCA method automatically chooses the optimal number of eigenvalues to be investigated, in order to filter out unnecessary eigenvalues, and reduces problem scale.
4.  $P$  gives a clue on grouping the estimated parameters (selecting subset  $j$  in Eq. 5.19), so it is easy to design an experiment for improving specific parameter estimation, compared with traditional methods.

5. The criteria can be easily embedded in parallel and sequential DOE routines.

## 5.4 Case Study

### 5.4.1 Yeast Fermentation Reactor Model

Nearly all fermentation processes are based on mass and energy balances. Neglecting heat effects lead to a set of differential equations:

$$\frac{dc}{dt} = A \cdot r(c(t)) - G \cdot c(t) + u(t) \quad (5.24)$$

where  $A$  ( $n \times m$ ) contains the stoichiometry information of the  $m$  reactions described in  $r$ , and  $G$  ( $1 \times 1$ ) is the dilution rate and  $u(n \times 1)$  contains the input/output information. In this example, a fed-batch model for baker's yeast is used [24]:

$$\frac{d}{dt} \begin{bmatrix} x \\ s \end{bmatrix} = \begin{bmatrix} 1 \\ -1/\theta_3 \end{bmatrix} \cdot r(\theta, x, s) \cdot x + u_1 \left( \begin{bmatrix} 0 \\ u_2 \end{bmatrix} - \begin{bmatrix} x \\ s \end{bmatrix} \right) + \begin{bmatrix} -\theta_4 x \\ 0 \end{bmatrix} \quad (5.25)$$

$x$  and  $s$  are biomass and substrate concentration, respectively and  $r$  is the specific growth rate, which is assumed to be of the Monod-type:

$$r(\theta, x, s) = \frac{\theta_1 s}{\theta_2 + s} \quad (5.26)$$

There are four parameters to be estimated while two of them are kinetic rate coefficients ( $\theta_1, \theta_2$ ) and the other two are yield coefficients ( $\theta_3, \theta_4$ ). The true values are  $[0.31, 0.18, 0.55, 0.05]$ , and there is no distribution information regarding the parameters but there are upper and lower bounds:

$$\begin{aligned} 0.05 &\leq \theta_{1,2,3} \leq 0.98 \\ 0.01 &\leq \theta_4 \leq 0.98 \end{aligned}$$



This classical model has been successfully used to perform D-optimal [3], E-optimal [32], SV-optimal [32] and mini-max robust-optimal [4] experiment design.

The total batch time is 40 hours and initially four switching points are chosen for both control inputs (dilution factor  $u_1$  and substrate concentration in feed  $u_2$ ):  $t_{sw} = 5, 13, 21, 29 \text{ hour}$ .

The design variables include sampling time ( $t_s$ ), initial concentration for biomass ( $x_0$ ) and substrate ( $s_0$ ), and control input magnitude for  $u_1$  and  $u_2$ . Assume both biomass and substrate concentrations can be measured online and the process noise is  $N(0, 0.04)$ . Ten samples are taken from each batch and any neighboring two samples should be taken at least 1 hour apart and at most 20 hours.  $u_1$  is bounded by 0.05 and  $0.2 \text{ h}^{-1}$  while  $u_2$  lies between 5 and  $35 \text{ g} \cdot \text{l}^{-1}$ . The initial biomass concentration ranges from 1 to  $10 \text{ g} \cdot \text{l}^{-1}$  and initial substrate concentration is always  $0.1 \text{ g} \cdot \text{l}^{-1}$ . All of these constraints can be summarized by following inequalities:

$$\begin{aligned}
0 &\leq t_{s,1}, \dots, t_{s,10} \leq 40 \quad \text{hours} \\
1 &\leq t_{s,i+1} - t_{s,i} \leq 20 \quad \text{hours}; i = 1, 2, \dots, 9 \\
0.05 &\leq u_{1,j} \leq 0.2 \quad \text{h}^{-1}; j = 1, 2, \dots, 5 \\
5 &\leq u_{2,j} \leq 35 \quad \text{g} \cdot \text{l}^{-1}; k = 1, 2, \dots, 5 \\
1 &\leq x_0 \leq 10 \\
s_0 &= 0.1
\end{aligned}$$

The initial design variables are chosen as:

$$\begin{aligned}
\phi(0) : \\
t_s &= 2, 6, 10, \dots, 38 \quad \text{hours} \\
u_{1,j} &= 0.12 \quad h^{-1}; j = 1, 2, \dots, 5 \\
u_{2,j} &= 15 \quad g \cdot l^{-1}; k = 1, 2, \dots, 5 \\
x_0 &= 5.5 \\
s_0 &= 0.1
\end{aligned}$$

Two different initial parameter estimates are used to test the robustness of the optimal design criteria:  $\theta_I = [0.5, 0.5, 0.5, 0.5]$ ;  $\theta_{II} = [0.527, 0.054, 0.935, 0.015]$ .  $\theta_I$  represents the case when no a priori information is available so the guess is in the middle of the region.  $\theta_{II}$  is the case when the initial information is inaccurate (up to a 70% error from the true value) [32].

### Case I

Figure 2 shows the DOE results using different criteria. For the P-optimal criteria, the initial information matrix ( $M_0$ ) is:

$$M_0 = \begin{bmatrix} 1450 & -189.5 & -153.5 & -165.3 \\ -189.5 & 27.8 & 24.9 & 216.1 \\ -153.5 & 24.9 & 29.8 & 164.7 \\ -165.3 & 216.1 & 164.7 & 1903.5 \end{bmatrix}$$

The corresponding eigenvalue and eigenvector matrices are:

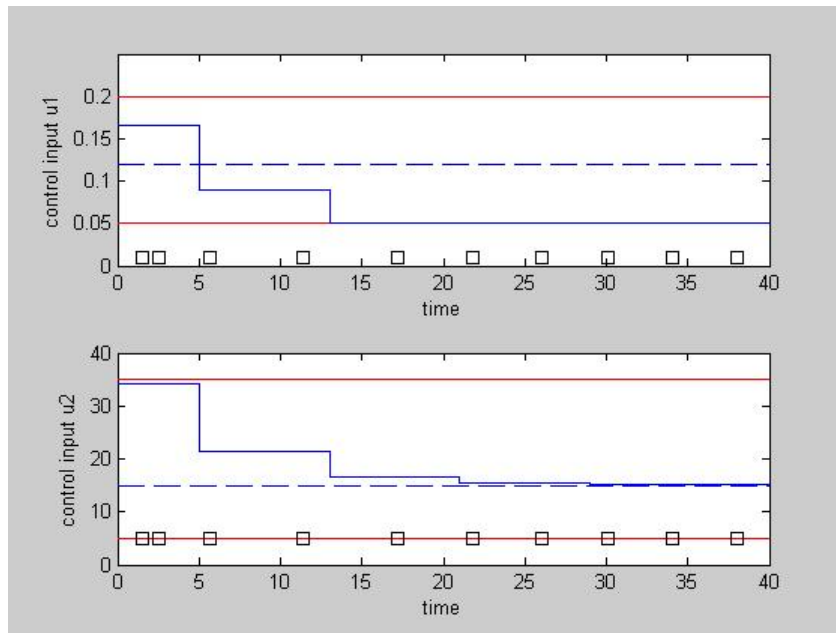
$$\Lambda_0 = \text{diag}(3384.7, 21.0, 5.3, 0.1)$$

$$P_0 = \begin{bmatrix} -0.655 & -0.333 & -0.638 & 0.234 \\ 0.086 & 0.266 & -0.522 & -0.801 \\ 0.067 & 0.814 & -0.317 & 0.482 \\ 0.749 & -0.394 & -0.469 & 0.253 \end{bmatrix}$$

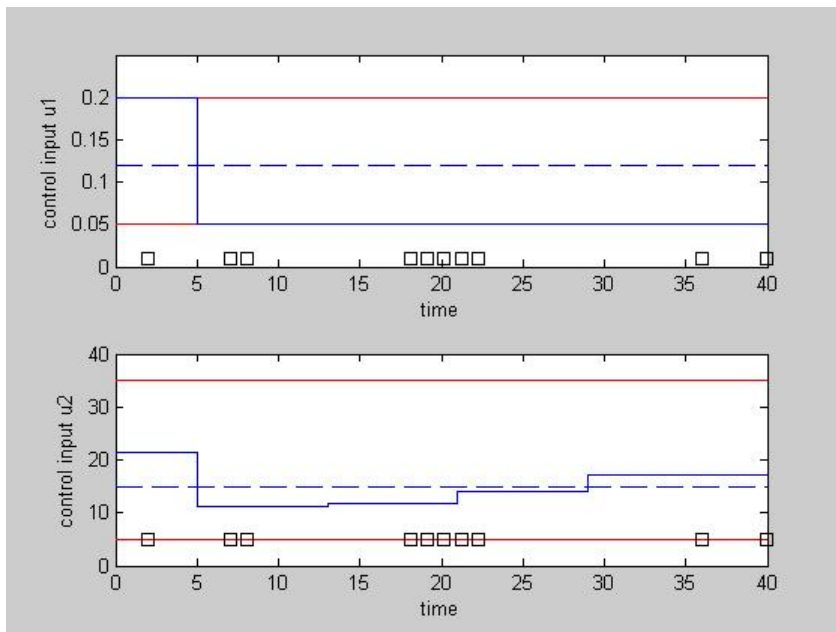
According to CPV, the first eigenvalue of  $M$  captures 99% of the variance and the second to fourth eigenvalues ( $i = 2, 3, 4$ ) need to be included in the objective function if all four parameters are to be estimated ( $j = 1, 2, 3, 4$ ). From  $P$  we can see the values are more or less equally distributed in each column, which means no eigenvector focuses on increasing the accuracy of a single parameter. As a result, the objective function for P-optimal is:

$$F = \min_{\phi \in \Phi} \left( \prod_{i=2}^4 (b_i \cdot \sum_{j=1}^4 P_{ji}^2) \right) = \max_{\phi \in \Phi} \left( \prod_{i=2}^4 \left( \frac{1}{b_i} \right) \right) = \max_{\phi \in \Phi} \left( \prod_{i=2}^4 (\lambda_i) \right) \quad (5.27)$$

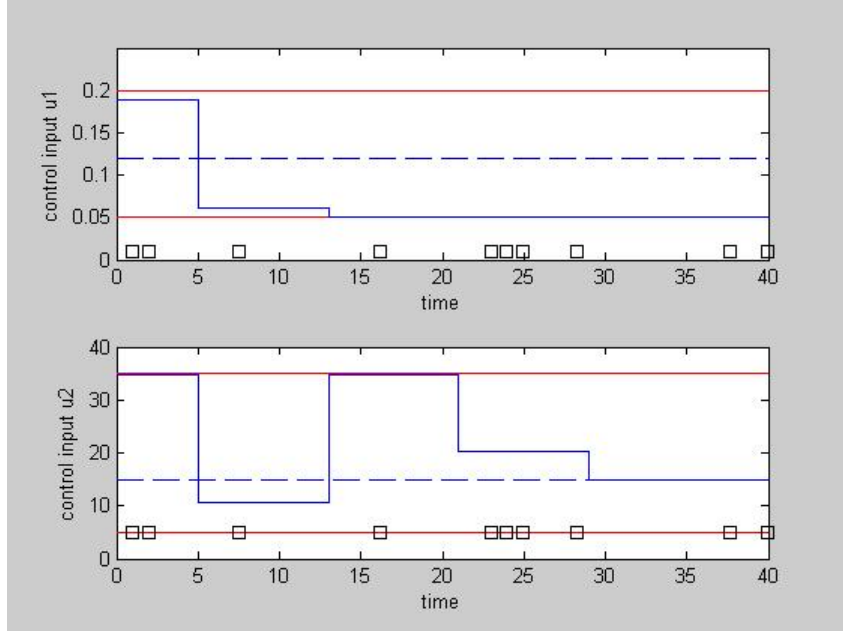
Figure 5.2 shows the DOE results according to different optimal design criteria. We note that the three criteria lead to three different experimental approaches. For feed rate  $u_1$ , E-optimal and P-optimal suggest bang-bang control while D-optimal has more gradual step changes before reaching the lower control limits. For substrate concentration in the feed ( $u_2$ ), the three methods give rather different results. Table 5.2 lists two important values from DOE:  $\Lambda$  and  $P$  are the eigenvalue and eigenvector matrices of information matrix ( $M$ ), respectively. One can see D-optimal results lead to a much larger  $\Lambda(1,1)$  and the product of all the eigenvalues is  $4.29 \times 10^8$  (volume of the parameter covariance matrix is  $0.23 \times 10^{-8}$ ). E-optimal focuses on increasing the value of the smallest eigenvalue which is 2.4, larger than the other two methods. According to P-optimal, the new method optimizes the product of the last three eigenvalues, which is  $1.07 \times 10^4$ . This is larger than the other two methods



(a) Control inputs designed by D-optimal



(b) Control inputs designed by E-optimal



(c) Control inputs designed by P-optimal

Figure 5.2: Control inputs and sampling points calculated by different optimal design criterion. Red: control limits; Blue: designed control inputs; Dash: initial control guesses; Square: sampling points

( $4.65 \times 10^3$  for D-optimal;  $8.74 \times 10^2$  for E-optimal), so in this sense D-optimal is better than E-optimal.

In Section 5.3, the drawbacks of D- and E- optimal methods have been discussed and in order to verify these assumptions, Table 5.3 lists the parameter estimation results by carrying out the corresponding designed experiments. *t-test* [52] indicates if the results show a reliable estimation:

$$t = \frac{\hat{\theta}}{\sigma_{\hat{\theta}}} \in t_{\alpha}(n - m) \quad (5.28)$$

where  $\alpha$  is the confidence value (0.95, 0.99);  $\sigma_{\hat{\theta}}$  is the standard deviation of the estimation which can be calculated by Eq. 5.5;  $n$  and  $m$  are the number of individual

Optimal Criteria	$\Lambda(\phi)$	$P(\phi)$
D-optimal	$\begin{bmatrix} 92378 & & 0 \\ & 3660 & \\ & & 10.5 \\ 0 & & & 1.21 \end{bmatrix}$	$\begin{bmatrix} -0.694 & 0.318 & -0.589 & -0.264 \\ 0.061 & -0.158 & -0.529 & 0.831 \\ 0.059 & -0.849 & -0.353 & -0.390 \\ 0.715 & 0.392 & -0.498 & -0.294 \end{bmatrix}$
E-optimal	$\begin{bmatrix} 8179.6 & & 0 \\ & 145.7 & \\ & & 2.5 \\ 0 & & & 2.4 \end{bmatrix}$	$\begin{bmatrix} -0.658 & -0.376 & -0.491 & 0.428 \\ 0.109 & 0.021 & -0.741 & -0.662 \\ 0.075 & 0.825 & -0.353 & 0.434 \\ 0.740 & -0.421 & -0.292 & 0.435 \end{bmatrix}$
P-optimal	$\begin{bmatrix} 10607 & & 0 \\ & 8880 & \\ & & 9.67 \\ 0 & & & 1.25 \end{bmatrix}$	$\begin{bmatrix} -0.66 & 0.412 & 0.361 & 0.506 \\ 0.082 & -0.481 & 0.873 & -0.478 \\ 0.042 & -0.809 & 0.244 & 0.533 \\ 0.739 & 0.415 & 0.218 & 0.481 \end{bmatrix}$

Table 5.2: DOE Results by Different Criteria for Case I

observations and number of parameters in the model. If the  $t$  value is larger than the threshold value ( $t_\alpha(n - m) = t_{0.95}(10 - 4) = 1.94$ ), the parameter is said to be reliable, otherwise it is statistically unacceptable. According to Table 5.3, the P-optimal method estimation results are the closest to the true value (which is known in this simulation study). The correlation coefficient matrix has a similar result for all three methods,  $\langle \theta_1, \theta_2 \rangle$  and  $\langle \theta_3, \theta_4 \rangle$  are highly correlated. In order to answer why  $\theta_2$  cannot be estimated with high accuracy, we examine the elements of  $P$  in Table 5.2. The second row represents  $\theta_2$  and for all three cases, the third and fourth columns of the row have larger values compared to the other two columns. The third and fourth columns correspond to the small eigenvalues in  $M$  matrix, which agrees with our PCA analysis (small eigenvalue in  $M$  indicates a larger parameter estimation variance).

In real applications, as  $\theta_2$  is not statistically acceptable, the P-optimal method allows designing an experiment to increase its estimation accuracy only. From analyzing  $\Lambda$  in Table 5.2, again the first eigenvalue is dominant and the remaining three

Optimal Criteria	$\hat{\theta}$	t-test	Parameter Correlation			
D-optimal	$0.2922 \pm 0.0089$	32.8	1.000	0.986	0.099	-0.0026
	$0.1485 \pm 0.1693$	0.877*	0.986	1.000	-0.018	-0.118
	$0.5340 \pm 0.0010$	53.4	0.099	-0.018	1.000	0.919
	$0.0464 \pm 0.0010$	116.0	-0.0026	-0.118	0.919	1.000
E-optimal	$0.2599 \pm 0.0282$	9.2	1.000	0.972	0.189	0.046
	$0.0883 \pm 0.1959$	0.45*	0.972	1.000	0.036	-0.102
	$0.5618 \pm 0.0055$	102.2	0.189	0.036	1.000	0.947
	$0.0526 \pm 0.0002$	263.0	0.046	-0.102	0.947	1.000
P-optimal	$0.3070 \pm 0.0085$	36.2	1.000	0.935	0.353	0.270
	$0.1997 \pm 0.2447$	0.816*	0.935	1.000	0.059	-0.021
	$0.5546 \pm 0.0042$	132.1	0.353	0.059	1.000	0.964
	$0.0516 \pm 0.0001$	516.0	0.270	-0.021	0.964	1.000
True	0.310 0.180 0.550 0.050					

Table 5.3: Parameter Estimation Results by Carrying Out Designed Experiments for Case I

eigenvalues should be included in the new objective function:

$$F = \min_{\phi \in \Phi} \left( \prod_{i=2}^4 (b_i \cdot P_{2i}^2) \right) = \max_{\phi \in \Phi} \left( \prod_{i=2}^4 (\lambda_i \cdot P_{2i}^{-2}) \right) \quad (5.29)$$

and  $b_i$  and  $p_i$  are eigenvalues and eigenvectors of:

$$V(\hat{\theta}, \phi) = \left[ \sum_{r=1}^q \sum_{s=1}^q v_{rs}^{-1} J_r J_s + V_0^{-1} \right]^{-1}$$

$V_0 = M_0^{-1}$  is taken from the last design in Table 5.3 because  $V_0$  is the a priori information obtained from earlier estimations. Figure 5.3 shows the control inputs and sampling times suggested by the sequential design.

The sequential estimation results are:

$$\hat{\theta} = [0.321 \pm 0.0312 \quad 0.1776 \pm 0.0924 \quad 0.582 \pm 0.0459 \quad 0.0571 \pm 0.0188]^T$$

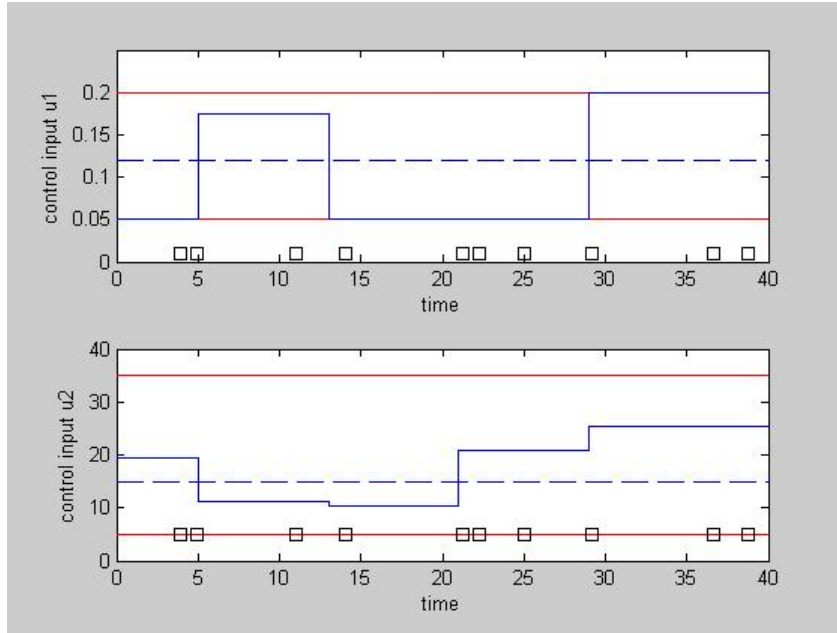


Figure 5.3: Control inputs and sampling points calculated by sequential P-optimal design criterion. Red: control limits; Blue: designed control inputs; Dash: initial control guesses; Square: sampling points

The corresponding  $t$ -test value is  $[29.48 \quad 5.51 \quad 36.327 \quad 8.670]$  and the reference value is 1.75. One can see that all the parameters converged to the true values and are statistically reliable according to the  $t$ -test. Moreover, the estimation accuracy for  $\theta_2$  is noticeably enhanced ( $t$  value changes from 0.816 to 5.51) and verifies our design goal, which is increased estimation accuracy of  $\theta_2$ .

## Case II

In this case, only D- and P- optimal criteria are considered and E-optimal is omitted because of its poorer performance in Case I. Figure 5.4 is the control strategy and corresponding sampling points suggested by DOE. Both methods suggest bang-bang control for  $u_1$  and the only difference is that the switching times are different. For  $u_2$ , both criteria also give similar results, which are composed of small set point



changes. Tables 5.4 and 5.5 listed the designed information matrix results and nonlinear regression results, respectively. As in Case I, D-optimal leads to a large leading eigenvalue ( $1.099 \times 10^5$ ) and the P-optimal criterion focuses on improving the product of the remaining three eigenvalues. Table 5.5 indicates both methods give similar regression results while the P-optimal method is slightly better for  $\theta_2$  estimation, because D-optimal gives a lower bound (0.05).

Because the initial guess is different from Case I, the information matrix ( $M$  in Table 5.4) is larger (in terms of volume) than in Table 5.2; the estimation results are worse by comparing Table 5.5 with Table 5.3. In Case I,  $P$  in Table 5.2 clearly shows  $\theta_2$  may not be accurately estimated. However, one examines  $P$  in Table 5.4, it is reasonable to assume that  $\theta_2$  can be estimated with high accuracy because  $P(2, 2)$  is dominant in the second row of  $P$  for both D- and P-optimal criteria. The results in Table 5.5 again indicate  $\theta_2$  is not statistically acceptable (using the  $t$ -test). The apparent contradiction can be explained by the fact that the initial parameter guesses are poor. Using the estimated parameter value of P-optimal criterion in Table 5.5 (0.3072, 0.0913, 0.5571, 0.0521) and the designed experimental condition (Figure 5.4(b)), the corresponding  $P$  matrix is:

$$\tilde{P} = \begin{bmatrix} -0.7427 & 0.5989 & -0.2972 & 0.0375 \\ 0.0372 & -0.0276 & -0.0227 & 0.9987 \\ 0.0529 & -0.3884 & -0.9193 & -0.0336 \\ 0.6665 & 0.6998 & -0.2569 & -0.0114 \end{bmatrix}$$

As the estimated parameter is closer to the true value compared with the initial guess, the corresponding  $\tilde{P}$  is more reasonable. One can see  $\tilde{P}(2, 4)$  is dominant in the second row, which indicates poor estimation accuracy as we saw in Case I. Moreover, when one analyzes  $\tilde{P}$  column by column,  $\tilde{P}(2, 4)$  is dominant in column

4 with the other three elements very close to zero, which means only  $\theta_2$  needs to be estimated with better accuracy. As a result, based on  $\tilde{P}$  matrix above, it is reasonable to design another experiment for  $\theta_2$ . According to CPV, the last three eigenvalues are included and the objective function is the same as Case I (Eq. 5.27).

Optimal Criteria	$\Lambda(\phi)$	$P(\phi)$
D-optimal	$\begin{bmatrix} 109900 & & & 0 \\ & 15264 & & \\ & & 54563 & \\ 0 & & & 1564 \end{bmatrix}$	$\begin{bmatrix} -0.057 & 0.392 & 0.905 & 0.152 \\ 0.017 & -0.913 & 0.405 & -0.051 \\ -0.090 & -0.112 & -0.122 & 0.982 \\ 0.994 & 0.028 & 0.034 & 0.099 \end{bmatrix}$
P-optimal	$\begin{bmatrix} 72456 & & & 0 \\ & 30722 & & \\ & & 40048 & \\ 0 & & & 3114 \end{bmatrix}$	$\begin{bmatrix} -0.082 & 0.176 & 0.970 & 0.146 \\ 0.136 & -0.972 & 0.191 & -0.021 \\ -0.082 & -0.061 & -0.144 & 0.984 \\ 0.984 & -0.144 & 0.042 & 0.097 \end{bmatrix}$

Table 5.4: DOE Results by Different Criteria for Case II

Optimal Criteria	$\hat{\theta}$	t-test	Parameter Correlation			
D-optimal	$0.2985 \pm 0.0130$	22.7	1.000	0.694	0.543	0.521
	$0.0500 \pm 0.1980$	0.252*	0.694	1.000	-0.196	-0.237
	$0.5310 \pm 0.0310$	17.48	0.543	-0.196	1.000	0.937
	$0.0467 \pm 0.0103$	5.86	0.521	-0.237	0.937	1.000
P-optimal	$0.3072 \pm 0.0130$	21.56	1.000	0.605	0.613	0.614
	$0.0913 \pm 0.2281$	0.399*	0.605	1.000	-0.216	-0.248
	$0.5571 \pm 0.0332$	15.89	0.613	-0.216	1.000	0.935
	$0.0521 \pm 0.0101$	5.01	0.614	-0.248	0.935	1.000
True		0.310	0.180	0.550	0.050	

Table 5.5: Parameter Estimation Results by Carrying Out Designed Experiments for Case II

Figure 5.5 gives the sequential design suggested by both methods, and the parameter estimation results are listed in Table 5.6. Because the P-optimal method is focusing on design of an experiment for estimating  $\theta_2$ , the corresponding results have a more accurate value and a tighter confidence region compared with D-optimal results. Meanwhile, D-optimal has a tighter confidence region for the other three

parameters because it has no preference in choosing which parameter to improve accuracy. *t* – *test* results suggest all parameters are statistically acceptable after two sequential designs.

Optimal Criteria	$\hat{\theta}$	t-test	Parameter Correlation			
D-optimal	$0.3060 \pm 0.0060$	47.15	1.000	0.260	0.835	0.830
	$0.2035 \pm 0.0567$	3.59	0.260	1.000	-0.253	-0.278
	$0.5317 \pm 0.0210$	24.71	0.835	-0.253	1.000	0.964
	$0.0463 \pm 0.0053$	9.29	0.830	-0.278	0.964	1.000
P-optimal	$0.3137 \pm 0.0071$	42.45	1.000	-0.065	0.891	0.865
	$0.1801 \pm 0.0520$	3.46	-0.065	1.000	-0.425	-0.531
	$0.5613 \pm 0.0261$	21.83	0.891	-0.425	1.000	0.955
	$0.0538 \pm 0.0071$	7.52	0.865	-0.531	0.955	1.000
True		0.310	0.180	0.550	0.050	

Table 5.6: Parameter Estimation Results by Sequential DOE for Case II

#### 5.4.2 Polymerization Reaction Model

Polyethylene is the most popular synthetic polymer with over 40 billion tons production every year [33]. Among different production techniques, gas phase reaction over Ziegler-Natta catalysts enjoys the advantage of moderate reaction conditions and simple downstream separation. Therefore a large proportion of polyethylene is produced in this way and a key operating constraint of this reactor is to keep the reactor temperature above the dew point of the reactant to avoid condensation and below the melting point of the polymer to prevent particle melting and agglomeration. At the same time, the system is prone to unstable steady states, limit cycles, and excursions toward unacceptably high temperature steady states [64]. Due to process understanding, optimization, and automation needs, there have been many efforts on first principles modeling during the last decade [64, 38, 65]. In this work, a gas phase polymerization reactor model [64] with seven ordinary differential equations (ODEs) and eight algebraic equations (AEs) is applied. It has been demonstrated that the model can reproduce the process features listed above and

has been widely applied in controller design and process optimization.

Figure 5.6 shows the schematic of the gas phase polymerization process [33, 64]. A feed including ethylene, comonomer, hydrogen, inerts, and catalysts are continuously added to the system and catalyst is also continuously added through another feed. Unreacted gas is cooled by a counter current heat exchanger. The ODEs and AEs used for this dynamic model are listed in Eq. 5.30 and 5.31, respectively. Detailed model parameters can be found in Appendix II.

$$\frac{d[In]}{dt} = \frac{F_{in} - \frac{[In]}{[M_1] + [In]} b_t}{V_g} \quad (5.30a)$$

$$\frac{d[M_1]}{dt} = \frac{F_{M_1} - \frac{[M_1]}{[M_1] + [In]} b_t - R_{M_1}}{V_g} \quad (5.30b)$$

$$\frac{dY_1}{dt} = F_c a_{c_1} - k_{d_1} Y_1 - \frac{R_{M_1} M_{W_1} Y_1}{B_W} \quad (5.30c)$$

$$\frac{dY_2}{dt} = F_c a_{c_2} - k_{d_2} Y_2 - \frac{R_{M_1} M_{W_1} Y_2}{B_W} \quad (5.30d)$$

$$\frac{dT}{dt} = \frac{H_f + H_{g_1} - H_{g_0} - H_r - H_{pol}}{M_r C_{pr} + B_W C_{ppol}} \quad (5.30e)$$

$$\frac{dT_{w_1}}{dt} = \frac{F_w}{M_w} (T_{w_i} - T_{w_1}) - \frac{U A}{M_w C_{pw}} (T_{w_1} - T_{g_1}) \quad (5.30f)$$

$$\frac{dT_{g_1}}{dt} = \frac{F_g}{M_g} (T - T_{g_1}) - \frac{U A}{M_g C_{pg}} (T_{w_1} - T_{g_1}) \quad (5.30g)$$

where

$$b_t = V_p C_v \sqrt{([M_1] + [In]) \cdot R R \cdot T - P_v} \quad (5.31a)$$

$$R_{M_1} = [M_1] \cdot k_{p0} \cdot \exp \left[ \frac{-E_a}{R} \left( \frac{1}{T} - \frac{1}{T_f} \right) \right] \cdot (Y_1 + Y_2) \quad (5.31b)$$

$$C_{pg} = \frac{[M_1]}{[M_1] + [In]} C_{pml} + \frac{[In]}{[M_1] + [In]} C_{pIn} \quad (5.31c)$$

$$H_f = F_{M_1} C_{pml} (T_{feed} - T_f) + F_{in} C_{pIn} (T_{feed} - T_f) \quad (5.31d)$$

$$H_{g1} = F_g(T_{g1} - T_f) \cdot C_{pg} \quad (5.31e)$$

$$H_{g0} = (F_g + b_t)(T - T_f) \cdot C_{pg} \quad (5.31f)$$

$$H_r = H_{reac}M_{W1}R_{M1} \quad (5.31g)$$

$$H_{pol} = C_{ppol}(T - T_f)M_{W1}R_{M1} \quad (5.31h)$$

In the original paper [64], the two types of active site have the same characteristics. In this work, two active sites have different concentrations:  $a_{c1} = 0.548 \text{ mol/kg}$  and  $a_{c2} = 0.750 \text{ mol/kg}$  and deactivation rates:  $k_{d1} = 0.36 \text{ h}^{-1}$  and  $k_{d2} = 0.72 \text{ s}^{-1}$ . All the other model parameters are kept the same as in [33, 64] (refer to Appendix II).

Suppose most thermodynamic properties are known and reaction-related kinetic parameters are unknown (Table 5.7). This can be mapped into the case where a new catalyst is introduced to the production system. The only unknown thermodynamic property is assumed to be the heat capacity of polymer ( $C_{ppol}$ ). Table 5.7 lists the true value of the parameters and the initial guesses of each parameter. One can see the parameters are distributed widely across many magnitudes (0 to  $10^5$ ), which lead to parameter estimation difficulty.

Model Parameters	$a_{c1}$ <i>mol/kg</i>	$a_{c2}$ <i>mol/kg</i>	$k_{p0}$ $\text{m}^3/(\text{mol} \cdot \text{h})$	$E_a$ <i>J/mol</i>	$k_{d1}$ $\text{h}^{-1}$	$k_{d2}$ $\text{h}^{-1}$	$C_{ppol}$ <i>J/(kg · K)</i>
True Value	0.548	0.750	0.306	$3.768 \times 10^4$	0.36	0.72	3559
Initial Knowledge	[0,1]	[0,1]	[0,1]	$[10^3, 10^6]$	[0,1]	[0,1]	$[10^3, 10^5]$
Initial Guess	0.5	0.5	0.5	$0.5 \times 10^6$	0.5	0.5	$5 \times 10^4$

Table 5.7: Unknown Parameters in Polymerization Model

Due to safety considerations, one can only perturb the catalyst flow rate ( $u_1$ ,  $F_c^{ss} = 5.8 \text{ kg/h}$ ) and feeding temperature ( $u_2$ ,  $T_{feed}^{ss} = 293 \text{ K}$ ) within a tight range ( $\pm 10\%$  around steady state value). All the state variables ([In], [M1],  $Y_1$ ,  $Y_2$ , T,  $T_{w1}$ , and  $T_{g1}$ ) are measurable (temperature related variables can be read on-

line and concentrations are analyzed offline) and the corresponding measurement noise are:  $N(0, [4 \ 6.7 \ 0.02 \ 0.02 \ 0.4 \ 0.4 \ 0.1])$ . Twenty samples are analyzed during one hundred hours experimental time and each two neighboring samples are taken longer than two hours and shorter than twenty hours. Four switches are scheduled for each control variable ( $u_1$  and  $u_2$ ) and the switching times are under certain constraints shown below. To eliminate the initial transient phase in simulation, the DOE phase is carried out from 100 to 200 hours:

$$\begin{aligned}
100 &\leq t_{s,i} \leq 200 \quad \text{hours}; \quad i = 1, 2, \dots, 20 \\
2 &\leq t_{s,i+1} - t_{s,i} \leq 20 \quad \text{hours}; \quad i = 1, 2, \dots, 19 \\
5.22 &\leq u_{1,j} \leq 6.38 \quad \text{kg} \times \text{h}^{-1}; \quad j = 1, 2, \dots, 5 \\
5 &\leq t_{u_1,j+1} - t_{u_1,j} \leq 30 \quad \text{hours}; \quad j = 1, 2, \dots, 4 \\
263.7 &\leq u_{2,k} \leq 322.3 \quad \text{K}; \quad k = 1, 2, \dots, 5 \\
5 &\leq t_{u_2,k+1} - t_{u_2,k} \leq 30 \quad \text{hours}; \quad k = 1, 2, \dots, 4
\end{aligned}$$

Pseudo Random Binary Sequence (PRBS) signal is widely used for model identification in both academia and industry. Suppose a PRBS signal is applied to this system with switching time equal to ten hours and constant sampling rates (every five hours), the designed variable will be:

$$\begin{aligned}
\phi_{PRBS} : \\
t_s &= 105, 110, \dots, 200 \quad \text{hours} \\
u_1 &= 5.22, 6.38, 6.38, 5.22, 5.22, 5.22, 6.38, 6.38, 6.38, 6.38 \quad \text{kg} \times \text{h}^{-1} \\
u_2 &= 263.7, 322.3, 322.3, 263.7, 322.3, 322.3, 322.3, 263.7, 263.7, 322.3 \quad \text{K} \\
t_{u_1} &= 105, 115, 125, 135, 145, 155, 165, 175, 185 \quad \text{hours} \\
t_{u_2} &= 110, 120, 130, 140, 150, 160, 170, 180, 190 \quad \text{hours}
\end{aligned}$$

With this PRBS input and equal sampling rates, the eigenvalue and eigenvector matrices are:

$$\Lambda_{PRBS} = \text{diag}(6.0 \times 10^7 \quad 2.8 \times 10^5 \quad 9.2 \times 10^3 \quad 2.2 \times 10^3 \dots \\ 3.4 \times 10^1 \quad 1.6 \times 10^1 \quad 4.5 \times 10^{-3})$$

$$P_{PRBS} = \begin{bmatrix} -0.3810 & -0.0280 & -0.0816 & -0.1169 & -0.5941 & 0.5777 & 0.3834 \\ -0.3810 & -0.0280 & -0.0816 & -0.1169 & 0.5941 & 0.5777 & -0.3834 \\ -0.7621 & -0.0558 & -0.1654 & -0.2372 & 0.0000 & -0.5767 & 0.0000 \\ -0.0878 & 0.9738 & 0.1932 & -0.0820 & 0.0000 & 0.0001 & 0.0000 \\ 0.2461 & 0.0471 & -0.3606 & -0.5543 & 0.3834 & 0.0015 & 0.5941 \\ 0.2461 & 0.0471 & -0.3606 & -0.5543 & -0.3834 & 0.0015 & -0.5941 \\ 0.0076 & -0.2065 & 0.8136 & -0.5434 & -0.0000 & 0.0000 & -0.0000 \end{bmatrix}$$

One can see that the first eigenvalue is dominant and the corresponding eigenvector focuses on explain  $k_{p0}$  ( $P_{13} = -0.76$ ) and  $a_c s$  and  $k_d s$  are also included. Furthermore, the second one is dominated by  $Ea$  related terms ( $P_{24} = 0.97$ ) and the third by polymer heat capacity ( $C_{ppol}$ ). Because E-optimal performance was unsatisfactory in Section 5.1, only D- and P- optimal are used here and the objective functions are:

*D-optimal:*

$$F_1 = \min_{\phi \in \Phi} \left( \prod_{i=1}^7 b_i \right) = \max_{\phi \in \Phi} \left( \prod_{i=1}^7 \lambda_i \right) \quad (5.32)$$

*P-optimal (all parameters):*

$$F_2 = \min_{\phi \in \Phi} \left( \prod_{i=2}^7 \left( b_i \cdot \sum_{j=1}^7 P_{ji}^2 \right) \right) = \max_{\phi \in \Phi} \left( \prod_{i=2}^7 \lambda_i \right) \quad (5.33)$$

*P-optimal* ( $E_a$ ):

$$F_1 = \min_{\phi \in \Phi} \left( \prod_{i=2}^7 (b_i \cdot P_{4i}^2) \right) = \max_{\phi \in \Phi} \left( \prod_{i=2}^7 (\lambda_i \cdot P_{4i}^{-2}) \right) \quad (5.34)$$

Two cases are considered in P-optimal,  $F_2$  aims to improve the precision of all the parameters while  $F_3$  only focuses on activation energy ( $E_a$ ) because activation energy is directly related to optimum operation temperature choice.

Figure 5.7 compares the suggested control set point changes for catalyst flow rate ( $u_1$ ) and feed temperature ( $u_2$ ) and the sampling times ( $t_s$ ) for different optimum criteria. It is obvious that D-optimal and P-optimal for all parameters (Figure 5.7(a) and (b)) give similar suggestions on sampling time. Furthermore, during the frequent sampling phase ( $100 < t < 135h$ ), catalyst flow rate set points in both cases are very similar while P-optimal suggests a two-step movements for feed temperature and D-optimal suggests bang-bang control. The reason for this small difference could be due to the continuous operation mode. Compared with batch operation, a continuous process loses some degrees of freedom for DOE such as initial condition and time dependency. This is also one of the reasons why parameter identification is difficult for continuous processes. For case (c), the calculated trajectories differ from cases (a) and (b), the suggested sampling times are roughly divided into two parts:  $110 - 135h$  and  $190 - 200h$ , respectively. The control modes are bang-bang control as well. Interestingly, the frequent sampling time periods are roughly under maximum operation limits. The difference between (c) and the previous two cases can be attributed to the significantly different objective functions because the  $P$  matrix is employed to focus on estimate activation energy better.

In order to testify the DOE results, simulation is carried out with measurement noise indicated above. The parameter estimation results for PRBS test, D-optimal,



and P-optimal criteria are listed in Table 5.8. Same as Table 5.3, the estimated parameter value ( $\hat{\theta}$ ),  $t$  test and estimated parameter correlation matrix are shown. It can be seen that the PRBS test gave the worst estimation results with large confidence regions for nearly all the parameters. As indicated by  $\phi_{PRBS}$  above, there are nine switching points compared with four for the designed runs but PRBS still gives unsatisfactory results.

For designed experiments, P-optimal for all parameters indicate a better overall parameter estimation result than D-optimal with closer to true values and tighter confidence intervals for almost all parameters. Furthermore, the parameter correlation matrix suggests some very strong linear dependency in D-optimal results as  $(\theta_1, \theta_2), (\theta_1, \theta_5), (\theta_1, \theta_6), (\theta_5, \theta_6), (\theta_4, \theta_7)$ , which indicates unreliable results. These correlated parameters can be replaced by a single parameter for each pair and still give a good fit to the D-optimal designed experimental data. This is not surprising for a continuous process because it is regulated as close to the steady state as possible, hence it does not provide much excitation. But this will surely increase the difficulty for parameter identification. Mathematically, the reason of highly correlated parameters could be attributed to ill-conditioned information matrix ( $M$ ) because the correlation matrix is proportional to the inverse of the  $M$  matrix. D-optimal maximized the dominant eigenvalue, which led to a large condition number  $5 \times 10^9$ . In comparison, P-optimal case shows well-decoupled results, which means the two-step movement between 100 to 140  $h$  for feed temperature in Figure 5.7(b) increases the nonlinearity of the response, which is helpful for identification. The condition number reduces to  $9 \times 10^6$ .

At the mean time, P-optimal design for activation energy only gives the best estimation for  $E_a$  with the tightest confidence region, although other parameters do not converge as close to the true value as P-optimal does for all parameters (Eq.

5.33). It is worthwhile to notice that for this continuous example, few estimation cases meet the  $t - test$  requirement (1.77). Again, a continuous process with fixed operation condition caused difficulty for DOE and parameter estimation. However, by using P-optimal for one important parameter only,  $E_a$  estimation meets the criterion ( $2.01 > 1.77$ ) and an optimum operation temperature can be defined based on this estimate for later controller design.

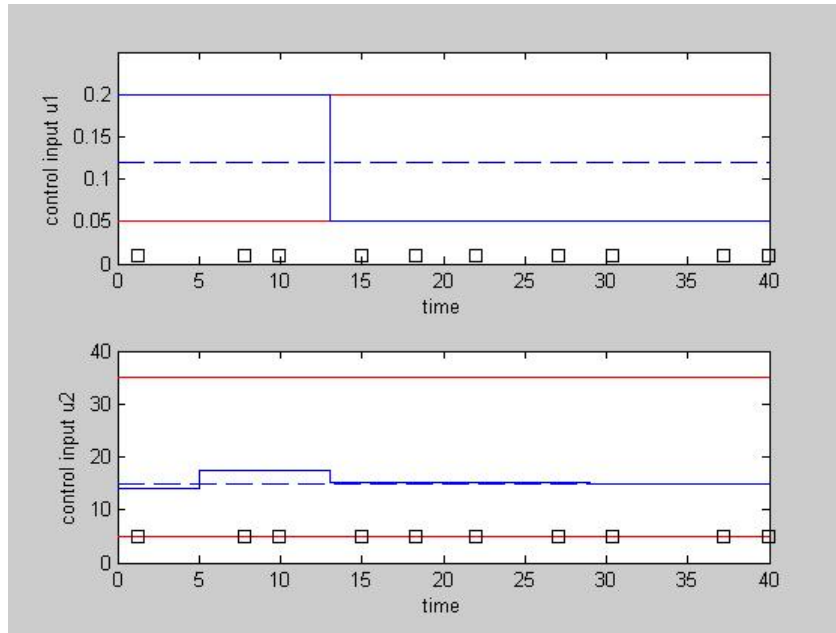
	$a_{c1}$	$a_{c2}$	$k_{p0}$	$E_a$	$k_{d1}$	$k_{d2}$	$C_{ppol}$
PRBS							
$\hat{\theta}$	0.6478	0.5249	0.3509	42670	0.5719	0.5260	5840
$\sigma_{\hat{\theta}}$	3.7562	3.5924	0.9840	200200	3.7744	4.1985	18710
$t - test(t_{ref} = 1.77)$	0.1725	0.1461	0.3566	0.2131	0.1515	0.1253	0.3121
Parameter Correlation	1.0000	-0.3111	-0.5028	-0.3488	0.4970	-0.3828	-0.3482
	-0.3111	1.0000	-0.6611	0.4444	-0.6692	0.5503	0.3731
	-0.5028	-0.6611	1.0000	-0.1029	0.1825	-0.1485	0.0018
	-0.3488	0.4444	-0.1029	1.0000	-0.7483	0.7345	0.6626
	0.4970	-0.6692	0.1825	-0.7483	1.0000	-0.9718	-0.6219
	-0.3828	0.5503	-0.1485	0.7345	-0.9718	1.0000	0.7051
	-0.3482	0.3731	0.0018	0.6626	-0.6219	0.7051	1.0000
D-optimal							
$\hat{\theta}$	0.6588	0.5540	0.3080	46300	0.4831	0.4709	5549
$\sigma_{\hat{\theta}}$	0.6757	0.7283	0.1720	103000	0.5263	0.6717	8600
$t - test(t_{ref} = 1.77)$	0.9750	0.7607	1.7907	0.4495	0.9179	0.7011	0.6452
Parameter Correlation	1.0000	-0.9900	-0.0717	-0.0178	0.9950	-0.9950	0.0127
	-0.9900	1.0000	-0.0695	0.0189	-0.9950	0.9950	-0.0105
	-0.0717	-0.0695	1.0000	-0.0002	-0.0011	0.0010	-0.0004
	-0.0178	0.0189	-0.0002	1.0000	-0.0179	0.0190	0.8456
	0.9950	-0.9950	-0.0011	-0.0179	1.0000	-1.0000	0.0128
	-0.9950	0.9950	0.0010	0.0190	-1.0000	1.0000	-0.0106

	0.0127	-0.0105	-0.0004	0.8456	0.0128	-0.0106	1.0000
P-optimal (All Parameters)							
$\hat{\theta}$	0.6455	0.6482	0.3035	40100	0.4675	0.5779	5076
$\sigma_{\hat{\theta}}$	0.6189	0.6858	0.1765	90060	0.3947	0.5351	9480
$t - test(t_{ref} = 1.77)$	1.0430	0.9452	1.7195	0.4453	1.1844	1.0800	0.5354
Parameter Correlation	1.0000	-0.3321	-0.5653	-0.3128	0.5966	-0.5142	-0.3320
	-0.3321	1.0000	-0.5870	0.3222	-0.6333	0.5538	0.3544
	-0.5653	-0.5870	1.0000	-0.0014	0.0051	0.0044	-0.0011
	-0.3128	0.3222	-0.0014	1.0000	-0.5589	0.5365	0.5421
	0.5966	-0.6333	0.0051	-0.5589	1.0000	-0.9854	-0.5950
	-0.5142	0.5538	0.0044	0.5365	-0.9854	1.0000	0.5925
	-0.3320	0.3544	-0.0011	0.5421	-0.5950	0.5925	1.0000
P-optimal (Activation Energy Only)							
$\hat{\theta}$	0.7255	0.5635	0.3053	38190	0.5594	0.4769	5532
$\sigma_{\hat{\theta}}$	0.7488	0.6786	0.1757	18100	0.5883	0.5928	9060
$t - test(t_{ref} = 1.77)$	0.9741	0.8304	1.7376	2.1099	0.9509	0.8045	0.6106
Parameter Correlation	1.0000	-0.4507	-0.5246	-0.1447	0.6745	-0.7043	0.1719
	-0.4507	1.0000	-0.5177	0.0900	-0.6308	0.6842	-0.0193
	-0.5246	-0.5177	1.0000	0.0012	-0.0034	-0.0026	-0.0007
	-0.1447	0.0900	0.0012	1.0000	-0.2086	0.1360	-0.4300
	0.6745	-0.6308	-0.0034	-0.2086	1.0000	-0.9868	0.2477
	-0.7043	0.6842	-0.0026	0.1360	-0.9868	1.0000	-0.1379
	0.1719	-0.0913	-0.0007	-0.4300	0.2477	-0.1379	1.0000
True Value							
$\theta$	0.5480	0.7500	0.3060	37680	0.3600	0.7200	3559

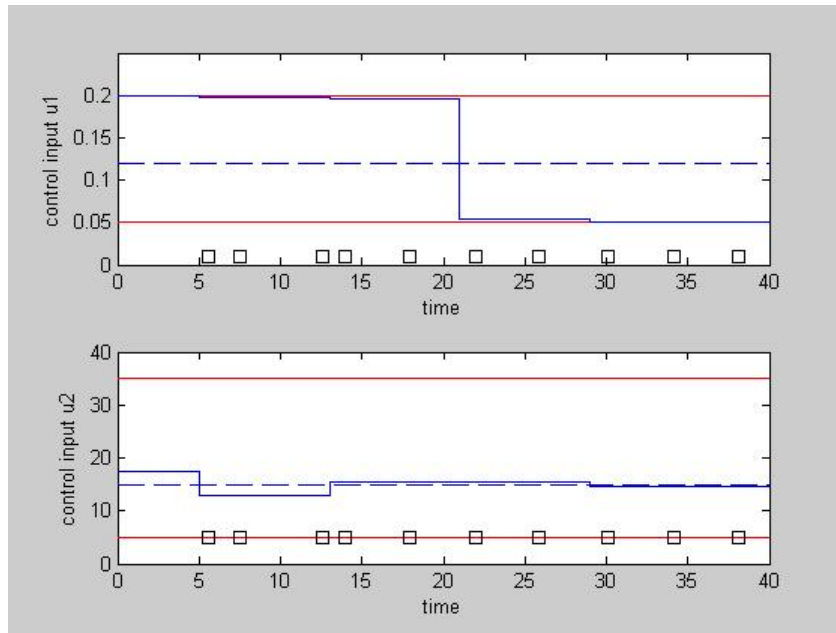
Table 5.8: Parameter Estimation Results for Polymerization Process

## 5.5 Summary

A generalized model-based DOE criterion is proposed in this chapter by introducing PCA into the sensitivity matrix ( $J$ ) and information matrix ( $M$ ) analysis. The main advantages of this method include easy rescaling of a large DAE system into small components. In addition, it is easy to focus on improving a specific subset of parameters, and parameters can be grouped according to their sensitivity behavior. Two engineering examples are presented to show the effectiveness of the new proposed method. According to the analysis, it can be concluded that P-optimal is more efficient for complex systems than the traditional D-, or E- optimal criteria, and it is robust to different initial parameter estimates. It may be a good idea to combine this identification objective function with optimal control objective function to achieve simultaneous control and identification, which will be the next step in future work.

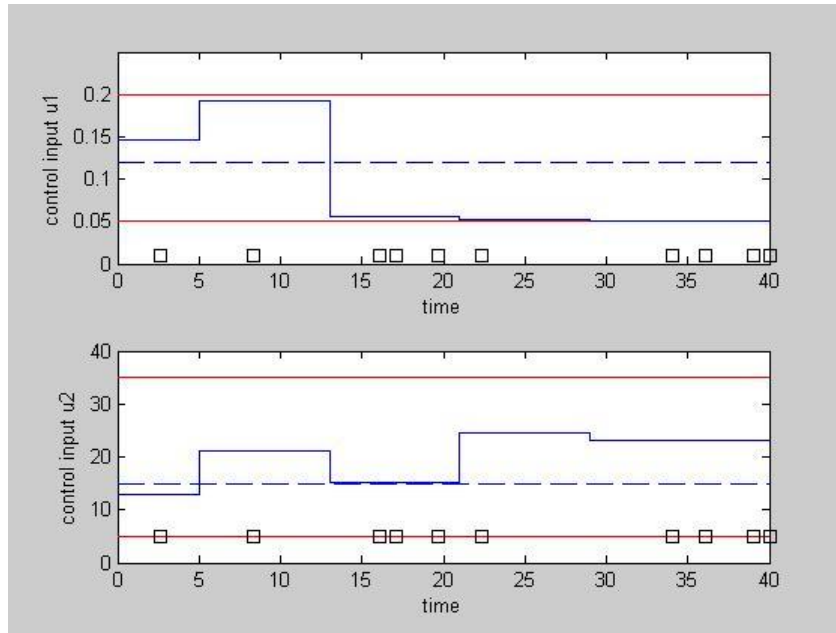


(a) Control inputs designed by D-optimal

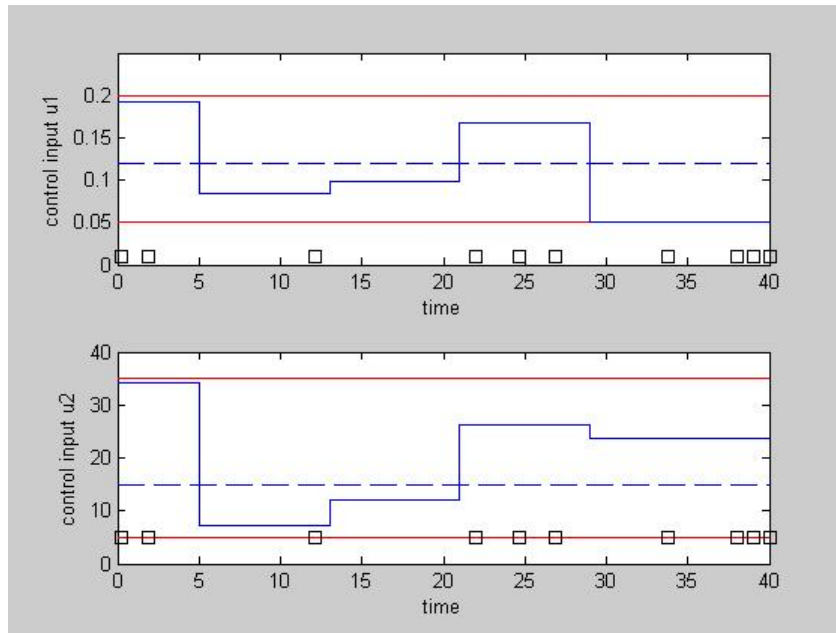


(b) Control inputs designed by P-optimal

Figure 5.4: Control inputs and sampling points calculated by different optimal design criterion. Red: control limits; Blue: designed control inputs; Dash: initial control guesses; Square: sampling points



(a) Control inputs designed by D-optimal



(b) Control inputs designed by P-optimal

Figure 5.5: Control inputs and sampling points calculated by different optimal design criterion. Red: control limits; Blue: designed control inputs; Dash: initial control guesses; Square: sampling points

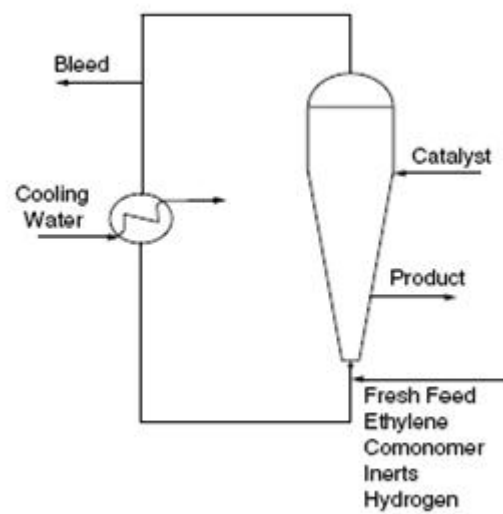
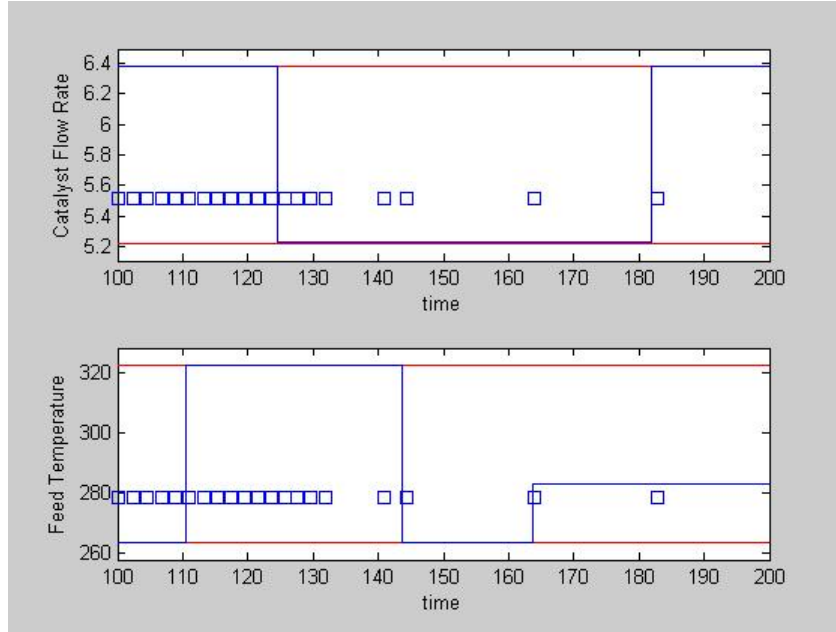
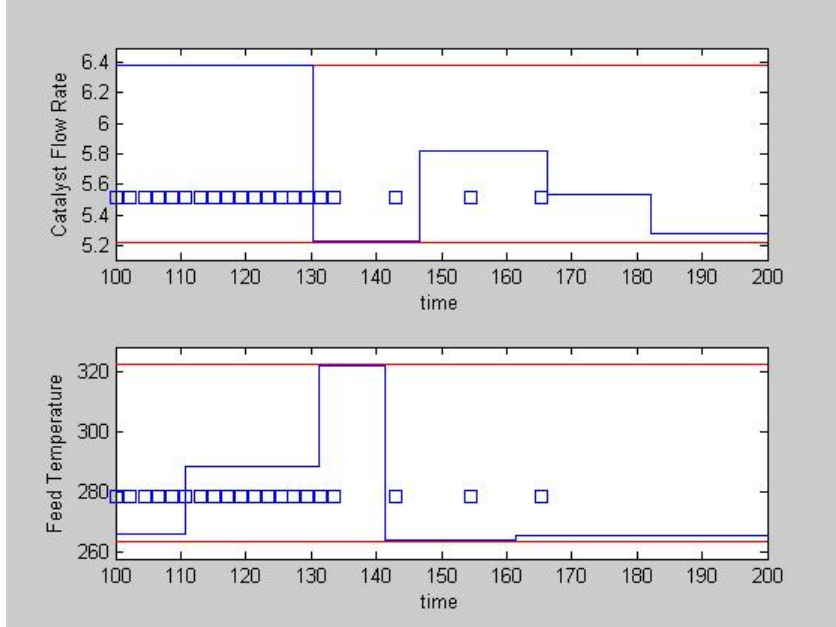


Figure 5.6: Gas phase polymerization system

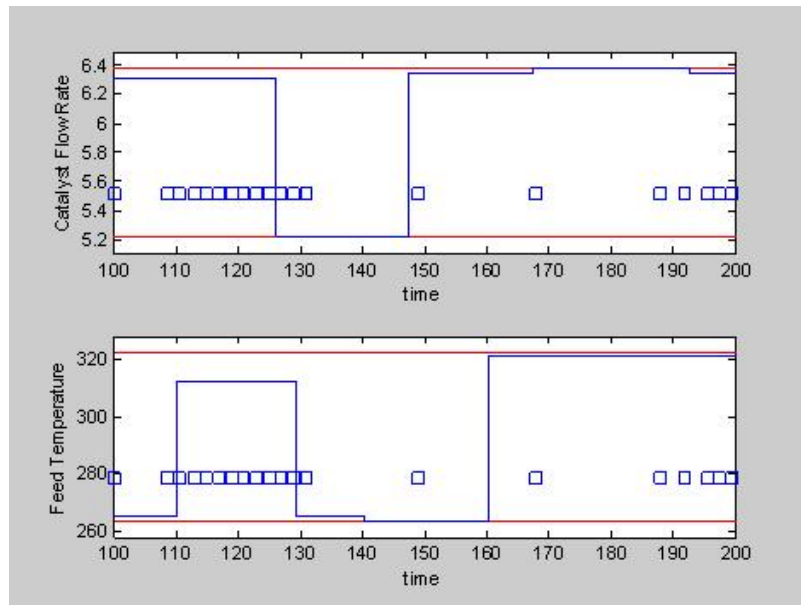


(a) Control inputs designed by D-optimal



(b) Control inputs designed by P-optimal for all parameters





(c) Control inputs designed by P-optimal for activation energy ( $E_a$ )

Figure 5.7: Control inputs and sampling points calculated by different optimal design criterion. Red: control limits; Blue: designed control inputs; Dash: initial control guesses; Square: sampling points

# Chapter 6

## Summary and Recommendations

### 6.1 Summary of Contributions

Mathematical modeling is a central activity in the chemical industry and many other process industries. The success of this activity relies on “people plus procedures plus tools - in that order of importance [16]”. This work mainly focuses on building and improving procedures and tools for the most widely used modeling approaches: statistical and first principles modeling. Meanwhile, most attention has been focused on batch processes because of its high nonlinear dynamics, non steady states and broad applications across many process industries.

One goal of this work is to propose and justify a complete and efficient statistical modeling algorithm package for online batch monitoring purpose. To achieve this goal, seven available commercialized software packages are evaluated and results suggest batch online application features are largely missing due to real time data sharing and robust online algorithm development difficulties. By focusing on

algorithm package development, a detailed Multiway PCA (MPCA) based batch monitoring procedure is given which serves as the starting point of Chapters 3 and 4.

In Chapter 3, current batch trajectory dynamic synchronization methods are compared and challenges are presented. Dynamic Time Warping (DTW) and Derivative DTW (DDTW) are the most suitable approaches so far but DTW creates many singularity points and DDTW fails when the process data are noise corrupted. A new dynamic optimization algorithm: Robust DDTW (RDDTW) is proposed by combining Savitzky-Golay filter with DDTW. By including SG-filter, the noisy data were fit into a  $m^{th}$  order polynomial then the numerical derivative is taken based on the polynomial rather than the noisy data. In this way, the precision of numerical derivative estimation is greatly enhanced. Furthermore, the weighting matrix for a given polynomial is calculated offline, thus the online synchronization part kept concise and efficient. Three case studies (NIR, dynamic simulation, and industrial batch data) further justify that the RDDTW algorithm is robust to process noise, filter parameter setting and also computationally efficient.

In Chapter 4, a new method to consider batch dynamics (time dependency) is proposed by employing EWMA into Hybrid-wise unfolding MPCA (HMPCA) routine. Theoretically, the new proposed E-HMPCA algorithm considers process dynamics by using a weighted low pass filter on score ( $t_k$ ) and residual vector ( $e_k$ ) and does not require large data matrix calculation as required by lagged window approach (Batch Dynamic PCA (BDPCA)). New control metrics ( $T^2$ ,  $SPE$ ) are proposed for the new methods. Three batch (polymer, chemical, and biochemical) processes are employed to compare HMPCA, BDPCA, and E-HMPCA. The results suggest E-HMPCA has following application advantages: 1. easy to implement online; 2. computationally efficient; 3. fast detection for small variations; 4. Small type I and

II errors.

A complete methodology is set up for batch process online monitoring. By collaborating with Emerson Process Management, the package is being developed and beta tested. Figures 6.1 and 6.2 contains snap shots of the beta test results. The package is working well in industrial process testing.

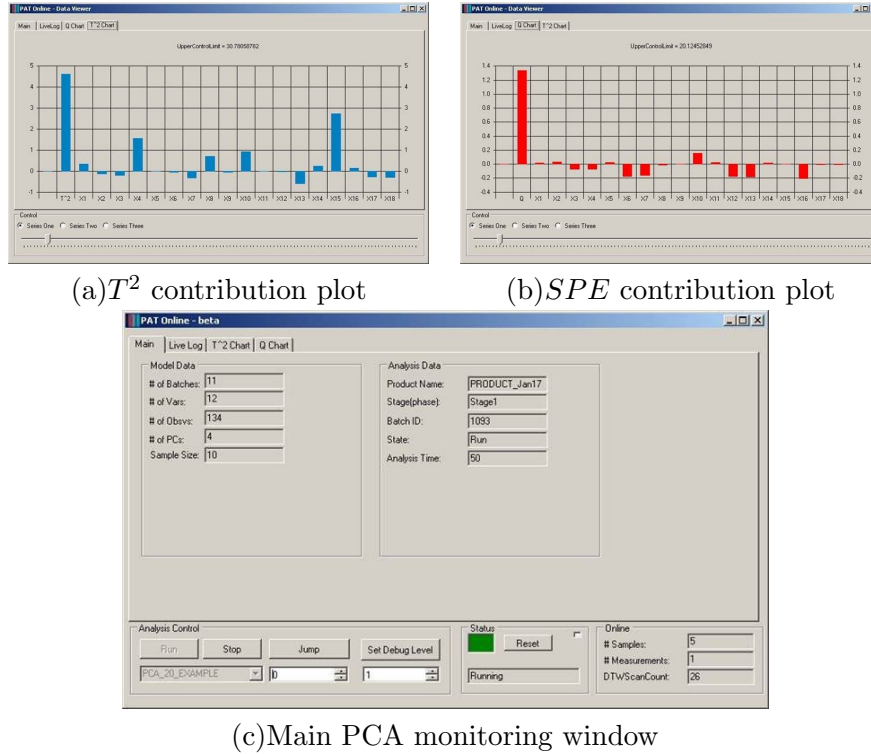
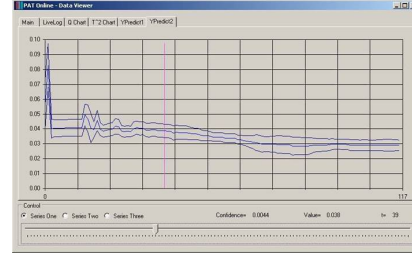
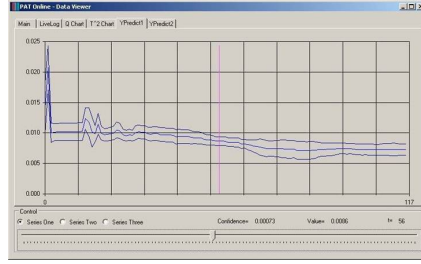
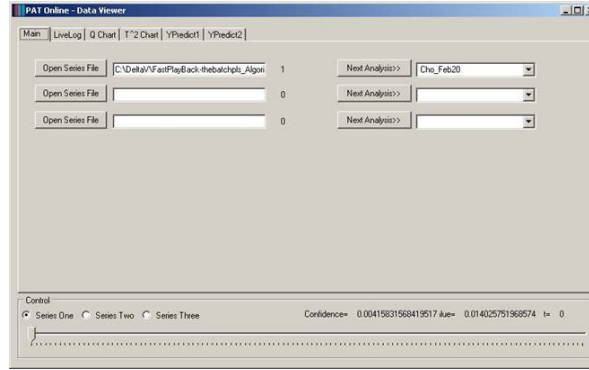


Figure 6.1: Interface for testing PCA model

In Chapter 5, a new dynamic Design of Experiments (DOE) (P-optimal) criteria is proposed for DAE system parameter estimation. The new criteria combines PCA with dynamic system sensitivity analysis technique. To implement the idea, nonlinear dynamic optimization is carried out by using numerical integration and nonlinear optimization. By using the P-optimal criteria, the information matrix



(a) End of batch estimation for V1 (b) End of batch estimation for V2



(c) Main PLS fast historical data playback window

Figure 6.2: Interface for testing PLS model

( $M$ ) can be elegantly divided into smaller compartments according to parameter sensitivity behavior. Furthermore, it is easy to design an experiment to improve the precision of a specific subset of parameters. Moreover, the P-optimal criteria may degenerate into D- or E-optimal criteria under certain conditions. One simple batch (four parameters, two ODEs) and one complex continuous (seven parameters, seven ODEs, eight AEs) process examples are applied to demonstrate the algorithm and idea.

## 6.2 Recommendations for Future Work

Despite the rapid advancements in first principles and statistical modeling, there still remain many areas need further research. For statistical modeling, the PCA-PLS type of generic tools can be applied to many processes efficiently. However,

these methods fail to include engineering insight and process knowledge. By taking engineering insights into account, the resulting model could be hybrid (statistical with first principles to describe mechanical relationship (e.g., mass balance)), therefore the detection and diagnosis efficiency should increase. Another approach to consider engineering insights is to introduce another artificial intelligence layer over PCA/PLS model. The upper layer can learn physical rules and evaluate control system overall performance to decide if and how the lower layer statistical model should be updated based on machine learning technique.

Although case studies in Chapter 3 indicate the RDDTW algorithm is superior over traditional dynamic synchronization methods, the algorithm may need further industrial robustness test especially for large transport delay and multiphase batch (e.g., loading, reaction, and separation compose a three phase batch) processes. For a large transport delay, it is crucial to set bandwidth constraint to a large value and for multiphase processes, it is better to perform alignment phase by phase.

On first principles modeling, the benefit by employing DOE is significant in terms of time and cost saving. However, great efforts are required for full industrial adoption. Theoretically, the challenges of applying DOE are mainly on numerical analysis, to be more specific: numerical integration, optimization and DAE system auto-differentiation. For each topic, there are good software packages, even open source packages are available. However, to make them robustly work together into a commercial software and meet industrial requirements is a very challenging task. Moreover, it is reasonable to combine DOE and optimal control objective together to achieve simultaneous control and identification. It can be a very difficult task but will result in a new closed-loop self-regulating control system which is mainly composed of Model Predictive Control (MPC), Moving Horizon Estimation (MHE), and DOE. The advantage of this new system could be fast model identification,

self regulation, and enhanced controller performance to process drift and product transition.

For other open issues in modeling applications, please refer to Table 3 (Suggestions for improving current modeling technology) in Foss et al. [29], and Section 4 (The future: enhancing the practice of industrial process modeling) in Cameron and Ingram [16].

After demonstrating the benefit of modeling all the time, the author wants to point out that although modeling is playing a center role in many applications, it is not the solution to everything. “Yet at the same time, “over-sell” of modeling capabilities remains a potential trap for advocates [16]”.

# Appendix A

## Nomenclature

*Abbreviation:*

AE	Algebraic Equation
ARX	Auto-Regression with eXogenous
BDPCA	Batch Dynamic PCA
COW	Correlated Optimized Warping
CPV	Cumulative Percent Variance
CUSUM	Cumulative Sum
DAE	Dynamic Algebraic Equation
DDTW	Derivative Dynamic Time Warping
DOE	Design Of Experiments
DPCA	Dynamic PCA
DTW	Dynamic Time Warping
E-HMPCA	EWMA combined HMPCA
EWMA	Exponentially Weighted Moving Average
FIR	Finite Impulse Response
HMPCA	Hybrid-wise unfolding Multiway PCA
ICA	Independent Component Analysis



LQR	Linear Quadratic Regulator
LS	Least Squares
MLE	Maximum Likelihood Estimation
MPCA	Multiway PCA
MSPC	Multivariate Statistical Process Control
NIPALS	Nonlinear Iterative Partial Least Squares
NLP	Nonlinear Programming
OLS	Ordinary Least Squares
PARAFAC	Parallel Factor Analysis
PCA	Principal Component Analysis
PDE	Partial Differential Equation
PID	Proportional, Integral, Derivative
PLS	Partial Least Squares
PRBS	Pseudo Random Binary Sequence
PRESS	Predicted Error Sum of Squares
RDDTW	Robust Derivative Dynamic Time Warping
SDE	Stochastic Differential Equation
SG filter	Savitzky Golay filter
SPC	Statistical Process Control
SVD	Singular Value Decompositoin
UCL	Upper Control Limit
<i>pdf</i>	Probability density function

*Mathematics Symbol:*

$A$	Stoichiometry information for fermentation problem
$B$	Eigenvalue matrix of $V$
$C$	Right eigenvector for SVD
$D$	Accumulated distance in synchronization

$E$	Residual matrix
$F$	$F$ distribution
$G$	Dilution rate for fermentation problem
$H$	DAE system coefficient matrix
$I$	Number of batch profile
$J$	Number of batch variables
$J$	Sensitivity matrix
$K$	Number of observations in a batch
$L$	Eigenvalue matrix for SVD
$M$	Information matrix
$N(\theta, \sigma^2)$	Normal distribution with mean $\theta$ and standard deviation $\sigma$
$P$	Loading matrix for PCA or PLS
$Q$	Loading matrix for PLS
$S$	Subspace
$T$	Score matrix for PCA
$U$	Score matrix for PLS
$V$	Estimated parameter covariance matrix
$W$	Weighting matrix in synchronization
$W$	Left eigenvector for SVD
$X$	Data matrix ( $n \times m$ )
$Y$	Data matrix ( $n \times l$ )
$X_{raw}$	Raw batch data
$\bar{X}$	Average trajectory of $X_{raw}$
$a_{nk}$	Polynomial coefficient in SG filter
$b$	Eigenvalues of $V$
$c$	Grid point in synchronization
$d$	Local distance in synchronization

$dx$	Derivative estimation of $x$ series
$e$	Number of parameters of less interests
$f(\theta, X)$	Nonlinear regression model
$f$	DAE model
$f_i$	Polynomial used in SG filter
$h$	Number of raw measurements for SG filter
$l$	Number of output in $Y$
$m$	Polynomial order in SG filter
$m$	Number of model parameters in DAE
$n$	Number of observations, experiments
$p$	Loading vector for PCA
$p$	Number of outputs for PLS
$q$	Loading vector for PLS
$q$	Number of equations in DAE system
$u$	Score vector for PLS
$u$	Controlled variables
$s$	Substrate concentration in fermentation model
$t$	Length of reference trajectory
$t$	Score vector for PCA
$t$	time for dynamic systems
$r$	Length of new trajectory
$x$	State variables
$x_{filter}$	Exponentially filtered trajectory
$x_{ref}$	Reference trajectory
$x_{new}$	New trajectory
$y$	Measured variables
<i>Greeks</i>	:

$K$	Number of points needed for the aligned path
$\Lambda$	Eigenvalue matrix
$\alpha$	Confidence level
$\chi^2$	$\chi^2$ distribution
$\eta$	Optimal aligned path
$\kappa$	Grid point index of the aligned path
$\gamma$	Equation index in synchronization
$\lambda$	Eigenvalue of covariance matrix
$\phi$	Design vector
$\sigma_\theta$	Confidence region of estimation
$\theta$	Model parameter vector
$\hat{\theta}$	Estimated parameter value

*Subscripts:*

0	Starting point or time; Initial knowledge
$E$	EWMA filtered
$f$	Final point or time
$i, j, k$	Index for score, loading matrices
$m$	Measured value
$p$	Principal subspace
$pc$	Principal component
$r$	Residual subspace
$s$	Sample
max	Upper bound
min	Lower bound
<i>mean</i>	Mean value of a vector
<i>std</i>	Standard deviation of a vector

# Appendix B

## Polymerization Reactor Model

Symbol	Parameter	Values	Units
$V_g$	Volume of gas phase in the reactor	500	$m^3$
$V_p$	Bleed stream valve position	0.5	
$P_v$	Pressure downstream of bleed vent	17	atm
$B_W$	Mass of polymer in the fluidized bed	$7 \times 10^4$	kg
$k_{p0}$	Pre-exponential factor for polymer propagation rate	$85 \times 10^{-3}$	$\frac{m^3}{mol \cdot s}$
$E_a$	Activation energy	$9000 \times 4.2$	J/mol
$C_{pml}$	Specific heat capacity of ethylene	$11 \times 4.2$	$\frac{J}{mol \cdot K}$
$C_v$	Vent flow coefficient	7.5	$atm^{-0.5} \cdot mol/s$
$C_{pw}$	Specific heat capacity of water	$10^3 \times 4.2$	$\frac{J}{kg \cdot K}$

$C_{pIn}$	Specific heat capacity of inert gas	$6.9 \times 4.2$	$\frac{J}{kg \cdot K}$
$C_{ppol}$	Specific heat capacity of polymer	$0.85 \times 10^3 \times 4.2$	$\frac{J}{kg \cdot K}$
$k_{d1}$	Deactivation rate constant for site 1	0.36	$h^{-1}$
$k_{d2}$	Deactivation rate constant for site 2	0.72	$h^{-1}$
$M_{W_1}$	Molecular weight of monomer	$28.05 \times 10^{-3}$	$\frac{kg}{mol}$
$M_w$	Mass holdup of cooling water in heat exchanger	$3.314 \times 10^4$	kg
$M_g$	Mass holdup of stream gas in heat exchanger	6060.5	mol
$M_r C_{pr}$	Product of mass and heat capacity of reactor wall	$1.4 \times 10^7 \times 4.2$	$\frac{J}{K}$
$H_{reac}$	Heat of reaction	$-894 \times 10^3 \times 4.2$	$\frac{J}{kg}$
$UA$	Product of heat exchanger coefficient with area	$1.14 \times 10^6 \times 4.2$	$\frac{J}{K \cdot s}$
$F_{In}$	Flow rate of inert gas	5	$\frac{mol}{s}$
$F_{M_1}$	Flow rate of ethylene	190	$\frac{mol}{s}$
$F_g$	Flow rate of recycle gas	8500	$\frac{mol}{s}$
$F_w$	Flow rate of cooling water	$3.11 \times 10^5 \times 18 \times 10^{-3}$	$\frac{kg}{s}$
$T_f$	Reference temperature	360	K
$RR$	Ideal gas constant	$8.206 \times 10^{-5}$	$\frac{m^3 \cdot atm}{mol \cdot K}$
$R$	Ideal gas constant	8.314	$\frac{J}{mol \cdot K}$
$a_{c1}$	Active site concentration of catalyst	0.548	$\frac{mol}{kg}$

$a_{c_2}$	Active site concentration of catalyst	0.750	$\frac{mol}{kg}$
-----------	---------------------------------------	-------	------------------

---

# Bibliography

- [1] Sarolta Albert and Robert D. Kinley. Multivariate statistical monitoring of batch processes: an industrial case study of fermentation supervision. *Trends in Biotechnology*, 19(2):53–62, 2001.
- [2] Ashraf AlGhazzawi and Barry Lennox. Monitoring a complex refining process using multivariate statistics. *Control Engineering Practice*, 16:294–307, 2008.
- [3] S. P. Asprey and S. Macchietto. Statistical tools for optimal dynamic model building. *Computers and Chemical Engineering*, 24:1261–1267, 2000.
- [4] S. P. Asprey and S. Macchietto. Designing robust optimal dynamic experiments. *Journal of Process Control*, 12:545–556, 2002.
- [5] S. P. Asprey and Yuji Naka. Mathematical problems in fitting kinetic models - some new perspectives. *Journal of Chemical Engineering of Japan*, 32(3):328–337, 1999.
- [6] Anthony C. Atkinson. Non-constant variance and the design of experiments for chemical kinetic models. In S. P. Asprey and S. Macchietto, editors, *Dynamic Model Development*, volume 16. Elsevier, 2003.
- [7] Anthony C. Atkinson and Barbara Bogacka. Compound , d- and ds-optimum designs for determining the order of a chemical reaction. *Technometrics*, 39:347–356, 1997.



- [8] Anthony C. Atkinson and Barbara Bogacka. Compound and other optimum designs for systems of nonlinear differential equations arising in chemical kinetics. *Chemometrics and Intelligent Laboratory Systems*, 61:17–33, 2002.
- [9] Anthony C. Atkinson and William G. Hunter. The design of experiments for parameter estimation. *Technometrics*, 10:271–289, 1968.
- [10] Bhavik R. Bakshi. Multiscale pca with application to multivariate statistical process monitoring. *AIChE Journal*, 44(7):1596–1610, 1998.
- [11] Yonathan Bard. *Nonlinear Parameter Estimation*. Academic Press, INC, New York, 1974.
- [12] Gulnur Birol, Cenk Undey, and Ali Cinar. A modular simulation package for fed-batch fermentation: Penicillin production. *Computers and Chemical Engineering*, 26:1553–1565, 2002.
- [13] Michael Boudreau and Gregory McMillan. *Multivariate Statistical Process Control*. ISA, 1st edition, 2006.
- [14] G. E. P. Box and H. L. Lucas. Design of experiments in non-linear situations. *Biometrika*, 46:77–90, 1959.
- [15] Rasmus Bro. Parafac, tutorial and applications. *Chemometrics and Intelligent Laboratory Systems*, 38:149–171, 1997.
- [16] I. T. Cameron and G. D. Ingram. A survey of industrial process modelling across the product and process lifecycle. *Computers and Chemical Engineering*, 32:420–438, 2008.
- [17] Kevin A. Chamness. *Multivariate Fault Detection and Visualization in the Semiconductor Industry*. PhD thesis, The University of Texas at Austin, 2006.

- [18] Jinghui Chen and Chien-Mao Liao. Dynamic process fault monitoring based on neural network and pca. *Journal of Process Control*, 12(2):277–289, 2002.
- [19] Junghui Chen, Chien-Mao Liao, Franz Ren Jen Lin, and Muh-Jung Lu. Principal component analysis based control charts with memory effect for process monitoring. *Ind. Eng. Chem. Res.*, 40:1516–1527, 2001.
- [20] Gregory A. Cherry. *Methods for improving the reliability of semiconductor fault detection and diagnosis with Principal Component Analysis*. PhD thesis, The University of Texas at Austin, 2006.
- [21] Leo H. Chiang, Riccardo Leardi, Randy J. Pell, and Mary Beth Seasholtz. Industrial experiences with multivariate statistical analysis of batch process data. *Chemometrics and Intelligent Laboratory Systems*, 81(2):109–119, 2006.
- [22] Leo H. Chiang, Evan L. Russell, and Richard D. Braatz. *Fault Detection and Diagnosis in Industrial Systems*. Springer, 1st edition, 2001.
- [23] B. Dayal and J. F. MacGregor. Improved pls algorithms. *Journal of Chemometrics*, 11:73–85, 1997.
- [24] Holger Dette and Weng Kee Wong. E-optimal designs for the michaelis-menten model. *Statistics & Probability Letters*, 44:405–408, 1999.
- [25] Dong Dong and Thomas J. McAvoy. Nonlinear principal component analysis-based on principal curves and neural networks. *Computers and Chemical Engineering*, 20(1):65–78, 1996.
- [26] T. F. Edgar. Computing practices of new engineers. *Control Engineering*, 51:8, 2004.
- [27] D. M. Espie and S. Macchietto. The optimal design of dynamic experiments. *AIChE Journal*, 35:223–229, 1989.

- [28] D. M. Espie and S. Macchietto. The optimal design of dynamic experiments. *AIChE Journal*, 35:223–229, 1989.
- [29] B. A. Foss, B. Lohmann, and W. Marquardt. A field study of the industrial modeling process. *Journal of Process Control*, 56:325–338, 1998.
- [30] Gaia Franceschini and Sandro Macchietto. Validation of a model for biodiesel production through model-based experiment design. *Ind. Eng. Chem. Res.*, 46:220–232, 2007.
- [31] Kapil G Gadkar, Rudiyanto Gunawan, and Francis Doyle III. Iterative approach to model identification of biological networks. *BMC Bioinformatics*, 6:155–174, 2005.
- [32] Federico Galvanin, S. Macchietto, and Fabrizio Bezzo. Model-based design of parallel experiments. *Ind. Eng. Chem. Res.*, 46:871–882, 2007.
- [33] Adiwinata Gani, Prashant Mhaskar, and Panagiotis D. Christofides. Fault tolerant control of a polyethylene reactor. *Journal of Process Control*, 17:439–451, 2007.
- [34] Paul Geladi and Bruce R. Kowalski. Partial least squares regression: A tutorial. *Analytica Chimica Acta*, 185:1–17, 1986.
- [35] G. C. Goodwin and R. L. Payne. *Dynamic Systems Identification: Experiment Design and Data Analysis*. Academic Press, New York, 1977.
- [36] Jon C. Gunther, Jeremy S. Conner, and Dale E. Seborg. Fault detection and diagnosis in an industrial fed-batch cell culture process. *Biotechnol. Prog.*, 23:851–857, 2007.
- [37] K. M. Hangos and I. T. Cameron. *Process Modeling and Model Analysis*. Academic Press, INC, London, 2001.

- [38] H. Hatzantonis, H. Yiannoulakis, A. Yiagopoulos, and C. Kiparissides. Recent developments in modeling gas-phase catalyzed olefin polymerization fluidized-bed reactors: The effect of bubble size variation on the reactor's performance. *Chemical Engineering Science*, 55:3237–3259, 2000.
- [39] David M. Himes, Robert H. Storer, and Christos Georgakis. Determination of the number of principal components for disturbance detection and isolation. In *American Control Conference*, pages 1279–1283, Baltimore, Maryland, 1994.
- [40] Agnar Høglundsson. Pls regression methods. *Journal of Chemometrics*, 2:211–228, 1988.
- [41] J. L. Horn. A rationale and test for the number of factors in factor analysis. *Psychometrika*, 30(2):73–77, 1965.
- [42] Sebastien Issanchou, Patrick Cognet, and Michel Cabassud. Sequential experimental design strategy for rapid kinetic modeling of chemical synthesis. *AIChE Journal*, 51:1773–1781, 2005.
- [43] J. Edward Jackson. *A User's Guide to Principal Components*. Wiley series in probability and statistics. Wiley, 2003.
- [44] J. Edward Jackson and Govind S. Mudholkar. Control procedures for residuals associated with principal component analysis. *Technometrics*, 21(3):341–349, 1979.
- [45] Ian T. Jolliffe. *Principal Component Analysis*. Springer Series in Statistics. Springer, 2nd edition, 2002.
- [46] Athanassios Kassidas, John F. MacGregor, and Paul A. Taylor. Synchronization of batch trajectories using dynamic time warping. *AIChE Journal*, 44(4):864–875, 1998.

- [47] Eamonn J. Keogh and Micheal J. Pazzani. Derivative dynamic time warping. In *First SIAM International Conference on Data Mining*, Chicago, Illionis, 2001.
- [48] Theodora Kourti. 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.
- [49] Theodora Kourti. Monitor. *Chemometrics and Intelligent Laboratory Systems*, 76:215–220, 2005.
- [50] Theodora Kourti, Jennifer Lee, and John F. MacGregor. Experiences with industrial applications of projection methods for multivariates statistical process control. *Computers and Chemical Engineering*, 20(S1):S745–S750, 1996.
- [51] J.V. Kresta, J. F. MacGregor, and T.E. Marlin. Multivariate statistical monitoring of process operating performance. *Canadian Journal of Chemical Engineering*, 69:35–47, 1991.
- [52] Niels Rode Kristensen, Henrik Madsen, and Sten Bay Jorgensen. An investigation of some tools for process model identification for prediction. In S. P. Asprey and S. Macchietto, editors, *Dynamic Model Development*, volume 16. Elsevier, 2003.
- [53] Wenfu Ku, Robert H. Storer, and Christos Georgakis. Disturbance detection and isolation by dynamic principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 30(1):179–196, 1995.
- [54] Jong-Min Lee. *Statistical Process Monitoring Based on Independent Component Analysis and Multivariate Statistical Methods*. PhD thesis, Pohang University of Science and Technology, 2004.

- [55] B. Lennox, G. A. Montague, H. G. Hiden, G. Kornfeld, and P. R. Goulding. Process monitoring of an industrial fed-batch fermentation. *Biotechnology and Bioengineering*, 74(2):125–135, 2001.
- [56] Weihua Li, H. Henry Yue, Sergio Valle-Cervantes, and S. Joe Qin. Recursive pca for adaptive process monitoring. *Journal of Process Control*, 10:471–486, 2000.
- [57] L. Ljung. *System Identification-Theory For the User*. Prentice Hall, Upper Saddle River, 2 edition, 1999.
- [58] J.A. Lopes, J. C. Menezes, J. A. Westerhuis, and A. K. Smilde. Multiblock pls analysis of an industrial pharmaceutical process. *Biotechnology and Bioengineering*, 80(4):419–427, 2002.
- [59] CA Lowry, WH Woodall, CW Champ, and SE Rigdon. A multivariate exponentially weighted moving average control chart. *Technometrics*, 34:46–53, 1992.
- [60] William L. Luyben. *Process modeling, simulation, and control for chemical engineers*. McGraw-Hill, 2 edition, 1989.
- [61] John F. MacGregor, Christiane Jaeckle, Costas Kiparissides, and M. Koutoudi. Process monitoring and diagnosis by multiblock pls methods. *AIChE Journal*, 40:826–838, 1994.
- [62] John F. MacGregor and Theodora Kourti. Statistial process control of multivariate processes. *Control Engineering Practice*, 3:403–414, 1995.
- [63] Robert L. Mason and John C. Young. *Multivariate Statistical Process Control with Industrial Applications*. Statistics and applied probability. ASA-SIAM, Philadelphia, 2002.

- [64] K. B. McAuley, D. A. Macdonald, and P. J. McLellan. Effects of operating conditions on stability of gas-phase polyethylene reactors. *AIChE Journal*, 41:868–879, 1995.
- [65] K. B. McAuley, J. F. MacGregor, and A. E. Hamielec. A kinetic model for industrial gas phase ethylene copolymerization. *AIChE Journal*, 36:837–850, 1990.
- [66] Ivan Miletic, Shannon Quinn, Michael Dudzic, Vit Vaculik, and Marc Champagne. An industrial perspective on implementing online applications of multivariate statistics. *Journal of Process Control*, 14:821–836, 2004.
- [67] N. P. V. Nielsen, J. M. Carstensen, and J. Smedsgaard. Aligning of single and multiple wavelength chromatographic profiles for chemometric data analysis using correlation optimized warping. *Journal of Chromatography A*, 805:17–35, 1998.
- [68] Paul Nomikos and John F. MacGregor. Monitoring of batch processes using multi-way principal component analysis. *AIChE Journal*, 40:1361–1375, 1994.
- [69] Paul Nomikos and John F. MacGregor. Multi-way partial least squares in monitoring batch process. *Chemometrics and Intelligent Laboratory Systems*, 30:97–108, 1995.
- [70] Paul Nomikos and John F. MacGregor. Multivariate spc charts for monitoring batch processes. *Technometrics*, 37:41–59, 1995.
- [71] R. K. Pearson. Selecting nonlinear model structures for computer control. *Journal of Process Control*, 13:1–26, 2003.
- [72] V. Pravdova, B. Walczak, and D. L. Massart. A comparison of two algorithms for warping of analytical signals. *Analytica Chimica Acta*, 456:77–92, 2002.

- [73] S. J. Qin. Recursive pls for adaptive data modeling. *Computers and Chemical Engineering*, 22:503–514, 1998.
- [74] S. J. Qin. *Multivariate Analysis, Monitoring, and Control of Processes Using Latent Variable Methods*. 2007.
- [75] S. J. Qin and Ricardo H. Dunia. Determining the number of principal components for best reconstruction. In *IFAC DYCOPS’98*, Greece, 1998.
- [76] S. Joe Qin. Statistical process monitoring: Basics and beyond. *Journal of Chemometrics*, 17:480–502, 2003.
- [77] Henk-Jan Ramaker, Eric N. M. van Sprang, John A. Westerhuis, and Age K. Smilde. Dynamic time warping of spectroscopic batch data. *Analytica Chimica Acta*, 498(1-2):133–153, 2003.
- [78] Y. Rotem, A. Wachs, and D. R. Lewin. Ethylene compressor monitoring using model-based pca. *AIChE Journal*, 46(9):1825–1836, 2000.
- [79] Abraham Savitzky and Marcel J.E. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36:1627–1639, 1964.
- [80] Dale E. Seborg, Thomas F. Edgar, and Duncan A. Mellichamp. *Process Dynamics and Control*. John Wiley and Sons, Inc, 2003.
- [81] Fabio R. Sidoli, Athanasios Mantalaris, and S. P. Asprey. Toward global parametric estimability of a large scale kinetic single cell model for mammalian cell cultures. *Ind. Eng. Chem. Res.*, 44:868–878, 2005.
- [82] Age K. Smilde. Three-way analysis. problems and prospects. *Chemometrics and Intelligent Laboratory Systems*, 15:143–157, 1992.



- [83] Eric N. M. van Sprang, Henk-Jan Ramaker, Johan A. Westerhuis, Stephen P. Gurden, and Age K. Smilde. Critical evaluation of approaches for online batch process monitoring. *Chemical Engineering Science*, 57:3979–3991, 2002.
- [84] Jean Steinier, Yves Termonia, and Jules Deltour. Smoothing and differentiation of data by simplified least square procedure. *Analytical Chemistry*, 44:1906–1909, 1972.
- [85] Giorgio Tomasi, Frans van den Berg, and Claus Andersson. Correlation optimized warping and dynamic time warping as preprocessing methods for chromatographic data. *Journal of Chemometrics*, 18(5):231–241, 2004.
- [86] Sergio Valle, Weihua Li, and S. Joe Qin. Selection of the number of principal components: The variance of the reconstruction error criterion with a comparison to other methods. *Industrial and Engineering Chemistry Research*, 38:4389–4401, 1999.
- [87] Sergio Valle-Cervantes. *Plant-wide Monitoring of Processes Under Closed-loop Control*. PhD thesis, University of Texas at Austin, 2001.
- [88] E. Walter and L. Pronzato. Qualitative and quantitative experiment design for phenomenological models - a survey. *Automatica*, 26:195–213, 1990.
- [89] Johan A. Westerhuis, Stephen P. Gurden, and Age K. Smilde. Generalized contribution plots in multivariate statistical process monitoring. *Chemometrics and Intelligent Laboratory Systems*, 51:95–114, 2000.
- [90] Johan A. Westerhuis, Theodora Kourti, and John F. MacGregor. Comparing alternative approaches for multivariate statistical analysis of batch process data. *Journal of Chemometrics*, 13:397–413, 1999.
- [91] H. Wold. *Estimation of principal component and related models by iterative least squares*. Multivariate Analysis. Academic Press, NY, 1966.

- [92] Svante Wold. Cross-validatory estimation of the number of components in factor and principal components models. *Technometrics*, 20(4):397–406, 1978.
- [93] Svante Wold. Exponentially weighted moving principal component analysis and projections to latent structures. *Chemometrics and Intelligent Laboratory Systems*, 23:149–161, 1994.
- [94] Svante Wold, Nouna Kettaneh, Hakan Friden, and Andrea Holmberg. Modelling and diagnostics of batch processes and analogous kinetic experiments. *Chemometrics and Intelligent Laboratory Systems*, 44:331–340, 1998.
- [95] Chang Kyoo Yoo and In-Beum Lee. Nonlinear multivariate filtering and bio-process monitoring for supervising nonlinear biological processes. *Process Biochemistry*, 41:1854–1863, 2006.
- [96] H. Henry Yue and S. Joe Qin. Reconstruction-based fault identification using a combined index. *Industrial and Engineering Chemistry Research*, 40:4403–4414, 2001.
- [97] Yang Zhang and Thomas F. Edgar. Bio-reactor monitoring with multiway pca and model based pca. In *AIChE Annual Meeting*, San Francisco, CA, USA, 2006.
- [98] Yang Zhang and Thomas F. Edgar. Multivariate statistical process control. In Michael Boudreau and Gregory McMillan, editors, *New Directions in Bio-process Modeling and Control*, volume Chapter 8. ISA, 2006.
- [99] Yang Zhang and Thomas F. Edgar. On-line batch process monitoring using modified dynamic batch pca. In *ACC*, pages 2551–2556, NY, 2007. IEEE.
- [100] L. Zullo. *Computer aided design of experiments. An engineering approach*. PhD thesis, University of London, 1991.

# Vita

Yang Zhang was born in Beijing, China on 16 October 1980, the son of Shanlin Zhang and Yulan Lin. He received Bachelor of Engineering and Master of Engineering degrees in Chemical Engineering from Tsinghua University in 2003 and 2005, respectively. In Fall 2005, he began Doctorate degree study under the direction of Thomas F. Edgar in the University of Texas at Austin.

Permanent Address: 1650 West 6th Street, Apt A.,  
Austin, Texas, 78703

This dissertation was typeset with  $\text{\LaTeX 2}_{\epsilon}$ <sup>1</sup> by the author.

---

<sup>1</sup> $\text{\LaTeX 2}_{\epsilon}$  is an extension of  $\text{\LaTeX}$ .  $\text{\LaTeX}$  is a collection of macros for  $\text{\TeX}$ .  $\text{\TeX}$  is a trademark of the American Mathematical Society. The macros used in formatting this dissertation were written by Dinesh Das, Department of Computer Sciences, The University of Texas at Austin, and extended by Bert Kay, James A. Bednar, and Ayman El-Khashab.