SYSTEM IDENTIFICATION AND TRAJECTORY TRACKING CONTROL OF BATCH PROCESSES USING LATENT VARIABLE MODELS

SYSTEM IDENTIFICATION AND TRAJECTORY TRACKING CONTROL OF BATCH PROCESSES USING LATENT VARIABLE MODELS

BY

MASOUD GOLSHAN, M.Sc., B.Sc.

A THESIS

SUBMITTED TO THE DEPARTMENT OF CHEMICAL ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES AT

MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

DOCTOR OF PHILOSOPHY (2010) McMaster University
(Chemical Engineering) Hamilton, Ontario

TITLE: System Identification and Trajectory Tracking Control of Batch Processes Using Latent Variable Models

AUTHOR: Masoud Golshan
M.Sc., Sharif University of Technology,
Tehran, Iran
B.Sc., Iran University of Science and Technology,
Tehran, Iran

SUPERVISORS: Dr. John F. MacGregor
Dr. Prashant Mhaskar

NUMBER OF PAGES: xxiv, 163

This Thesis is dedicated to my wonderful Friend, my beloved wife, and my beautiful lady, *Bahareh*

ABSTRACT

This thesis considers the problem of model identification and trajectory tracking control in batch processes. From the point of view of a control engineer, finite duration of the process, not operating around the equilibrium point, and time-varying operating conditions with nonlinear behavior are amongst the most significant differences between batch and continuous processes. The major contribution of this thesis is to develop an alternative to Nonlinear Model Predictive Control (NMPC) by incorporating Latent Variable Models (LVMs) in the course of an MPC algorithm (called LV-MPC).

Two control formulations are developed in this study: Control in the latent variable space and control in the original variable space. The algorithms are based on multi-phase PCA models developed on batch data arrays. In both cases prediction of the future trajectories is accomplished using statistical latent variable missing data imputation methods. It is shown that the two control formulations are complementary to each other. The control in the latent space is the infinite horizon LV-MPC, while the control in the original variable space is the finite horizon LV-MPC. The proposed LV-MPCs can handle constraints. The methods are tested on two simulated batch reactor case studies.

Furthermore, different latent variable modeling alternatives for modeling of batch processes are investigated from the view point of their application in the course of LV-MPC. Two modeling alternatives previously proposed in the literature are incorporated in the course of the LV-MPC methodology: Batch-Wise Unfolding (BWU), and Observation-Wise with Time-lag Unfolding (OWTU). The BWU modeling approach addresses the nonlinearity and time varying properties of the batch process. However, it

needs a large number of batch runs in the training dataset. The OWTU approach leads to a Linear Time Invariant (LTI) modeling of the process which captures the average process dynamics. However, it needs only 1-3 batch runs for building the process model which makes this approach attractive for practical situations. In addition, a new modeling approach is proposed in this study which tries to capture the major benefits of both BWU and OWTU while avoiding the drawbacks of each one. It is called the Regularized Batch-Wise Unfolding (RBWU) modeling approach. This modeling approach has the capability of modeling the nonlinearity and time-varying properties almost as accurately as BWU and at the same time it leads to a smoother PCA model and needs fewer numbers of observations for building the model as compared to BWU. The performances of the three latent variable modeling approaches in the course of LV-MPC for trajectory tracking are illustrated using two simulated batch reactor case studies. Recommendations are then given on which modeling approach to use under different scenarios.

In the last stage of this research, various issues on the closed-loop identification of empirical latent variable models for model predictive control (MPC) of batch processes are investigated. It is shown that in most cases, it is possible to identify the batch process models only from historical batches without the need for external excitation of the closed-loop system by dither signal on top of the controller output. The maximum requirement would be to use extra batch runs with different set-point trajectories in the training dataset. The issue of model bias in closed-loop identification is investigated and the desirable controller characteristics to be used in the data generation step to minimize this bias in the latent variable models are discussed.

ACKNOWLEDGEMENTS

I would like to express my deepest and most sincere gratitude to my supervisor, Dr. John F. MacGregor, whose excellent guidance, ceaseless encouragement, patience, and expertise supported me throughout the course of my research.

I am also deeply grateful to my co-supervisor, Dr. Prashant Mhaskar, whose kind helps, cooperation, and patience made my Ph.D. track smooth and enjoyable.

I would also like to thank my committee members, Dr. Shahin Sirouspour, and Dr. Benoît Chachuat, for their invaluable questions and suggestions that enhanced the quality of this research. Also, many thanks go to administrative staff, Nanci Cole, Lynn Falkiner, Kathy Goodram, and Andrea Vickers.

My sincere thanks to my friends at McMaster University and MACC specially Siamak, Emily, and Kevin, for making a friendly and warm working environment and for their invaluable discussions that helped me understand many concepts and overcome many difficulties during my studies at McMaster University.

I would also like to express my gratitude to my friends at ProSensus Inc. for cooperation with this research and for helpful discussions and suggestions that improved the quality of this thesis.

Warmest thanks go to my dear Bahareh for her endless love and supports and encouragements. I am also profoundly thankful of my parents and family from the early stages of learning up to the current time for their eternal helps and supports. My journey would not have reached here without their inspirations, encouragements and sacrifices.

PUBLICATION LIST

This thesis has been prepared in accordance with the regulations for a "Sandwich" thesis format or as a compilation of papers stipulated by the Faculty of Graduate Studies at McMaster University. Each chapter includes materials that have or will be published in a separate journal paper.

Chapter2:

Masoud Golshan, John F. MacGregor, Mark-John Bruwer, Prashant Mhaskar, 2010, Latent Variable Model Predictive Control (LV-MPC) for Trajectory Tracking in Batch Processes, Journal of Process Control 20, 538-550.

Contributions: Masoud Golshan developed the theoretical formulation of the control in the latent variable space and incorporation of different batch process modeling alternatives studied in this chapter into the LV-MPC in consultation with Dr. John F. MacGregor. Mathematical formulation of the control algorithm in the original variable space is proposed by Mark-John Bruwer. All the simulation studies related to the LV-MPC methodologies are performed by Masoud Golshan in consultation with Dr. John F. MacGregor. Formulation and simulation of NMPC is performed by Masoud Golshan in consultation with Dr. Prashant Mhaskar. The paper is written by Masoud Golshan and edited by Dr. John F. McGregor and Dr. Prashant Mhaskar.

Chapter3:

Masoud Golshan, John F. MacGregor, Prashant Mhaskar, Latent Variable Model Predictive Control for trajectory tracking in batch processes: Alternative modeling approaches, *Materials of this paper are published in two conference proceedings and this paper will be submitted for publication to the* ***Journal of Process Control***

Contributions: The study is performed by Masoud Golshan in Consultation with Dr. John F. MacGregor. The Paper is written by Masoud Golshan and Edited by John F. MacGregor and Prashant Mhaskar.

M. Golshan, J.F. MacGregor, M.J. Bruwer, and P. Mhaskar, "Latent Variable MPC for trajectory tracking in batch processes: role of model structure," *Proceedings of American Control Conference*, St. Louis, USA: July 2009, pp. 4779-4784.

M. Golshan and J.F. MacGregor, "Latent Variable Modeling of Batch Processes for Trajectory Tracking Control," *Proceedings of 9th international Symposium on Dynamics and Control of Process Systems*, Leuven, Belgium: July 2010, pp. 13-18.

Chapter4:

Masoud Golshan, John F. MacGregor, Prashant Mhaskar, Identification for Control of Batch Processes using Latent Variable Models, accepted for presentation at *AIChE annual meeting*, Salt Lake City, UT, November 7-12, 2010, *To be submitted for publications to the* ***Journal of Industrial & Engineering Chemistry Research***

Contributions: The study is performed by Masoud Golshan in Consultation with Dr. John F. MacGregor. The Paper is written by Masoud Golshan and Edited by John F. MacGregor and Prashant Mhaskar.

TABLE OF CONTENTS

ABSTRACT vi
ACKNOWLEDGEMENTS viii
PUBLICATION LIST ix
TABLE OF CONTENTS xii
LIST OF FIGURES xv
LIST OF TABLES xix
NOMENCLATURE xx

CHAPTER 1: Introduction 1

1.1. Batch Processes 1
1.2. Latent Variable Models (LVMs) 3
1.3. Model Predictive Control 5
1.4. Research Objectives and Scopes 8
1.5. Thesis Outline 9
References 11

CHAPTER 2 : Latent Variable Model Predictive Control (LV-MPC) for Trajectory Tracking in Batch Processes 13

2.1. Introduction 14
2.2. Latent variable modeling, identification and prediction 17
2.2.1. Structure of batch data 17
2.2.2. Latent Variable (LV) models 18
2.2.3. Identification 19
2.2.4. Prediction of future trajectories 20
2.3. Methodology 24
2.3.1. Multiphase LV models 24
2.3.2. Control 29

2.4. Case Study: Temperature Control in a Batch Reactor 37
2.4.1. Trajectory tracking 39
2.4.2. Disturbance Rejection 46
2.5. Conclusion 50
References 52

CHAPTER 3: Latent Variable Model Predictive Control for Trajectory Tracking in Batch Processes: Alternative Modeling Approaches 58

3.1. Introduction 59
3.2. Latent Variable Modeling of Batch Processes 60
3.2.1. Rearrangement of batch dataset 61
3.2.2 Comparison of BWU and OWTU approaches 63
3.2.3. Regularized Batch-Wise Unfolding (RBWU) 65
3.2.4. Multi-phase Modeling 69
3.2.5. Identification Experiments 71
3.3. Control Methodology 72
3.4. Simulation Studies 73
3.4.1. SISO Process 75
3.4.2. MIMO Process 82
3.5. Conclusion 90

References 92

CHAPTER 4: Identification for Control of Batch Processes using Latent Variable Models 96

4.1. Introduction 97
4.2. Closed-loop identifiability conditions in batch processes 101
4.2.1. Identifiability conditions for LTI systems operating in closed-loop 102
4.2.2. Connections between different identification approaches 105
4.2.3. Pseudo Identifiability conditions for batch processes operating in closed-loop 108

4.3. Design Of Experiment (DOE) to reduce the bias in the closed-loop identification by LVMs 116
4.4. Simulation studies 121
4.4.1. Identifiability tests 121
4.4.2. Investigation of controller characteristics 127
4.5. Conclusion 134
References 137

CHAPTER 5: Conclusions and Future Work 143
5.1. Conclusoins 143
5.2. Future Possible Directions of the Research 148
References 152
Appendix A: Derivation of Equation (2.18) 156
Appendix B: Controllability and Observability in the PCA Model 158
Appendix C: Process Model for the Temperature Control Case Study 162

LIST OF FIGURES

Fig 1.1: Illustration of a typical dataset to which PCA can be applied.....	4
Fig.1.2: An example of illustration of the MPC algorithm.....	6
Fig. 2.1: a) Cube of Batch process dataset, b) Batch-wise unfolding of the dataset.	18
Fig. 2.2: An observation (Z) containing already observed data (Z^*) and missing data ($Z^{\#}$), its corresponding PCA model (Matrix P), and its equivalence in the latent space (τ).....	24
Fig. 2.3: Schematic of Moving Window approach along the batch-wise unfolded dataset.....	26
Fig. 2.4: Multiphase construction and overlapping in a batch-wise unfolded dataset	29
Fig. 2.5: Schematic of the Reactor and the Temperature Control Instrumentation...	38
Fig. 2.6: Control in the latent variable space using PCA models based on the multiphase batch-wise unfolded dataset with 6 phases. Each variable is centered by its own mean in the batch-wise unfolded dataset.....	40
Fig. 2.7: Control in the latent variable space using PCA models based on the multiphase batch-wise unfolded dataset as in Fig. 2.6, but where y is now mean-centered by y_{sp}.....	41
Fig. 2.8: Control in the original MV space using Multiphase batch-wise unfolded dataset with 6 phases along the batch, where y is mean-centered by y_{sp}, CH=20 and PH=60.....	42
Fig. 2.9: a) Implementation of a tightly tuned PI controller for the temperature control problem, b) Implementation of a loosely tuned PI controller for the temperature control problem.....	46
Fig. 2.10: Performance of the LV-MPC in the latent variable space for both tracking the set-point trajectory and rejecting a random walk disturbance occurring on top of the output. Multiphase PCA models, based on the batch-wise unfolded dataset with 6 phases are used.....	49
Fig. 2.11: Performance of LV-MPC in the original MV space for both tracking the	

set-point trajectory and rejecting a random walk disturbance occurring on top of the output. Control and prediction horizons *of CH*=20 and *PH*=60 are used along with multiphase PCA models based on the batch-wise unfolded dataset with 6 phases.....	50
Fig. 3.1: batch process dataset, batch-wise unfolding, and observation-wise unfolding.....	62
Fig. 3.2: Observation-wise with Time-lag unfolding approach.....	63
Fig. 3.3: Schematic of the Regularized Batch-Wise Unfolding (RBWU).....	66
Fig. 3.4: The Left figures show the loading plots corresponding to the first principal component obtained from a batch-wise unfolded dataset for Controlled (T_r) and manipulated (U) variables with 20 batches. Right figures show the same plot but for a RBWU data derived from the same BWU dataset by considering $L=5$. The first case study presented in section 3.4.1 is used to generate this figure.....	68
Fig. 3.5: MLV-MPC for set-point tracking using PCA model on batch-wise unfolded dataset.....	76
Fig. 3.6: Random Walk disturbance rejection of MLV-MPC using PCA model on batch-wise unfolded dataset.....	76
Fig. 3.7: MLV-MPC for set-point tracking using PCA model based on the OWTU dataset.....	78
Fig. 3.8: Random Walk disturbance rejection of multiphase LV-MPC using PCA model on OWTU dataset.....	79
Fig. 3.9: MLV-MPC for set-point tracking using PCA model on RBWU dataset.....	80
Fig. 3.10: Random Walk disturbance rejection of the MLV-MPC with control formulation in LV space using PCA model on RBWU dataset	80
Fig. 3.11: Schematic diagram of the Nylon 6,6 Autoclave.....	83
Fig. 3.12: MLV-MPC based on batch-wise unfolding modeling approach on the nylon 6,6 autoclave using control formulation in LV space	85

Fig. 3.13: MLV-MPC based on batch-wise unfolding modeling approach on the nylon 6,6 autoclave using control formulation in the original variable space. *RMSE* of T/P= 1.4150/ 2.2523 , "*STD of* Δu" for T/P loop= 0.5876/ 65.4617.....	85
Fig. 3.14: Random Walk disturbance rejection by MLV-MPC based on batch-wise unfolding modeling approach on the nylon 6,6 autoclave using control formulation in LV space	86
Fig. 3.15: Random Walk disturbance rejection of MLV-MPC based on batch-wise unfolding modeling approach on the nylon 6,6 autoclave using control formulation in original variable space	86
Fig. 3.16: LV-MPC based on OWTU modeling approach on the nylon 6,6 autoclave	88
Fig. 3.17: Random Walk disturbance rejection by LV-MPC based on OWTU modeling approach on the nylon 6,6 autoclave	88
Fig. 3.18: MLV-MPC using Regularized Batch-wise unfolding approach on the nylon 6,6 autoclave using control formulation in LV space	89
Fig.4.1: Schematic diagram of the closed-loop system in the training data generation step.....	103
Fig. 4.2: (a) Cube of a batch process dataset (b) Observation-Wise with Time-lag Unfolded of the dataset.....	112
Fig. 4.3: (a) Cube of a batch process dataset (b) Batch-Wise Unfolding of a batch process dataset (c) matrix of LV scores.....	113
Fig. 4.4: (a) Cube of a batch process dataset (b) Regularized Batch-Wise Unfolding of a batch process dataset (c) matrix of LV scores.....	114
Fig.4.5: LV-MPC based on BWU modeling approach, using 40 historical batches (with PI controller) plus one extra batch run (with PI controller) on a similar set-point trajectory with a small different level throughout the batch (shifted set-point trajectory).....	122
Fig.4.6: LV-MPC based on BWU modeling approach, using 40 similar batches	

plus one extra batch run on a similar set-point trajectory with a small different level throughout the batch. However, the level difference is positive in the first half time of the batch and negative in the second half time of the batch as a possible way to expect the final product quality to stay within the desired specification tolerance.....	123
Fig. 4.7: a) Performance of the PI controller on the similar but slightly shifted set-point trajectory, b) Performance of the PI controller on the set-point which is above the original set-point for the first half of the batch and then smoothly switches to below the original set-point trajectory for the rest of the batch.....	124
Fig.4.8: LV-MPC based on BWU modeling approach, using 40 batches with similar set-point plus one extra batch run based on a similar but slightly shifted set-point trajectory. The RBS signal of is also delivered on top of the controller output for all batch runs.....	124
Fig.4.9: LV-MPC based on RBWU modeling approach, using 20 similar batches plus one similar but shifted set-point trajectory, *STD of* Δu*=6.5075, RMSE of (y-*y_{sp}*)=0.6807*.....	126
Fig.4.10: LV-MPC based on OWTU modeling approach, using 2 historical batches in the dataset run with PI controller, *STD of* Δu*=6.4766, RMSE of (y-ysp)= 0.9938*.....	126
Fig.4.11: Trajectory tracking by (a) Loose PI controller used in the identification data generation step, *(Kc=10,* τ_i*=30),* (b) LV-MPC based on a PCA model obtained from batch-wise unfolded closed-loop data which is generated by the loose PI controller.....	128
Fig.4.12: Trajectory tracking by (a) tight PI controller used in the identification data generation step with smaller *Kc* and larger K_i *(Kc=40,* τ_i*=4), (b)* LV-MPC based on a PCA model obtained from batch-wise unfolded closed-loop data which is generated by the tight PI controller.....	129
Fig.4.13: Trajectory tracking by (a) tight PI controller used in the identification data generation step with larger *Kc* and smaller K_i *(Kc=60,* τ_i*=120), (b)* LV-MPC	

based on a PCA model obtained from batch-wise unfolded closed-loop data which is generated by the tight PI controller.....	130
Fig. 4.14: Bode diagram for different PI controllers used in the data generation step and an approximated process transfer function.....	132
Fig. 4.15: a) The pseudo disturbance resulting from mean centering y with y_{sp} b) the spectrum of the pseudo disturbance.....	133

LIST OF TABLES

Table 2.1- Comparison of different control algorithms for the temperature control problem.....	46
Table 3.1- Comparison of the effect of different modeling alternatives for the Temperature set-point tracking control by LC-MPC	81
Table 3.2- Comparison of the effect of different modeling alternatives on the set-point tracking control of the Nylon 6,6 process by LV-MPC	90
Table 4.1: Comparison of the LV-MPC performance based on different alternative identification dataset with the NMPC.....	125
Table 4.2: summary of performance of the LV-MPC based on a PCA model built on closed-loop data generated by different tunings of a PI controller.....	131

NOMENCLATURE:

Latin Letters

A(k),B(k): Coefficient matrices in a time-varying state space model

CH: Control Horizon

C_{pj} : Thermal capacity of the fluid in the jacket in the SISO case study *(kcal/kmol°C)*

d: dither signal for external excitation on top of controller output

D_T: set of parameters for which plant and model behave the same

e: White noise considered in the system

fh: future horizon in a multiphase model

F: Controller Dynamics

G: Plant Dynamics

: estimated transfer function from spectral analysis method

$\tilde{G}$: estimated transfer function from PEM method

: estimated transfer function from correlation method

H: Disturbance (Noise) Dynamics

I: number of batches in the dataset

$\mathcal{I}$: Identification Method

J: number of measurements

K: time duration of a batch (phase)

L: Set-point Filter

M: Model Structure

P: Loading matrix transforming X to T

PCA: Principal Component Analysis

PH: Prediction Horizon

PLS: Projection to Latent Structure (Partial Least Squares)

ph: past horizon in a multiphase model

n : total number of observations in the identifiability definitions

Q_j : Heat transfer between reactor and jacket in the SISO case study *(kcal)*

r: Number of Controller settings in the training data generation step

r_u: correlation matrix of the input variable

r_{yu}: cross-correlation matrix between input and output variables

S : System

T: Matrix of latent variable scores corresponding to X

T_j, T_j^{sp} : Jacket Temperature and its set-point in the SISO case study (*°C)*

u_c: manipulated variable

u: Input Variable

V : The matrix of Eigen vectors of the spectral density function of the time series "X"

V_1, V_2: LQ matrices

V_j: Jacket volume in the SISO case study(m^3)

x: a new observation (in the control step) in which the past data are known and future data are missing

x_{me}: extra measurements, any measurement other than u_c, y_{cv}, and y_{sp}

X: batch dataset

$\mathcal{X}$: Identification Experiment

y_{cv}: controlled variable

y_{sp}: Set-point variable

y: OutputVariable

W: Left Singular Matrix

Y: Right Singular Matrix

z: a new observation (in the missing data imputation step) containing missing data

Greek Letters

$\hat{\alpha}_k$: Coefficients of the numerator of the process transfer function

$\hat{\beta}_j$: Coefficients of the denominator of the process transfer function

β_k : Coefficient of past information in the missing data imputation algorithm

γ_u : Auto-correlation function of signal u

γ_{yu} : Cross-correlation function of signals y and u

γ_z : Auto-correlation function of signal z

γ_v : Auto-correlation function of signal v

Γ: State transition matrix in a state space model

$\Delta\hat{\tau}_k$: correction of the current score in the control algorithm in the LV space

ε: White noise considered in the model

ζ : a vector containing all measurements at a sample time

: vector of parameters used in the model structure

Θ : Covariance of score matrix (T)

v : Set-point signal in the training data generation step for identification

ξ : the Principal Components (PCs) in the frequency domain

ρ_j : density of the fluid in the jacket in the SISO case study (kg/m^3)

Σ: Matrix of singular values

$\hat{\tau}_k$: estimated score of the batch at sample time k

τ_j : Time constant of the jacket temperature in the SISO case study (*min*)

ϕ:Corrected score to take into account the effect of past happened data

Φ_u: Spectral density function of signal u (manipulated variable)

Φ_{yu}: cross-spectral density function of the input and output variables

Ψ : Controllability Matrix for the state space model

$\overline{\Psi}$: Controllability Matrix for the PCA model

List of Acronyms and Short Words

CCA: Canonical Correlation Analysis

CMR: Conditional Mean Replacement

CV: Controlled Variable

DOE: Design Of Experiment

IVM: Instrumental Variable Method

LQG: Linear Quadratic Gaussian

LTI: Linear Time Invariant

LTV: Linear Time Varying

LVM: Latent Variable Model

LV-MPC: Latent Variable Model Predictive Control

MIMO: Multi Input Multi Output

MPC: Model Predictive Control

MV: Manipulated Variable

NMPC: Nonlinear MPC

PARAFAC: Parallel Factor Analysis

PEM: Prediction Error Method

PID controller: Proportional Integral Derivative controller

PMP: Projection to the Model Plane

randn: Random number

RBS signal: Random Binary Sequence signal

RMSE: Root Mean Square Error

RRA: Reduced Rank Analysis

SCP: Single Component Projection

SI: System Identifiability

SID: System Identification

SIM: Subspace Identification Method

SISO: Single Input Single Output

SSE: Sum of Square Errors

SSI: Strong System Identifiability

STD: STandard Deviation

SVD: Singular Value Decomposition

TSR: Trimmed Score Regression

CHAPTER 1

Introduction

This thesis addresses the problem of trajectory tracking control in batch processes. The research focuses on the Model Predictive Control (MPC) algorithm for trajectory tracking control by utilization of Latent Variable Models (LVMs) in the prediction step. A novel model predictive control algorithm for the trajectory tracking of batch processes is developed and investigated to address several practical situations. Different modeling alternatives are studied in the course of the proposed control methodology. Furthermore, specific identification issues of batch processes are explored in this study. The current chapter presents an overview of batch processes, Latent Variable Models (LVMs), and the concept of the model predictive control. Furthermore, the thesis objectives and outline are presented.

1.1. Batch Processes

Batch processes are processes that happen in a finite duration of time. In a batch process, materials are initially charged to the process. Then the batch starts without

further addition or withdrawal of materials. At the end of the process, final products are discharged from the batch. A semi-batch process is a special case of batch process in which make-up and/or removal of materials may occur a few times during the batch completion. The operating conditions of a batch process including the specifications of initial materials and the process variable trajectories are not fixed and determine the final product quality. Batch operation also occurs in continuous plants. A continuous process is also considered as a batch process during the start ups and transitions between different product grades.

Batch processes are of interest for many industrial applications such as pharmaceuticals and specialty chemicals. The main reason to use batch processes is their flexibility in producing a wide range of products. It is possible to produce a vast number of products using the same batch process equipment by changing the initial conditions and process variables trajectories.

The control of batch processes is an open subject of research because of many distinguishing features of batch processes, as compared to continuous processes, such as lack of steady-state point, finite time duration of the process, and significant nonlinear and time-varying behavior existing in batch processes because of wide range of operating conditions.

There are two hierarchical control objectives in a batch process. The higher level control is the control of final product quality by manipulating the initial conditions of the batch as well as the process variables trajectories [1-3]. The lower level control is the control of the process variables to track their corresponding set-point trajectories dictated

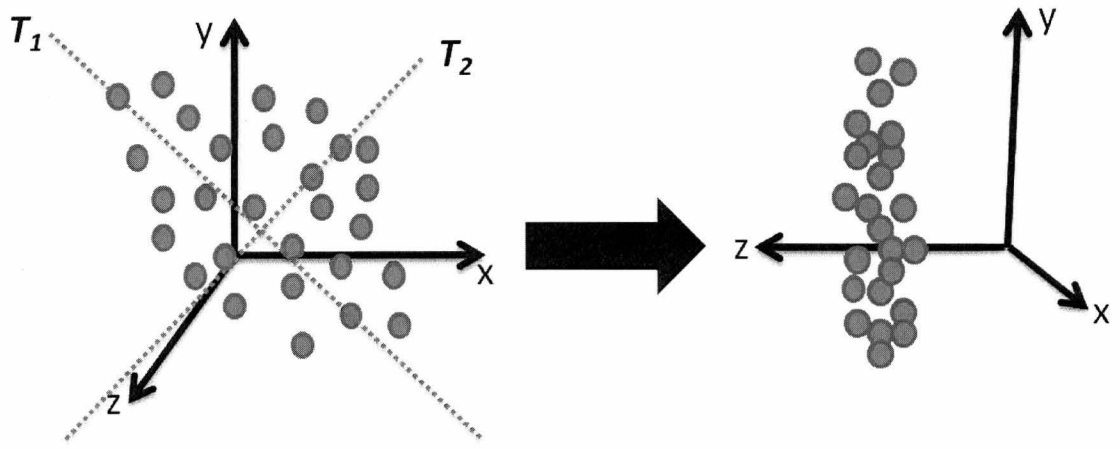
by the higher level controller [4-6]. The focus of this research is the trajectory tracking control of batch processes (lower level control) by identifying the batch process model using latent variable modeling approach and incorporating the process model in the course of a model-based controller.

1.2. Latent Variable Models (LVMs)

Latent variable modeling approach is a class of modeling which focuses on dimension reduction of the dataset by explanation of the major variations as well as quantification of the correlation structure in the dataset. There are several LVMs such as Projection to Latent Structure (PLS), Canonical Correlation Analysis (CCA), Reduced Rank Analysis (RRA), and Principal Component Analysis (PCA). A detailed discussion of the LVMs is presented in [7],[8]. PCA model is the latent variable model utilized in this study.

Principal Component analysis (PCA) is a useful method in data compression by orthogonal transformation of the data matrix. The main idea behind the PCA is to reduce the dimension of a dataset consisting of a large number of possibly correlated variables while preserving as much information in the dataset as possible [9],[10]. In other words, PCA is a method of transforming a set of correlated variables into a set of fewer uncorrelated variables. The new uncorrelated variables are called Principal Components (PCs) and capture the most variations in the dataset. Figure 1.1 shows an illustration of this concept.

Figure 1.1a demonstrates a typical dataset in which each observation is presented in a three dimensional coordinate and depicted as a circle. Figure 1.1b is a rotated version of Figure 1.1a revealing that most of the variations are in the XY plane and most of the information content of the dataset can be explained by projecting the observations on a two dimensional plane. T_1 and T_2 axes depicted in Figure 1.1a represent the directions of the maximum variances in the dataset and are numbered in the order of importance (T_1 is the first direction of maximum variance and T_2 is the second direction of maximum variance). Hence, the PCA model is a map that transforms the observations from the original space (X,Y,Z) to the reduced latent space (T_1,T_2).



(a) (b)

Figure 1.1: Illustration of a typical dataset to which PCA can be applied.

Latent variable models including PCA are frequently used in the literature for the purpose of prediction [1],[11],[12]. In this research, the prediction of future process outputs obtained by the PCA model is utilized in the course of the Model Predictive Control algorithm for trajectory tracking in batch processes.

1.3. Model Predictive Control

Model Predictive Control (MPC) is a class of control methodologies which utilizes the process model to predict the plant future outputs and calculates a set of control actions to be implemented to the process to optimize an objective function. This idea is implemented through a sequence of steps as follows [13]:

a) Prediction of the future output of a specific horizon by explicit use of process model.

b) Calculation of a sequence control actions which minimizes an objective function.

c) Implementation of the control action in the receding horizon manner, so that at each sample time, the first element in the sequence of control actions is implemented to the process.

The above-mentioned sequences of steps are repeated at each sample time after the new measurements become available. Various MPC algorithms are different due to diverse types of models used in the prediction step and different objective functions considered to be minimized. Figure 1.2 is an illustration of the MPC algorithm. The man looking backward and forward represents the process model. At the sample time *"k"* the model looks back to use all past information, including current measurements, needed to calculate the prediction of future output of the process. It also looks forward in the sense of prediction to estimate the behavior of process output in a specified horizon called Prediction Horizon (*PH*) and calculates sequence of "CH" (the acronym for "Control Horizon") corrective actions to minimize an objective function. In Fig. 1.2, the objective

function can be a quadratic function of the process output deviation from its corresponding set-point. At the next sample time, $k+1$, all of the above steps are repeated to take into account any new information obtained from the new measurements.

One of the major bottlenecks in the application of MPC is the availability of a reliable process model to be used for the prediction purpose. Mechanistic process models are not always available or reliable. Transfer Functions and State Space models, that may be representatives of mechanistic or empirical models, are amongst the most popular models used in the course of MPC[13-16]. The Linear Time-Invariant (LTI) versions of such models are satisfactory for the application of MPC in most continuous processes as the operating condition is usually a fixed point under normal operation. In contrary, in batch processes, the operating condition changes significantly during the batch completion which results in the nonlinear and time-varying behavior of the process. Thus, application of a LTI model in the course of MPC for batch processes may not be satisfactory in many cases. On the other hand, nonlinear models are not preferred to be used in the MPC algorithm due to the difficulties they bring to online computations.

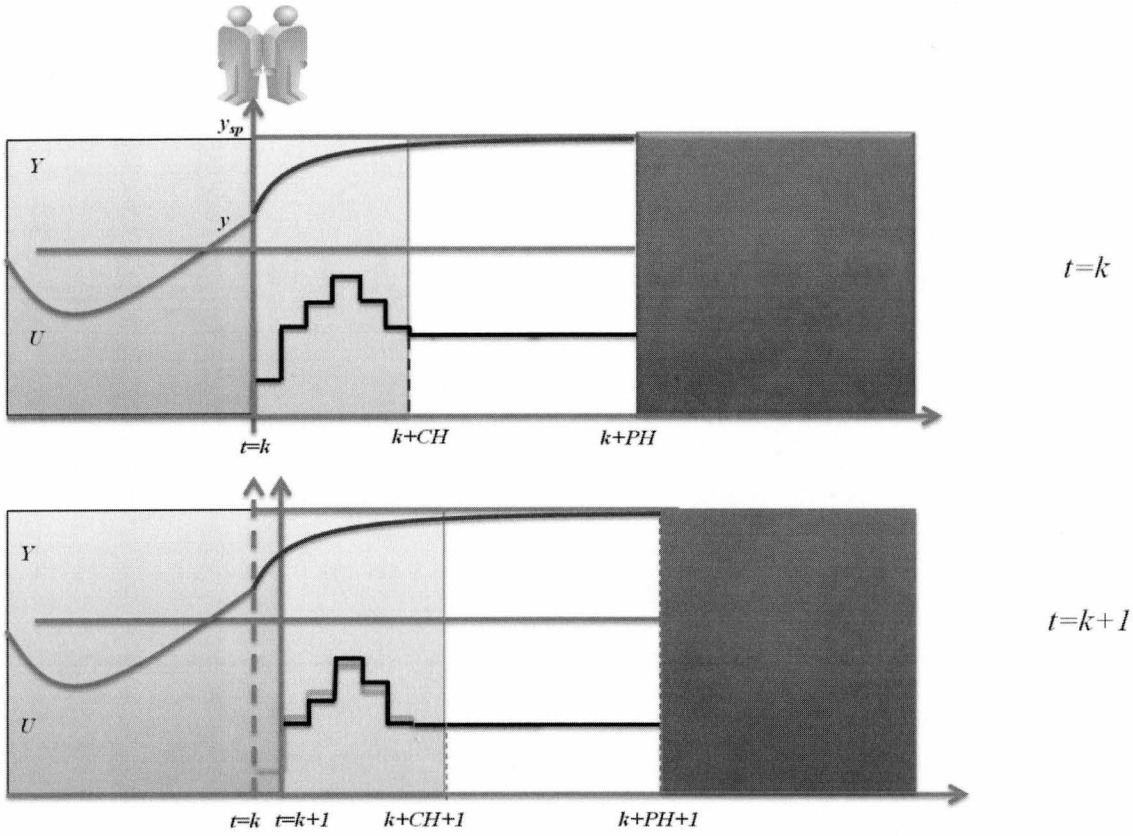


Figure 1.2: An example of illustration of the MPC algorithm

There have been few studies in the literature to address the problem of model-based trajectory tracking in batch processes[4],[5]. The state of the art is to use the most accurate model of the process which is a nonlinear mechanistic model in the course of MPC, called Nonlinear MPC (NMPC), for trajectory tracking in batch processes. The current research aims to address the problem of incorporating a reliable and achievable model that accounts for the special characteristics of batch systems into the MPC framework. The LVMs are selected in this study to play the important role of the process model. They are easy to obtain and their prediction techniques are computationally efficient which make them suitable for online applications. The LVM used in this study is

the PCA model. Although PCA is a linear model, different techniques are used in this study to model nonlinear and time-varying behavior of the batch processes.

1.4. Research Objectives and Scopes

The Goal of this research is to develop a MPC methodology for trajectory tracking in batch processes which uses the PCA model in the prediction step. The proposed methodology is called LV-MPC.

The LV-MPC algorithm aims to address several problems existing in trajectory tracking control of batch processes. Nonlinearity and time-varying behavior of the batch processes must be addressed in the modeling step. Furthermore, the need for digestible computational burden necessary for online applications should be considered. Moreover, major practical difficulties such as lack of enough number of observations to build a process model must to be dealt with in the LV-MPC framework.

All of the above objectives may not be addressed in one algorithm. Thus, the objective of this research is to perform a comprehensive study on various control formulations and modeling alternatives and to develop a system and methodology consisting of several alternative algorithms to address the abovementioned objectives and to propose a set of guidelines and criteria that defines the situations under which each algorithm is recommended for practitioners.

The LV modeling approach is amongst the empirical modeling approaches. A training dataset is required to build an empirical model for a process. The process of training data generation and building an empirical model on the generated dataset is

called System IDentification (SID). The quality of the identified model has a direct influence on the quality of the model-based control algorithm which utilizes the model. Therefore, a number of important SID issues are also addressed in this research. The system identification for batch processes is an undiscovered area of research and there are many unanswered questions in this regard. The scope of this research is to investigate the problem of batch process identification and to address a number of questions that are most relevant to the LV-MPC methodology including closed-loop indentifiability conditions for batch processes and optimal training data generation. These questions are to be answered in this research by focusing on the LV modeling approach for batch processes.

1.5. Thesis Outline

The remainder of the thesis is organized as follows:

Chapter 2 reviews the control objectives in batch processes and the state of the art techniques for trajectory tracking control in batch processes. Then the novel LV-MPC methodology is developed. The motivation of this study is to develop a MPC algorithm based on a reliable and achievable empirical model that can be used when an accurate mechanistic model is not available. Two control alternatives are investigated in the course of LV-MPC. The concept of multiphase modeling is incorporated in the proposed LV-MPC. Finally, the proposed two LV-MPC alternatives are tested on a simulated Single-Input Single-Output (SISO) batch reactor case study for temperature trajectory tracking and disturbance rejection. The performance of the proposed LV-MPC on this

batch case study is also benchmarked against PI controller as well as Nonlinear MPC (NMPC).

Chapter 3 incorporates two more modeling alternatives in the LV-MPC framework. The motivations for this study are to address the practical problem of LV modeling of batch processes using few batch runs in the dataset and the aim to improve the LV modeling techniques. The finalized LV-MPC methodology consists of three modeling alternatives and two control formulation options. The three modeling alternatives incorporated into the LV-MPC method are implemented on two simulated examples of batch processes: the SISO batch example studied in chapter 2 and a Multi Input Multi Output (MIMO) batch example.

Chapter 4 investigates the identification issues for batch processes in the course of LV modeling approach which is the focus of this research. The identifiability conditions in the context of batch processes are discussed. Then, the necessary conditions for training data generation from a closed-loop system which is informative enough to build a LVM and leads to a small bias in the identified model are proposed.

Chapter 5 presents the summary of the contributions of the thesis followed by several suggestions for related future work.

References

[1] J. Flores-Cerrillo and J.F. MacGregor, "Control of batch product quality by trajectory manipulation using latent variable models," *Journal of Process Control*, vol. 14, 2004, pp. 539-553.

[2] T.J. Crowley, E.S. Meadows, E. Kostoulas, and F.J. Doyle III, "Control of Particle Size Distribution described by a Population Balance Model of Semibatch Emulsion Polymerisation," *Journal of Process Control*, vol. 10, 2000, pp. 419-432.

[3] D.J. Kozub and J.F. MacGregor, "Feedback control of polymer quality in semi-batch copolymerization reactors," *Chemical Engineering Science*, vol. 47, 1992, pp. 929-942.

[4] T. Clarke-Pringle and J.F. MacGregor, "Nonlinear adaptive temperature control of multi-product, semi-batch polymerization reactors," *Computers & Chemical Engineering*, vol. 21, 1997, pp. 1395–1409.

[5] C. Kravaris, R.A. Wright, and J.F. Carrier, "Nonlinear controllers for trajectory tracking in batch processes," *Computers & Chemical Engineering*, vol. 13, 1989, pp. 73-82.

[6] J. Flores-Cerrillo and J.F. MacGregor, "Latent variable MPC for trajectory tracking in batch processes," *Journal of Process Control*, vol. 15, Sep. 2005, pp. 651-663.

[7] T. Kourti, "Application of latent variable methods to process control and multivariate statistical process control in industry," *International Journal of Adaptive Control and Signal Processing*, vol. 19, 2005, pp. 213-246.

[8] A.J. Burnham, R. Viveros, and J.F. MacGregor, "Frameworks for latent variable

multivariate regression," *Journal of Chemometrics*, vol. 10, 1996, pp. 31-45.

[9] I.T. Jolliffe, *Principal Component Analysis*, Springer-Verlag, 1986.

[10] E. Johansson, N. Kettaneh-wold, S. Wold, and L. Ericksson, *Multi- and Megavariate Data Analysis Principles and Applications*, Umetrics Academy, 2001.

[11] P. Nomikos and J.F. MacGregor, "Monitoring batch processes using multiway principal component analysis," *AIChE Journal*, vol. 40, 1994, pp. 1361-1375.

[12] P.R.C. Nelson, P.A. Taylor, and J.F. MacGregor, "Missing data methods in PCA and PLS: Score calculations with incomplete observations," *Chemometrics and Intelligent Laboratory Systems*, vol. 35, Nov. 1996, pp. 45-65.

[13] E.F. Camacho and C. Bordons, *Model Predictive Control*, Springer, 2004.

[14] C.E. Garcia, M.D. Prett, and M. Morari, "Model predictive control: Theory and practice - A survey," *Automatica*, vol. 25, 1989, pp. 335-348.

[15] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, Jun. 2000, pp. 789-814.

[16] J.B. Rawlings, "Tutorial overview of model predictive control," *Control Systems Magazine, IEEE*, vol. 20, 2000, pp. 38-52.

CHAPTER 2

Latent Variable Model Predictive Control (LV-MPC) for Trajectory Tracking in Batch Processes

Contents of this chapter have been published in the **Journal of Process Control**.

Citation:

Masoud Golshan, John F. MacGregor, Mark-John Bruwer, Prashant Mhaskar, 2010, Latent Variable Model Predictive Control (LV-MPC) for Trajectory Tracking in Batch Processes, Journal of Process Control 20, 538-550.

Contributions of the Ph.D. Candidate:

Masoud Golshan developed the theoretical formulation of the control in the latent variable space and incorporation of different batch process modeling alternatives studied in this chapter into the LV-MPC in consultation with Dr. John F. MacGregor. Mathematical formulation of the control algorithm in the original variable space is proposed by Mark-John Bruwer. All the simulation studies related to the LV-MPC methodologies are performed by Masoud Golshan in consultation with Dr. John F. MacGregor. Formulation and simulation of NMPC is performed by Masoud Golshan in consultation with Dr. Prashant Mhaskar. The paper is written by Masoud Golshan and edited by Dr. John F. MacGregor and Dr. Prashant Mhaskar.

2.1. Introduction

Batch processes exhibit a number of characteristics that lead to interesting control problems. They are finite duration processes in which the objective is to achieve a desired product by the very end of the batch. The product quality is usually only measured off-line in a quality control lab after completion of the batch. Furthermore, batch processes are nonlinear in that the gains and dynamics often vary continuously throughout the duration of the batch.

There are basically two levels of control for batch processes. The higher level control is the control of final product quality at the end of the batch. The lower level control is the set-point tracking of certain process variable trajectories. Several approaches to both of these control problems have been proposed and applied. In practice, the high level problem of controlling final product properties is not treated directly. Through well automating the batch process (the sequencing of all batch stages and the implementation of low level controls), the batch processes are simply run open-loop and acceptable final quality is achieved in most cases. Theoretical approaches in the literature to the high level control of final product quality are often based on the use of mechanistic models. Nonlinear optimization/control algorithms combined with nonlinear state estimation have often been proposed to continually re-compute the optimal trajectories throughout the batch [1]. The difficulties in practice are the computational effort of such approaches as well as the need for good mechanistic models. Less computationally intensive differential geometric control approaches based on achieving necessary and sufficient conditions for optimal final quality have been proposed for

specific situations [2],[3]. Much simpler approaches to the high level control of final quality have been based on the idea of mid-course corrections and the use of empirical models such as latent variable models [4-8] or Artificial Neural Network models[9]. The assumption is that with good automation, most batches will achieve acceptable quality and that continual manipulation of the process every few seconds is not necessary, nor desirable. For those batches in which the final quality (as inferred by an inferential model) is projected to be outside of an acceptable window, mid-course corrections are applied.

The lower level control problem that is addressed in this research involves the tracking of set-point trajectories for key process variables such as temperature, pressure or a measured reactant concentration. The desired set point trajectories are either predetermined from experience, from off-line optimization, or from the execution of the high level control. In industrial practice these trajectory tracking control problems are usually handled by simple PID controllers. However, when control is not uniformly acceptable over the entire batch, due to changing gains and dynamics or the need to track complex set-point trajectories, gain scheduled PID controllers are used [10] or feed-forward terms are added [11],[12]. Nonlinear controller approaches based on mechanistic models have also been described in the literature. Differential geometric approaches [12],[13] and various versions of nonlinear MPC have been proposed. Garcia et al.[14] first proposed a nonlinear MPC for batch processes based on a nonlinear prediction model and control using a linearization of the model. Fully nonlinear MPC, formulated in an optimization framework, has more recently been proposed [15],[16].

However, the modeling, implementation and computational effort of such approaches are large.

This study considers an alternative to the nonlinear MPC approach for trajectory tracking and disturbance rejection based on latent variable models built from closed-loop batch data. Flores-Cerillo and MacGregor [8] first proposed this approach (LV-MPC) and illustrated it on the tracking of temperature and pressure in simulated batch processes. This chapter reformulates the LV-MPC problem in two different ways and uses multi-phase PCA model (overlapping models from each phase of the batch). The two formulations proposed are: (i) where all the optimization is computed in the reduced dimensional space of the latent variables and the future MV trajectories then computed from these optimized latent variables, and (ii) where all the control calculations are performed in the space of the original manipulated variables (more conventional approach).

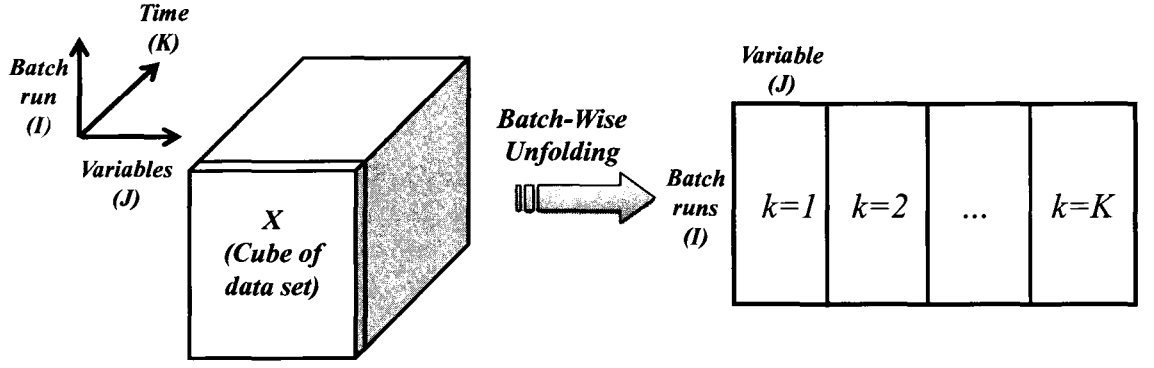
The rest of the chapter is organized as follows: Section 2.2 discusses latent variable models, system identification for dynamic batch processes, and prediction methods associated with latent variable models. Section 2.3 presents the basic methodology of two versions of multi-phase LV-MPC. Section 2.4 illustrates the methodology for trajectory tracking and disturbance rejection on a batch reactor simulation and Section 2.5 presents conclusions.

2.2. Latent variable modeling, identification and prediction

2.2.1. Structure of batch data

The structure of the data collected on time-varying trajectories from a batch process is a cube as shown in Fig. 2.1a. There are different approaches for rearranging these data for analysis [17-20]. The main difference among different unfolding approaches is how they construct a 2-dimensional array (a matrix) out of the cube of dataset. Nomikos [19] and Nomikos and McGregor [20-22] suggested many possibilities, but proposed the batch-wise unfolding approach shown in Fig. 2.1 as the most logical way for modeling differences among batches. In this approach all the variables at different sample times are unfolded as shown and each batch history constitutes one observation or row in the unfolded matrix (Fig. 2.1b). When all the batches are analyzed by performing Principal Component Analysis (PCA) or Projection to Latent Structures (PLS) on this unfolded matrix a few LV scores summarize the major differences among the batches, and the LV loadings capture all the time-varying dynamics of the batch process. Other unfolding approaches [17],[19],[23] either do not capture the time-varying dynamics of the process or do not extract the differences among the batches. However, they can provide alternative approaches to LV-MPC and these are explored by Golshan et al. [24]. Nomikos and MacGregor [19],[20], and Wold et al.[25] presented the variable-wise unfolding approach in which variables of each sample time are considered as an independent observation. The schematic diagram is the same as Fig. 2.1b but the

time slices are arranged underneath each other instead of beside each other. Louwerse and Smilde [17] have also considered PARAFAC and Tucker 3 multiway models for batch processes but Westerhuis et al.[26] illustrate and discuss why these approaches are not well suited for batch processes.



(a) $X_{cube(I\times J\times N)}$ (b) Batch-wise Unfolding

Fig. 2.1: a) Cube of Batch process dataset, b) Batch-wise unfolding of the dataset

2.2.2. Latent Variable (LV) models

Modeling batch-wise unfolded data as in Fig. 2.1b using PCA results in a reduced dimension latent variable model for the $(I\times JK)$ matrix X of the form:

$$X = TP^T + E \tag{2.1}$$

$$T = XP \tag{2.2}$$

where T is a $(I\times A)$ matrix $(A<<JK)$ of latent variable scores that summarize the major differences among the batch trajectories, and P is a $(JK\times A)$ matrix of loadings that show how the latent variable scores are related to the trajectory data (X). The score values of the A latent variables for each batch summarize the time-varying behavior of its trajectories relative to all other batches.

A common misconception of these latent variable models (e.g. PCA or PLS) is that they are linear models and therefore cannot capture any of the time-varying nonlinear dynamics of batch processes. However, this is not true. In the LV modeling of batch-wise unfolded data (Fig. 2.1), the data are first mean-centered and scaled [19]. Mean centering the data automatically removes the average trajectories of all variables and hence removes the main nonlinearities relating to the absolute levels of the variables. Then PCA or PLS applied on these deviations provides different loadings (p_{ak}'s) at every point in time ($k = 1, .. ,K; a=1, .., A$)), thereby modeling the covariance structure at every point in time and its changes with time (see for example the loading plots in Garcia-Munoz et al. [27]). In effect, it provides a nonlinear model in the form of a locally linearized model of the covariance structure of the variables at every point in time. These LV methods, applied to such batch-wise unfolded data, have now been applied to industrial batch processes in a variety of industries (see for example[5],[20],[27-30]) with little evidence of poor behavior due to not modeling nonlinearities. Nonlinear PLS and PCA methods have also been applied to many of these same industrial data sets and no improvement in the fit of the model nor any reduction in the number of components have been found [31].

2.2.3. Identification

The training data can be the data from the previous batches run under normal conditions augmented with additional batches executed according to Design Of Experiments (DOE) for identification to provide information on the causal relationship between the manipulated variables and the controlled variables at every time interval

throughout each batch. The direct identification approach based on closed-loop data is used in this study. Closed-loop identification is preferred over open-loop identification for batch processes in order to maintain the process close to its desired trajectories and to minimize the final product quality variations. A dither signal in the form of a Random Binary Sequence (RBS) is added on top of the manipulated variable trajectories coming from an existing controller (PID) to provide some additional excitation of the process. The RBS signal is chosen to have its switching frequency in a suitable range (~ 1/6-1/3 of the dominant time constant of the process). The designed RBS signals simply improve the causal relationships between the MV and CV along their trajectories. The historical batch data are also very important in providing models for the effects of inherent disturbances in the batch process and their influence on the behavior of the evolving trajectories. This information is essential for the prediction of the future trajectories as discussed in the next section.

Another important issue in the selection and design of identification experiments is the inclusion of experiments using somewhat different set-point trajectories. This is mainly of importance if the MPC is to be required to track a range of set-point trajectories such as might be needed for achieving different grades of the product. In practice, historical batch data would usually be available on these different grades and could be included in the training data.

2.2.4. Prediction of future trajectories

Prediction plays an important role in the MPC algorithm since the optimization problem embedded in the MPC needs future prediction of the outputs up to the prediction

horizon. The prediction method depends on the type of model being utilized in the MPC. For most linear and nonlinear dynamic models used for MPC, the future prediction is calculated using integration of the dynamic model over the prediction horizon (*PH*) and adapting it assuming a simple random walk type disturbance model on the CV. For some nonlinear MPC's state prediction is often improved by modeling disturbance states entering through selected variables (e.g., impurity concentrations in polymerization reactors [3].

The PCA models built on historical plus DOE data use all measured variables (not just MVs and CVs) and therefore contain within them detailed structure on the propagation of both CV changes and disturbances throughout the batch. These finite duration batch models also have a loading matrix (*P*) that shows how the covariance structure among the variables changes over the future horizon. The prediction step for the PCA latent variable model is therefore accomplished via statistical missing data imputation methods. These are efficient estimators for the future trajectories and serve exactly the same role as state estimators (e.g., Kalman Filters, etc.) for prediction of the future trajectories in traditional MPC approaches. These methods use all past data up to the current point in time and the time-varying batch model to impute the future (missing data) in any batch. Several missing data imputation methods have been proposed for latent variable models in the literature. For batch processes the aim is to predict the final latent variable scores at the end of any batch and then from the PCA model of the X-space, the values of all the missing trajectories over the remainder of the batch can be computed. Nelson et al. [32] presented an analysis of several methods including the

Single Component Projection (SCP) method, the Projection to the Model Plane (PMP) method, and the Conditional Mean Replacement (CMR) method. Arteaga and Ferrer [33] discussed the methods proposed by Nelson et al. [32] as well as some other methods including a Trimmed Score Regression (TSR) method. Since the PMP and TSR are the methods utilized in this study, they are briefly discussed here.

Projection to the Model Plane (PMP)

This method was used by Nomikos and MacGregor [19],[21] in their original batch monitoring methodology. As the name implies it projects the new vector of observations with missing data onto the plane defined by the latent variable model (2.1) to obtain an estimate of the missing part of the data vector that is consistent with the model.

A new observation (z) can be divided in two parts as shown in Fig. 2.2:

$$z^T = [z^{*T}\ z^{\#T}] \tag{2.3}$$

where, z* is the known data and $z^{\#}$ is the missing data. For the batch process analysis z^* corresponds to the past data and $z^{\#}$ to the future data. The loading matrix can also be divided into two parts in the same way as z.

$$P^T = [P^{*T}\ P^{\#T}] \tag{2.4}$$

Thus, the PCA model can be partitioned as:

$$z = \begin{bmatrix} z^* \\ z^{\#} \end{bmatrix} = \begin{bmatrix} P^*\tau \\ P^{\#}\tau \end{bmatrix} \tag{2.5}$$

If the known part of the data is used for score estimation, the following relation is obtained [32]:

$$\hat{\tau} = \left(P_{1:A}^{*T} P_{1:A}^{*}\right)^{-1} P_{1:A}^{*T} z^{*} \tag{2.6}$$

where $\hat{\tau}^T = [\hat{t}_1, \hat{t}_2, ..., \hat{t}_A]$ is the vector of estimated score. Subscript *1:A* means that A principal components are considered in the PCA model. The estimates of the trajectories of all the variables for the remainder of the batch ($z^{\#}$) are then obtained from equation (2.5).

Trimmed Score Regression method (TSR)

For this method the same partitioning is applied to the data. The scores are computed based on the assumption that the known part of the data is the complete data in the observation. Thus,

$$\tau^{*} = P^{*T} z^{*} \tag{2.7}$$

and the estimated scores are calculated by regressing the estimated scores ($\hat{\tau}$) on the fake scores (τ^*). Finally, the score estimation formula is [33]:

$$\hat{\tau} = \Theta_{1:A} P_{1:A}^{*T} P_{1:A}^{*} (P_{1:A}^{*T} P_{1Q}^{*} \Theta_{1Q} P_{1Q}^{*T} P_{1:A}^{*})^{-1} P_{1:A}^{*T} z^{*} \tag{2.8}$$

where, Θ is the covariance matrix of the scores ($\Theta = (T^T \times T)/I$) in the PCA model, where "*I*" is the total number of batches in the dataset. The number of scores considered in Θ (Q) can be more than or equal to A.

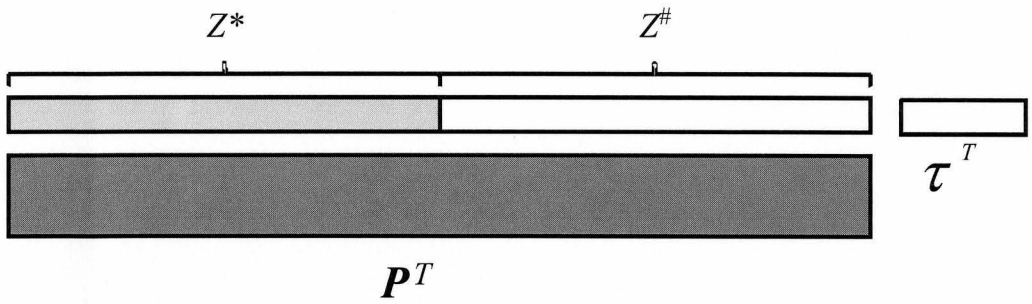


Fig. 2.2: An observation (Z) containing already observed data (Z^*) and missing data ($Z^\#$), its corresponding PCA model (Matrix P), and its equivalence in the latent space (τ)

2.3. Methodology

2.3.1. Multiphase LV models

For the high level control of final product quality, a single PLS model built on the batch-wise unfolded matrix of the entire trajectories (K time intervals) for all batches is desirable since it allows for modeling the effect of variations at every time point throughout the batch on the final product. Furthermore, the number of time intervals needed to define the batch for this high level control problem is usually small ($K \sim 100$ to 300). However, in the low level control problem where the concern is the local tracking of certain trajectory variable set-points, the number of time intervals used (K) is usually much larger (control every few seconds), and the objective is to model the covariance structure of the data over some relatively local horizon. (The impact of current manipulated variable changes on the final quality or on the trajectories in the distant future is less important.) As a result, one very large global LV model (for the entire K time intervals) is less desirable as it requires more latent variables (which implies more batches may be needed in the training set), it leads to more ill-conditioned matrices in the

model used during the control computations, and it does not focus as well on the local behavior of the trajectories. Furthermore, utilizing a large model in the control action calculations at each sample time imposes a larger computational expense. Therefore, use of more local, multi-phase LV models, as presented here, offers several advantages for the low level trajectory tracking problem.

The multiphase modeling approach is again based on the batch-wise unfolding of the cube of the batch dataset, but then identifying multiple phases within the batch and building separate, but overlapping PCA models for each phase. Two approaches are discussed below.

Moving window LV model

One approach is to build PCA models applicable to every time point based on a moving window along the batch-wise unfolded dataset. In this technique, a fixed-size window of data is selected at each sample time in a way that the data of the current time is located at the middle of the window and a model future horizon (*fh*) and past horizon (*ph*) of data on the two sides. Then, a PCA model is built on that part of data and that model used for application of the control algorithm for that time interval of the batch. This modeling approach is applied for every time point throughout the batch leading to as many models as one has time points. The moving window approach is depicted in Fig.2.3. However, at the beginning or end of the batch, there is not enough information to define the past or future horizon respectively and hence models with an expanding past horizon or shrinking future horizon are considered. This approach removes the

aforementioned problems associated with a single global PCA model over the whole batch and allows for smooth progression from one model to the next because of the large overlap in the data used for successive models. However, building as many PCA models as the number of sample times is clearly not necessary because the correlation structure among the process variables does not change that fast. Therefore, a more reasonable phase-based multi-block approach is suggested below.

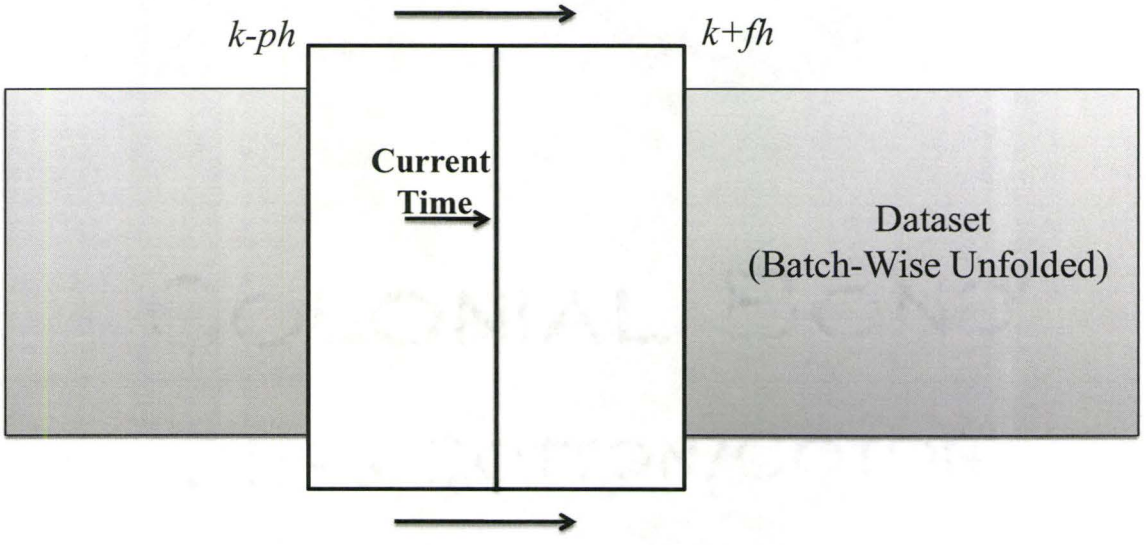


Fig. 2.3: Schematic of Moving Window approach along the batch-wise unfolded dataset

Multiphase LV model

One can often identify certain phases during the batch. By dividing the batch up into a number of such phases a separate PCA model can be built for each of those phases. By then using these models for control within their respective phases the disadvantages of the two extreme approaches above (single global model or a moving window model) can be avoided.

A question is how to select the phases. Although there are some recent studies on phase selection in a batch dataset, there is still no unified approach. Camacho and Pico [18] proposed a method for phase selection based on the prediction error or explained variance error and Zhao et al. [34] presented a method based on a clustering approach. However, these approaches are based on alternative variable-wise unfolding of the batch data that do not naturally capture the time-varying dynamics of the batch, but rather assume a constant covariance structure within each phase. The objective of these approaches is therefore to select phases in a way that the correlation structure among the data in the same phase has the minimum variation. For example, variables in the preheating step in a reactor and reaction step are likely to have different behavior and can be considered as different phases. The more suitable the phases that are selected, the fewer principal components are needed for the PCA model, which in turn means the fewer batch runs are needed in the training dataset. Furthermore, the size of the phases must not be too large or problems similar to the single phase approach may occur. The batch-wise unfolding approach used here will model all the time-varying covariance structure within each phase and so the selection of phases is much less critical. The main objective is to simply reduce the phase size to improve the local predictability of the models, to reduce the number of latent variables needed in the models (and thereby the number of batch runs required for identification), to minimize the ill-conditioning resulting from large time windows, and to reduce the computation time. In this work the number of phases selected was based on finding a balance between the number of observations and the number of required principal components in each phase. Studies by

the authors on changing the number of phases has shown that it does not have much effect on the resulting control as long as the number of latent variables necessary for adequately modeling each phase is smaller than the number of batches available in the training data set.

An important issue with such multiple models is the switching between models for two adjacent phases (bumpless transfer). This is easily achieved by the use of overlapping data at the beginning and end of each phase as illustrated in Fig. 2.4. This overlap achieves two benefits. It allows for consistency of the models between the end of one phase and the beginning of the next phase. This results because the same overlapping data are used for modeling the end of one phase and the beginning of the next. Furthermore, the overlapping allows one to continue to use consistent future prediction and past horizons (*fh* and *ph*) for the MPC. Consider the case where there is no overlap between phases and each phase is treated separately. When approaching the end of a phase there is a shrinking model future horizon and as the phase progresses the stability of the closed-loop system degrades as the available control and prediction horizons shrink. On the other hand, when a new phase starts there is not enough past information to have an efficient score estimation step. This situation makes the LV-MPC inefficient as the role of score estimation is critical. Making models for overlapping phases will remove these two problems.

After determining the borders of the phases, one should use data over as many sample times as the selected model future horizon (*fh*) from the next phase and the data of as many sample times as the selected model past horizon (*ph*) from the previous phase as

shown in Fig. 2.4. Then the current phase must be augmented with these two wings and the PCA model must be built based on the augmented phase. Now the algorithm can switch between phases as soon as the batch reaches the sample time corresponding to the border of the original phases. As a result the algorithm will never face the expanding past horizon and shrinking future horizon except at the beginning and end of the batch, respectively. The values of the *fh* and *ph* depend on the type of process. The range 10-30 sample times is typical, based on the authors' experience.

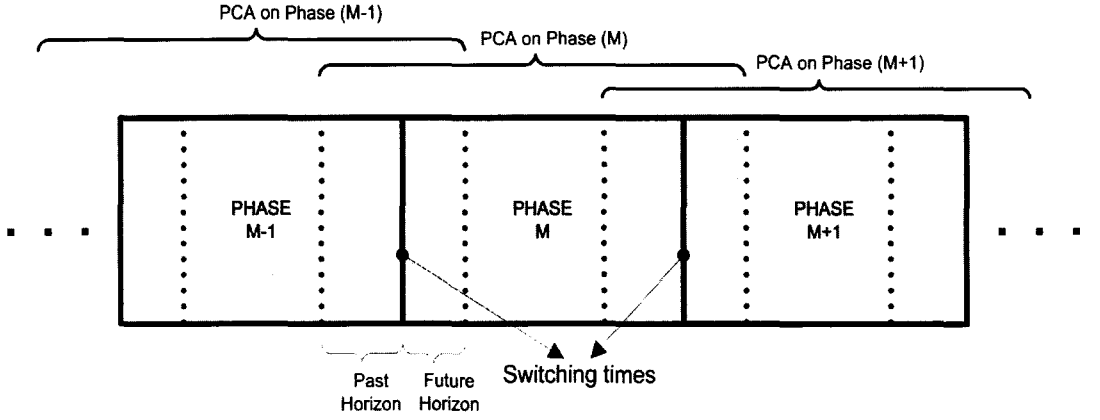


Fig. 2.4: Multiphase construction and overlapping in a batch-wise unfolded dataset

2.3.2. Control

McAvoy [35] proposed a PCA based MPC for continuous processes, and later Flores-Cerrillo and MacGregor [8] developed a version of LV-MPC for batch processes utilizing a PCA model of the process. The present work can be considered as an alternative to the work done by Flores-Cerrillo and MacGregor [8] in which the control formulation and modeling approaches are substantially revised. There are two major control approaches developed during this study, control based on the optimizing the final

scores in the PCA multi-phase model, τ, and control based on optimizing over the vector of future manipulated variables, u.

Control in the LV space

The objective of the control is to run a new batch to track certain trajectories and to simultaneously compensate for the effects of disturbances entering the batch. Suppose a multi-phase PCA model is developed based on a batch-wise unfolded dataset (Fig. 2.1). Each row (x^T) of the unfolded X matrix in Fig. 2.1 or Fig. 2.4 corresponds to the data from one complete batch or one complete batch phase, respectively.

$$x^T = [\zeta_1^T, \zeta_2^T, ..., \zeta_k^T, ..., \zeta_K^T] \tag{2.9}$$

where the data at any sample time k are defined by:

$$\zeta_k^T = [x_{me,k}^T, y_{cv,k}^T, u_{c,k}^T, y_{sp,k}^T] \tag{2.10}$$

The variables x_{me}, y_{cv}, u_c, and y_{sp} in (2.10) are measured variables, controlled variables, manipulated variables, and set-point variables respectively.

Assume that a new batch is currently at sample time k. The variables in that batch phase (x^T) can be rearranged into past and future terms according to whether the variables have already been observed or are known at time k or they are unknown future values as:

$$\begin{aligned} x_k^T = [&\zeta_j^T|_{j=1:k-1}, x_{me,k}^T, y_{cv,k}^T, y_{sp,k}^T \quad y_{sp,j}^T|_{j=k+1,...,K} \\ &u_{c,k}^T, u_{c,j}^T|_{j=k+1,...,K-1}, x_{me,j}^T|_{j=k+1,...,K} \quad y_{cv,j}^T|_{j=k+1,...,K}] \\ = [&x_{P1,k}^T, x_{P2,k}^T; x_{f1,k}^T, x_{f2,k}^T] \end{aligned} \tag{2.11}$$

where the subscript k denotes the separation made at time k. In (2.11) the rearranged rows of x^T are defined as follows: $x_{P1,k}^T = \left(\zeta_j^T \big|_{j=1:k-1}, x_{me,k}^T, y_{cv,k}^T, y_{sp,k}^T \right)$ and $x_{P2,k}^T = \left(y_{sp,j}^T \big|_{j=k+1:K} \right)$ are vectors of the known information at time k, while $x_{f1,k}^T = \left(u_{c,k}^T, u_{c,j}^T \big|_{j=k+1:K-1}, x_{me,j}^T \big|_{j=k+1:K} \right)$ and $x_{f2,k}^T = \left(y_{cv,j}^T \big|_{j=k+1:K} \right)$ are future data that are not known yet. K is the total number of sample times available in the phase. Separating the PCA loading vectors in the corresponding manner to the division of x, we have:

$$P_k = [P_{P1,k}; P_{P2,k}; P_{f1,k}; P_{f2,k}] \tag{2.12}$$

Note that since the algorithm is presented for online application, all of the variables mentioned in (2.11) and (2.12) change over time and have an index "k". However, for the sake of brevity the index "k" will be omitted. When all the information on a batch phase is complete, the corresponding final latent variable score for the observation at the end of the phase can be calculated from the PCA model for the phase as:

$$\tau^T = x_{P1}^T P_{P1} + x_{P2}^T P_{P2} + x_{f1}^T P_{f1} + x_{f2}^T P_{f2} \tag{2.13}$$

However, under MPC at the current time (k), the phase is not complete and the projected score at the end of the batch (phase), assuming no further adjustment to the control moves, must be estimated from only the data available up to and including the time step k using a missing data imputation method. This score prediction step and the corresponding predictions of the variable trajectories calculated from it play the same role as the state prediction algorithms in traditional MPC. The vector of imputed final scores for the phase, made based on data available at time k, is denoted by $\hat{\tau}_k$. A correction to

the score ($\Delta\hat{\tau}_k$) is then estimated by optimizing the MPC objective function to get a corrected final score:

$$\tau_{kc} = \hat{\tau}_k + \Delta\hat{\tau}_k \tag{2.14}$$

The objective function of the optimal control can be represented as follows:

$$\begin{aligned}\min_{\Delta\hat{\tau}_k} & \tfrac{1}{2}\left(\hat{y}_{cv} - y_{sp}\right)^T V_1\left(\hat{y}_{cv} - y_{sp}\right) + \tfrac{1}{2}\hat{u}_f^T V_2 \hat{u}_f \\ & = \tfrac{1}{2}\left(\hat{x}_{f2} - x_{P2}\right)^T V_1\left(\hat{x}_{f2} - x_{P2}\right) + \tfrac{1}{2}\hat{u}_f^T V_2 \hat{u}_f\end{aligned} \tag{2.15}$$

Now, Define:

$$\begin{aligned} x_P^T &= [x_{P1}^T \;\; x_{P2}^T], \\ x_f^T &= [x_{f1}^T \;\; x_{f2}^T] \\ P_p^T &= [P_{P1}^T \;\; P_{P2}^T], \\ P_f^T &= [P_{f1}^T \;\; P_{f2}^T] \end{aligned} \tag{2.16}$$

where V_1 and V_2 are LQ matrices. (V_1 is a diagonal positive definite matrix and V_2 is the derivative (positive semi-definite) matrix to put penalty on rate of change of input). The equation for the LV scores (2.2) can be written at time k for the current batch (phase) based on past and predicted future values as:

$$\hat{\tau}_k + \Delta\hat{\tau}_k = P_p{}^T \hat{x}_{p,k} + P_f{}^T \hat{x}_{f,k} \Rightarrow P_f{}^T \hat{x}_{f,k} = \hat{\tau}_k + \Delta\hat{\tau}_k - P_p{}^T \hat{x}_{p,k} \tag{2.17}$$

Then inverting this equation in a similar manner to that presented in Flores-Cerrillo and MacGregor [7] gives a prediction of the future trajectories taking into account continuity with past data, see Appendix A for details:

$$\begin{aligned} \hat{x}_{f,k} &= P_f \left(P_f^T P_f\right)^{-1}\left(\hat{\tau}_k + \Delta\hat{\tau}_k - P_p^T x_{p,k}\right) \\ [\hat{x}_{f1} \;\; \hat{x}_{f2}] &= \left[P_{f1} \;\; P_{f2}\right]\left(P_f^T P_f\right)^{-1}\left(\hat{\tau}_k + \Delta\hat{\tau}_k - P_p^T x_{p,k}\right) \end{aligned} \tag{2.18}$$

$$\hat{x}_{f2} = P_{f2}\left(P_f^T P_f\right)^{-1}\left(\hat{\tau}_k + \Delta\hat{\tau}_k - P_p^T x_{p,k}\right) \tag{2.19}$$

The future manipulated variable trajectory can be computed directly from the PCA model since no continuity is necessary for the input variables:

$$\hat{u_f} = P_{uf}\left(\hat{\tau}_k + \Delta\hat{\tau}_k\right) \tag{2.20}$$

Substituting (2.19) and (2.20) into the objective function (2.15) we get

$$\min_{\Delta\hat{\tau}_k} \tfrac{1}{2}\left\{\left[P_{f2}\left(P_f^T P_f\right)^{-1}\left(\hat{\tau}_k + \Delta\hat{\tau}_k - P_p^T x_p\right) - x_{P2}\right]^T V_1\left[P_{f2}\left(P_f^T P_f\right)^{-1}\left(\hat{\tau}_k + \Delta\hat{\tau}_k - P_p^T x_p\right) - x_{P2}\right]\right\}$$
$$+ \tfrac{1}{2}\left(P_{uf}\left(\hat{\tau}_k + \Delta\hat{\tau}_k\right)\right)^T V_2\left(P_{uf}\left(\hat{\tau}_k + \Delta\hat{\tau}_k\right)\right)$$

Or

$$\min_{\Delta\hat{\tau}_k} \tfrac{1}{2}\left[P_{f2}\left(P_f^T P_f\right)^{-1}\Delta\hat{\tau}_k - P_{f2}\left(P_f^T P_f\right)^{-1}\left(P_p^T x_p - \hat{\tau}_k\right) - x_{P2}\right]^T V_1$$
$$\left[P_{f2}\left(P_f^T P_f\right)^{-1}\Delta\hat{\tau}_k - P_{f2}\left(P_f^T P_f\right)^{-1}\left(P_p^T x_p - \hat{\tau}_k\right) - x_{P2}\right] +$$
$$\tfrac{1}{2}\left[P_{uf}\Delta\hat{\tau}_k + P_{uf}\hat{\tau}_k\right]^T V_2\left[P_{uf}\Delta\hat{\tau}_k + P_{uf}\hat{\tau}_k\right]$$

Therefore:

$$\min_{\Delta\hat{\tau}_k} \tfrac{1}{2}\left(A_1\Delta\hat{\tau}_k - b_1\right)^T V_1\left(A_1\Delta\hat{\tau}_k - b_1\right) + \tfrac{1}{2}\left(A_2\Delta\hat{\tau}_k + b_2\right)^T V_2\left(A_2\Delta\hat{\tau}_k + b_2\right)$$

where,

$$A_1 = P_{f2}\left(P_f^T P_f\right)^{-1}$$
$$b_1 = P_{f2}\left(P_f^T P_f\right)^{-1}\left(P_p^T x_p - \hat{\tau}_k\right) + x_{P2} \tag{2.21}$$
$$A_2 = P_{uf}$$
$$b_2 = P_{uf}\hat{\tau}_k$$

The above optimization problem can be solved analytically if there are no hard constraints or numerically if there are hard constraints on the manipulated variable, u_f,

which is written in terms of the decision variable in equation (2.20). In the unconstrained situation, taking the derivative of equation (2.21) and setting this equal to zero yields:

$$\begin{aligned}\Delta\hat{\tau}_k &= \left(A_1^T V_1 A_1 + A_2^T V_2 A_2\right)^{-1}\left(A_1^T V_1 b_1 - A_2^T V_2 b_2\right) \\ &= \left(\left[P_{f2}\left(P_f^T P_f\right)^{-1}\right]^T V_1\left[P_{f2}\left(P_f^T P_f\right)^{-1}\right] + P_{uf}^T V_2 P_{uf}\right)^{-1} \\ &\left(\left[P_{f2}\left(P_f^T P_f\right)^{-1}\right]^T V_1\left(P_{f2}\left(P_f^T P_f\right)^{-1}\left(P_p^T x_p - \hat{\tau}_k\right) + x_{P2}\right) - P_{uf}^T V_2 P_{uf}\hat{\tau}_k\right)\end{aligned} \tag{2.22}$$

Once the desired change in the latent variable for the end of the phase, $\Delta\hat{\tau}_k$, is calculated, the corrected score of the current batch is computed using equation (2.14) and the future control actions are computed from equation (2.20).

If the PCA model is built based on mean-centered and scaled data, equation (2.8) and all control computations up to equation (2.22) must also be considered for the mean-centered and scaled data. Thus, in order to obtain $\hat{u}_f$ in the original variable space, it should be scaled back. Then, its first element must be implemented on the process. At the next sample time the same procedure is repeated.

Control in the origina variable space

In the previous LV-MPC formulation the decision variables in the optimization were the LV scores at the end of the phase. Once optimal values of these were computed, the corresponding variable trajectories were computed from the PCA model. In this section a LV-MPC formulation is presented based on explicitly using a finite future trajectory of the manipulated variables as the decision variables, a formulation more common in the MPC literature.

The data vector for the current batch phase at time k (2.9) can be partitioned in a more explicit way with respect to the manipulated variable as follows:

$$\begin{aligned} x_k^T = [&\zeta_j^T|_{j=1:k-1}, x_{me,k}^T, y_{cv,k}^T, y_{sp,k}^T \quad y_{sp,j}^T|_{j=k+1,\ldots,k+PH} \\ &x_{me,j}^T|_{j=k+1,\ldots,K}, u_{c,j}^T|_{j=k+CH,\ldots,K}, y_{cv,j}^T|_{j=k+PH+1,\ldots,K}, y_{sp,j}^T|_{j=k+PH+1,\ldots,K} \\ &y_{cv,j}^T|_{j=k+1,\ldots,k+PH} \quad u_{c,j}^T|_{j=k,\ldots,k+CH-1}] \\ = [&x_{P1}^T, x_{P2}^T, x_{f1}^T, x_{f2}^T, u_f^T]_k \end{aligned} \tag{2.23}$$

where $x_{P1,k}^T = \left(\zeta_j^T|_{j=1:k-1}, x_{me,k}^T, y_{cv,k}^T, y_{sp,k}^T\right)$ and $x_{P2,k}^T = \left(y_{sp,j}^T|_{j=k+1,\ldots,k+PH}\right)$ are vectors of the known information at time k, while $x_{f1,k}^T = \left(x_{me,j}^T|_{j=k+1,\ldots,K}, u_{c,j}^T|_{j=k+CH,\ldots,K}, y_{cv,j}^T|_{j=k+PH+1,\ldots,K}, y_{sp,j}^T|_{j=k+PH+1,\ldots,K}\right)$, $x_{f2,k}^T = \left(y_{cv,j}^T|_{j=k+1,\ldots,k+PH}\right)$ and $u_{f,k}^T = \left(u_{c,j}^T|_{j=k,\ldots,k+CH-1}\right)$ are future data that are not known yet. CH and PH are the control horizon and prediction horizon respectively. The main point of this method is to formulate the problem in terms of future manipulated variables, u_f. At the sample time k, the known data are x_{P1}, x_{P2}, the unknown data are x_{f1}, x_{f2}, and the future decision variable is u_f. However, the term u_f is considered as the decision variable that will be determined through the optimization process. In this algorithm, the final batch phase score is again estimated using a weighted version of one of the missing data imputation methods discussed in Section 2.4 where the existing data is taken as $z^* = x_p$ and recent data are more heavily weighted [36]:

$$x_P^T = [x_{P1}^T \; x_{P2}^T \; \hat{u}_f], \quad P_p^T = [P_{P1}^T \; P_{P2}^T \; P_{uf}^T] \tag{2.24}$$

Either of the missing data imputations can be expressed as:

$$\hat{\tau}_k^T = x_p^T \beta_k \tag{2.25}$$

Once the scores are estimated the future process variables can be estimated as well from the PCA model:

$$\hat{x}_{f2} = P_{f2}\hat{\tau}_k \tag{2.26}$$

Combining (2.24), (2.25) and (2.26) gives:

$$\hat{x}_{f2} = C_k x_p, \text{ where } C_k = P_{f2}\beta_k^T \tag{2.27}$$

$$\hat{x}_{f2} = C_{P1,k}x_{P1} + C_{P2,k}x_{P2} + C_{uf,k}\hat{u}_f \tag{2.28}$$

The $C_{P1,k}$, $C_{P2,k}$, $C_{uf,k}$ come from partitioning the matrix C_k corresponding to the vector x_p. Now the predicted future controlled variable ($\hat{x}_{f2}$) is expressed as a function of future MV's (u_f) input. The following optimal control problem is considered:

$$\begin{aligned}\min_{\hat{u}_f} \tfrac{1}{2}\left(\hat{y}_{cv} - y_{sp}\right)^T V_1 \left(\hat{y}_{cv} - y_{sp}\right) + \tfrac{1}{2}\hat{u}_f^T V_2 \hat{u}_f \\ = \tfrac{1}{2}\left(\hat{x}_{f2} - x_{P2}\right)^T V_1 \left(\hat{x}_{f2} - x_{P2}\right) + \tfrac{1}{2}\hat{u}_f^T V_2 \hat{u}_f\end{aligned} \tag{2.29}$$

where, V_1 and V_2 are the LQ weights. Combining (2.28), (2.29), yields:

$$\begin{aligned}\min_{\hat{u}_f} \tfrac{1}{2}\left(C_{P1,k}x_{P1} + C_{P2,k}x_{P2} + C_{uf,k}\hat{u}_f - x_{P2}\right)^T V_1 \left(C_{P1,k}x_{P1} + C_{P2,k}x_{P2} + C_{uf,k}\hat{u}_f - x_{P2}\right) \\ + \tfrac{1}{2}\hat{u}_f{}^T V_2 \hat{u}_f\end{aligned} \tag{2.30}$$

Once again the above optimization problem can be solved either analytically (if there are no hard constraints) or numerically (if there are hard constraints on MV).

$\hat{u}_f$ is composed of the optimal input values of the current time up to the control horizon. According to the MPC rule, the first element of $\hat{u}_f$ is implemented on the plant and at the next sample time, the same computation is repeated. The control horizon (CH)

should be large enough to assure the process is controllable in the selected horizon. The concepts of controllability and observability in the PCA model are discussed in Appendix B.

2.4. Case Study: Temperature Control in a Batch Reactor

Aziz et al.[37] presented a nonlinear model of a batch reactor. This process model was originally proposed by Cott and Macchieto[38] as a case study for a temperature control problem on a batch reactor. The schematic of the reactor is shown in Fig. 2.5. The model for this process is presented in Appendix C. Our use of that model in this section is simply to illustrate the trajectory tracking capabilities and properties of the proposed LV-MPC algorithms using an arbitrarily complex set-point trajectory. We are not trying to find an optimal set-point trajectory as was the goal of the earlier publications that used this model. The objective is to control the reactor temperature to track the specified set-point trajectory and to do so in the face of non-stationary disturbances. The manipulated variable is the set point of the jacket temperature (Fig. 2.5). Once the set point is calculated by the controller, the desired jacket temperature is generated immediately, by combination of hot and cold water. However, it takes time for the jacket temperature (T_j) to achieve the T_j^{sp} (input). It is assumed to have linear dynamics:

$$\frac{dT_j}{dt} = \frac{\left(T_j^{sp} - T_j\right)}{\tau_j} - \frac{Q_j}{V_j \rho_j C_{pj}} \tag{2.31}$$

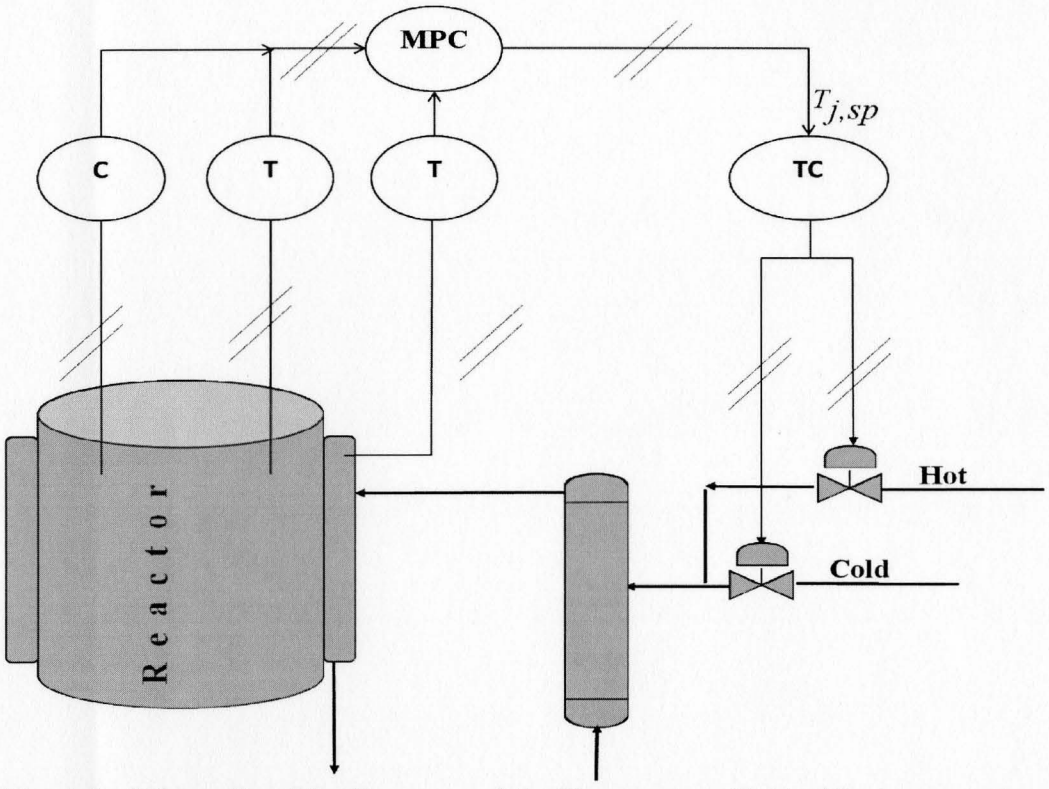


Fig. 2.5: Schematic of the Reactor and the Temperature Control Instrumentation

The total batch time is 120 min. Control based on two sampling intervals 0.1 min and 0.2 min are tested during this study. For the sake of improving identifiability, a RBS dither signal is superimposed on top of the manipulated variable. The minimum switching frequency of the RBS is selected to be between $1/(0.15+0.1\times randn)$ and $1/(0.1+0.1\times randn)$ where "*randn*" indicates "random number". (1/0.15 =6.67 means that the signal remains constant over at least 6-7 sample times which is suitable with respect to the process time constant). The dither magnitude (± 6%) on the input was small enough to have little noticeable effect on the temperature trajectories. White noise was added to the measurements and different starting points were randomly selected. For this control study data from 45 batches (40 batches with similar set-point trajectories plus 5

batches with a slightly different but in the range set-point trajectories) run under closed-loop control with a PI controller and added RBS dither signal were used to develop the PCA models.

2.4.1. Trajectory tracking

The first step is to identify the phases along the batch. Based on the loading plots of the PCA model and considering the trajectory of the set point, six phases are selected along the batch. These phases are constructed on the intervals of: 1-30, 30-50, 50-60, 60-80, 80-100, and 100-120. The phases were selected in a way that less variation is observed in the correlation structure (loading plots) of the resulting PCA model in each phase and a balance exists between the number of observations and the number of required PCs. However, for smooth transition between two adjacent phases overlap should be considered between phases (see Fig. 2.4) in the modeling step. In the modeling and control studies in this section overlaps of $ph = 20$ *and* $fh = PH$ intervals with both the immediate past phase and the next phase are included when modeling each current phase. For example the phase 2 model uses data over the time intervals $30\text{-}T\times ph$ to $50+T\times fh$, where T represents the process sample time. The following figures illustrate the results of various LV-MPC simulations.

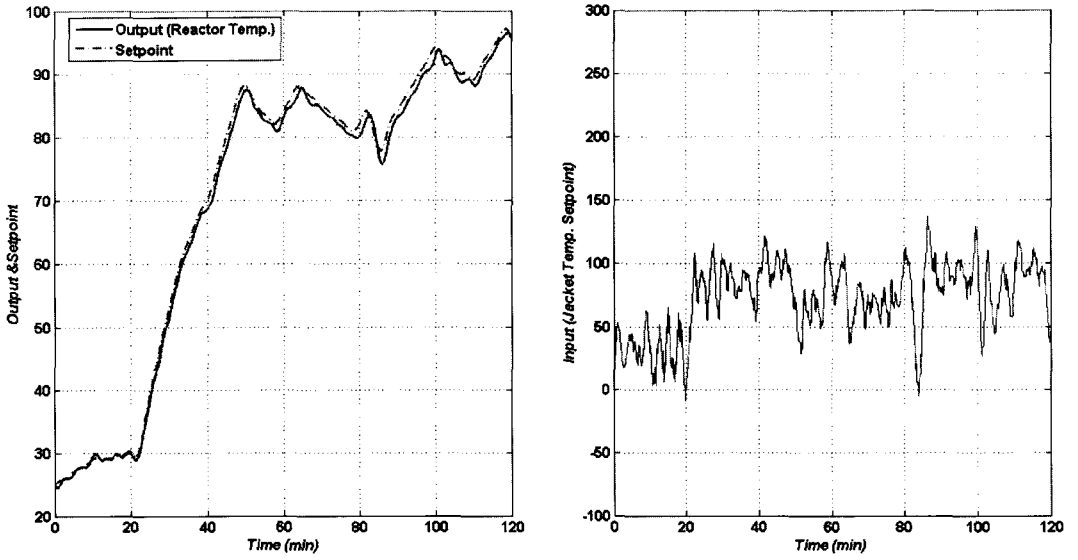


Fig. 2.6: Control in the latent variable space using PCA models based on the multiphase batch-wise unfolded dataset with 6 phases. Each variable is centered by its own mean in the batch-wise unfolded dataset.

Fig. 2.6 shows the trajectory tracking of the Multi-phase LV-MPC (MLV-MPC) using the control strategy based on scores. There are clear and persistent deviations throughout the batch between the CV and the set-point that were eventually shown not to be due to the control, but rather due to the model centering and scaling issues. Indeed, the control objective function is evaluated in the mean-centered and auto-scaled space and finally all variables are scaled back to original space. In the case of the study in Fig. 2.6, every variable was centered around its mean trajectory and scaled by its standard deviation from the identification dataset. However, since the set-point trajectory is a series of ramp changes and the data are collected from a closed-loop system using a PI controller, a persistent tracking offset exists in the identification dataset. Thus, the mean trajectory of the CV is not centered about the set-point in the dataset. Therefore, when the deviation variables were converted back into the original variables after the MPC

calculations in the mean centered space this unseen offset was still present. However, by simply centering the CV about the desired set-point trajectory, this problem disappears. The offset is captured within the PCA model and removed by the MPC. In this case the CV is still scaled by its standard deviation about its mean trajectory. The result of simply re-centering the CV in this way and re-running the simulation of Fig. 2.6 is shown in Fig.2.7. It is evident that trajectory tracking is much better and the persistent offsets have disappeared.

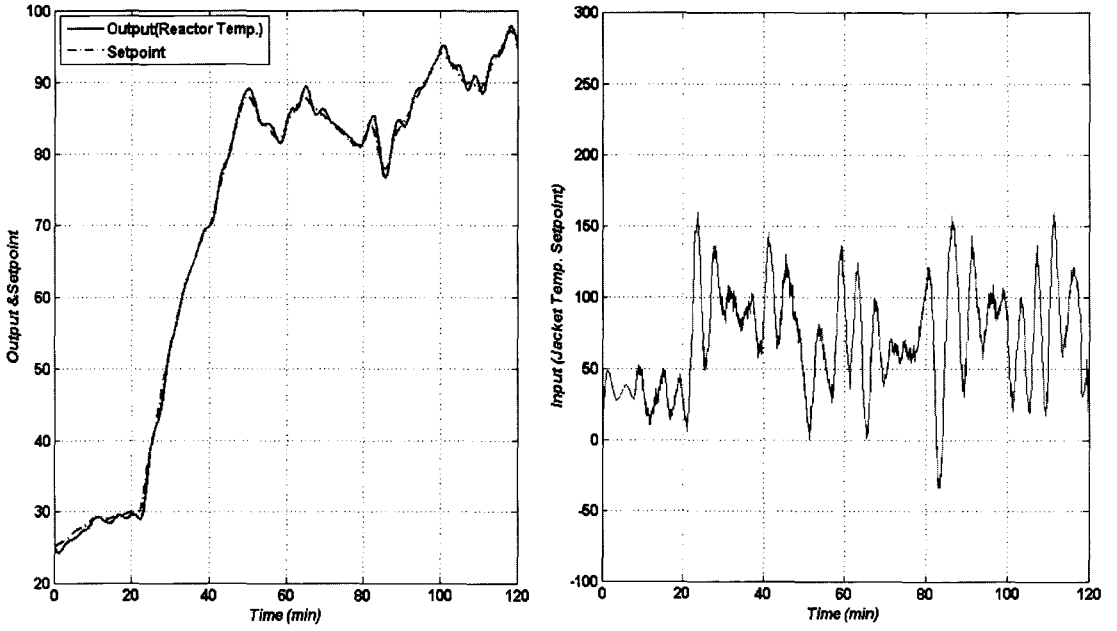


Fig. 2.7: Control in the latent variable space using PCA models based on the multiphase batch-wise unfolded dataset as in Fig. 2.6, but where y is now mean-centered by y_{sp}

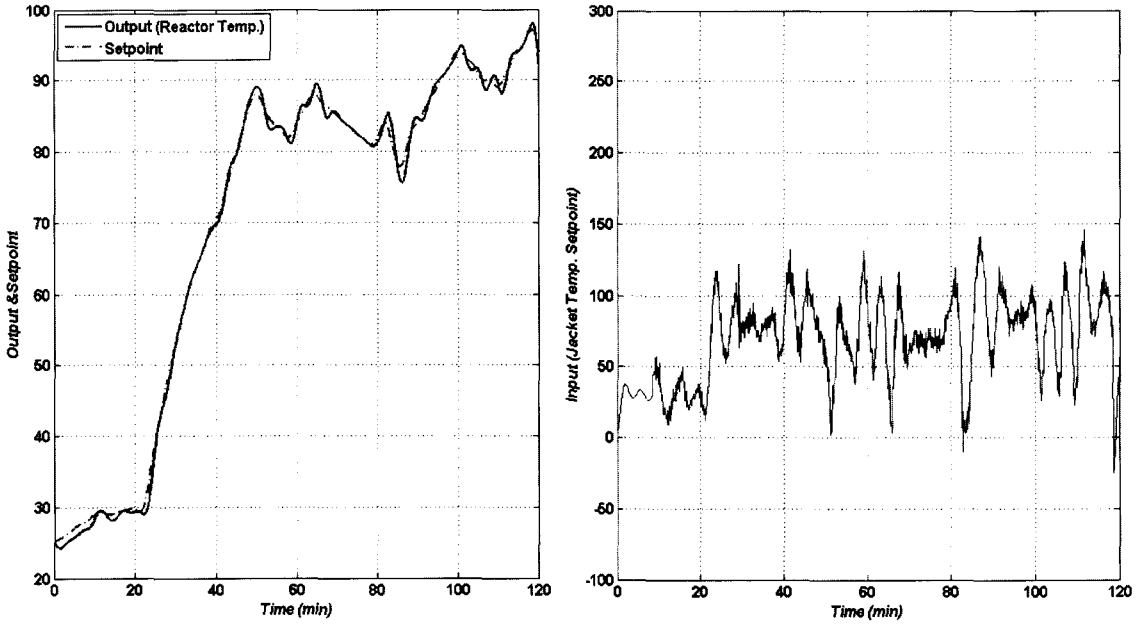


Fig. 2.8: Control in the original MV space using Multiphase batch-wise unfolded dataset with 6 phases along the batch, where y is mean-centered by y_{sp}, CH=20 and PH=60

Fig. 2.8 shows the same example, but solved by the second control methodology (explicit in u_f). Mean centering y by y_{sp} is again done here (the same offset problems appear again if this is not done).

It is observed that both control algorithms have similar performances in trajectory tracking. However, there are some interesting differences between the two proposed control formulations. One difference relates to the effective control horizon. Batch processes are of finite duration and so classical infinite dimensional LQ control is not possible. However, if we define "infinite horizon" control in a batch process to be control that always uses the (shrinking) horizon until the end of the phase, then the control algorithm based on latent variable scores (τ) is an "infinite horizon" controller. It computes at every time interval (k) the adjustment (equation 2.20) needed in the latent

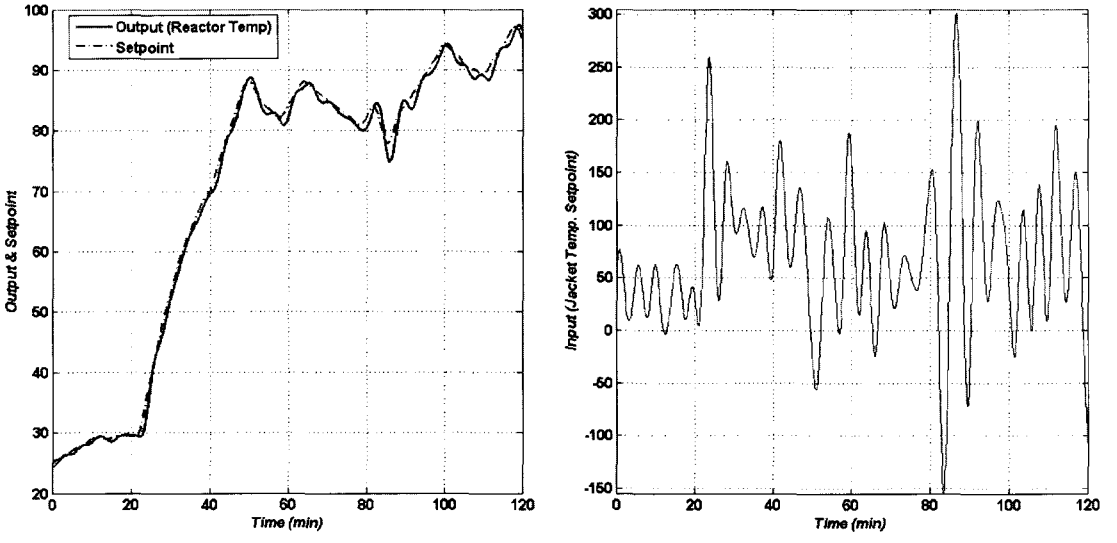
variable scores by the end of the phase in order to optimize the objective function (2.15). This LV score vector carries the information about future CV and MV trajectories until the end of the batch phase. In fact choice of phase size is an indirect way of selection of the model and control horizon in the control algorithm based on scores. The estimated MV trajectory is then computed from the optimal final scores using the LV model (equation 2.20). On the other hand, the MPC based on optimization in the MV space is very clearly a finite horizon controller that computes only *CH* future control actions. To implement an "infinite horizon" controller in the latter approach by using a large but shrinking *CH* horizon would lead to a much higher dimensional optimization problem. The "infinite horizon" in the control approach based on "τ" is partially responsible for the added smoothness in the optimal MV trajectories computed by the LV score approach versus the control directly in the MV space (compare MV trajectory plots in Figs. 2.7 and 2.8). Another reason for getting smoother input in the control based on the scores is that the model uses only the dominant latent variables and the inversion filters out many of the higher frequency variations that might appear in a full inversion.

In order to benchmark the proposed LV-MPC methodologies against commonly used controllers, the performance of a PI controller with two sets of tunings (tightly and loosely tuned) is presented in Fig. 2.9. Furthermore, Table 2.1 summarizes the performance indices of different control strategies including the proposed two LV-MPC methodologies, the PI controllers with different tunings, and a Nonlinear MPC (NMPC) [39]. The LV-MPC uses the PCA model built on a dataset generated by the tightly tuned PI controller. Table 2.1 shows that the LV-MPC designed in the latent space has a better

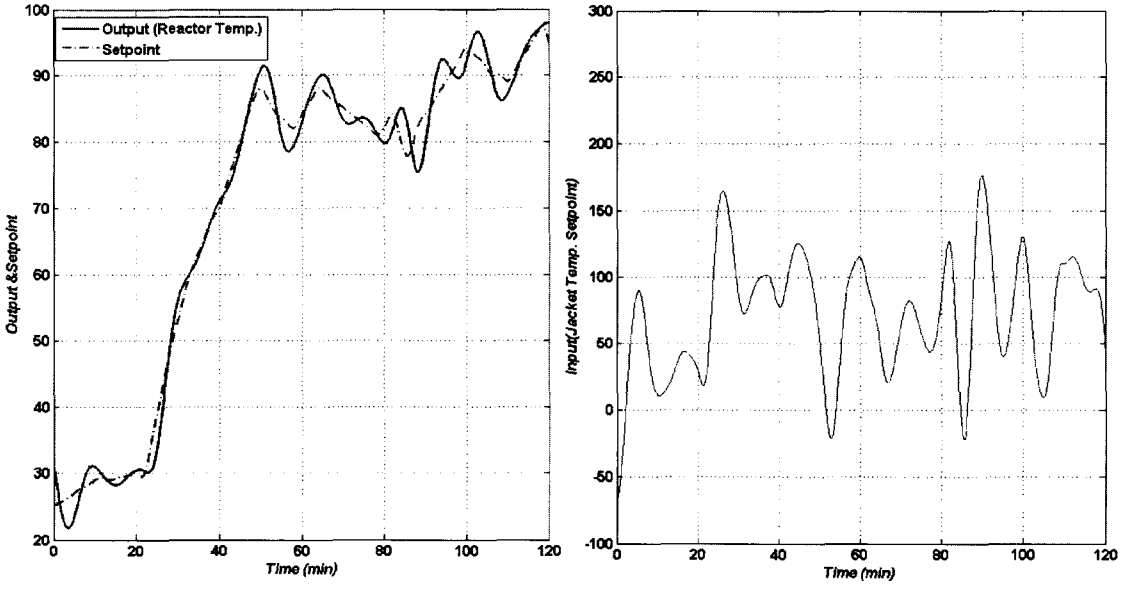
set-point tracking and also needs less change in the manipulated variable as compared to the LV-MPC designed in the original MV space. These properties are mainly due to the fact that a larger control horizon is inherent for the control in the latent space (infinite horizon control). However, for the control algorithm in the original MV space one can choose whatever control horizon is needed to manage the aggressiveness of the controller. If the process has disturbances that may have long term effects, it would be more desirable to have a longer control horizon and smoothly remove the disturbance. In such a situation, the control algorithm based on scores is more attractive since having a large control horizon significantly increases the computational burden for the control algorithm in the original MV space (based on u_f). On the other hand, if the disturbances have fast effects on the process, the controller has to be more aggressive and have a faster action. In this situation, the control algorithm in the original MV space can be more helpful where the control horizon can be selected arbitrarily small. Based on using an exact model for the process, NMPC has the best tracking quality in the sense of Root Mean Square Error (RMSE) and STandard Deviation (STD) of the rate of change in the input variable. However, since the applicability of the LV-MPC is in the cases where the exact mechanistic model is not in hand, the comparison of the LV-MPC, against the NMPC with an exact model is not a fair comparison. Thus, a model mismatch is introduced to the NMPC algorithm and the results are shown in the last column of Table 2.1. The mismatch is introduced in two parameters out of more than 15 available parameters in the simulator. 30% mismatch is considered for the time constant of the dynamics in the jacket temperature and for the heat capacity of the reactor contents. It is

seen that the required rate of change of manipulated variable is now more than that in both LV-MPC algorithms, and the tracking quality is worse as compared to both LV-MPC algorithms. Clearly, the comparison will depend upon the mismatch in the theoretical model used for the NMPC, but this comparison with the NMPC and PI controllers is included just to provide some relative benchmarks.

An implementation problem that often exists with the NMPC is the computation time. The NMPC results presented in Table 2.1 are based on the control and prediction horizon of four sample times which is a small horizon. However, even with this small control horizon, the computation time for the NMPC is about twice that of the LV-MPC in the latent space (which is an infinite horizon control) and is of the same order of time compared to the LV-MPC in the original MV space based on 20 sample times for the control horizon and 60 sample times for the prediction horizon.



(a)



(b)

Fig. 2.9: a) Implementation of a tightly tuned PI controller for the temperature control problem, b) Implementation of a loosely tuned PI controller for the temperature control problem.

Table 2.1- Comparison of different control algorithms for the temperature control problem

Performance criteria	Control based on T	Control Based on u	Tight PI controller	Loose PI controller	NMPC with Perfect Model	NMPC with Model Mismatch
RMSE of "y-ysp"	0.7795	0.9068	1.1540	2.4327	0.1168	1.4366
STD "Δu"	6.2459	9.0339	6.7139	2.2809	4.5246	9.2367

2.4.2. Disturbance Rejection

It is important to check the power of disturbance rejection and offset elimination for the proposed control methodologies, in particular, the ability of the controller to incorporate integral action to offset the effect of non-stationary disturbances. In all optimal controllers, the integral action results from incorporating into the model (or in the MPC prediction updating) adequate information about the real disturbances affecting the process. If appropriate non-stationary disturbance models are used in designing the

controller, then appropriate offset-free tracking should be achieved [3],[2],[40]. This integral action comes from the inclusion of non-stationary states into the model thereby giving rise to the ability of the state estimator to predict future offsets resulting from the appearance of these disturbances. In the LV-MPC methodology proposed here, offset is handled automatically since information on the non-stationary effects of the disturbances are built into the latent variable PCA model developed from the training data (historical plus designed experiments). These models include the measured controlled variables (y) as well as all other measured variables (e.g. x_{me}) and so the effects of all disturbances showing up in any of these measurements are modeled by some of the latent variables. It is observed in this study that if non-stationary effects are present, as external disturbances or as persistent offsets in the closed-loop training data, then some of the identified latent variables will be non-stationary in nature. As a result, the future predictions of the trajectories using the missing data imputation algorithms (the equivalent of state estimators in this methodology) will appropriately reflect drifts due to these disturbances. In the set-point tracking studies shown earlier no offsets were apparent even though the set-point trajectories were a sequence of ramps. This is because the models and controllers are built on the variable deviations about the mean trajectories and the deviation MV's calculated by the MPC are then added back onto the mean MV trajectory to get the final MV setting. Thus the dominant ramp behavior of the CV set-point is tracked with this feedforward application of the mean MV trajectory. However, the LV-MPC removes all other offset about the set-point through integral action resulting from latent variables capturing the non-stationarities in the training data (non-stationarities

resulting from persistent offsets remaining from the PI controller and batch to batch disturbances in the initial conditions).

However, to provide a more severe test of the disturbance rejection ability of the batch LV-MPC, a very large additional random walk disturbance [41] was superimposed upon the measured temperature (the CV) for several simulation runs and the ability of the LV-MPC to eliminate the large offsets coming from this disturbance was investigated. (The stochastic random walk disturbance is the equivalent of randomly occurring step changes from the point of view of control [42]). This study was not intended to represent reality since such a noisy, non-stationary disturbance, unfiltered by passage through any part of the system, and appearing in no other measured variable would probably never occur in practice. The study is intended only as a severe test of the ability of the LV-MPC to eliminate offset due to non-stationary disturbances.

Figs. 2.10 and 2.11 show the random walk disturbances superimposed directly on the CV. If there was no offset elimination (integral action) then the 10-20°C offsets would appear in the controller's tracking of the set-point trajectory. However, the proposed control methodologies clearly are able to reject the non-stationary disturbances. It is seen that the disturbance rejection and offset-free set-point tracking performance of the two proposed algorithms are good. Nonetheless, the input variable is aggressive. This is mainly because of the severity of the test disturbance.

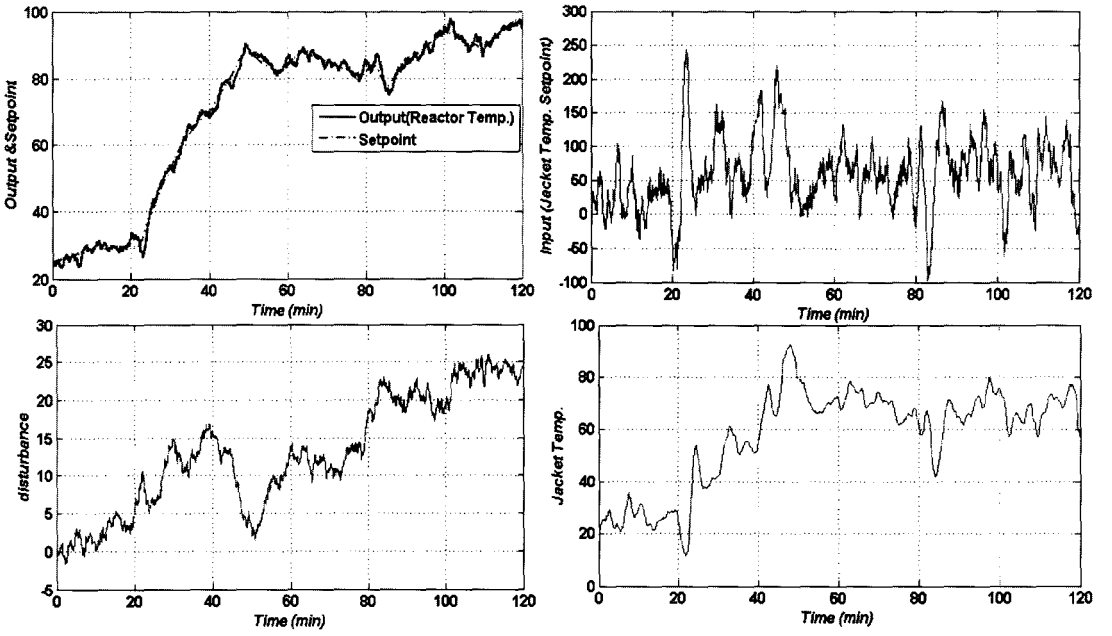


Fig. 2.10: Performance of the LV-MPC using Control in the latent variable space for both tracking the set-point trajectory and rejecting a random walk disturbance occurring on top of the output. Multiphase PCA models, based on the batch-wise unfolded dataset with 6 phases are used.

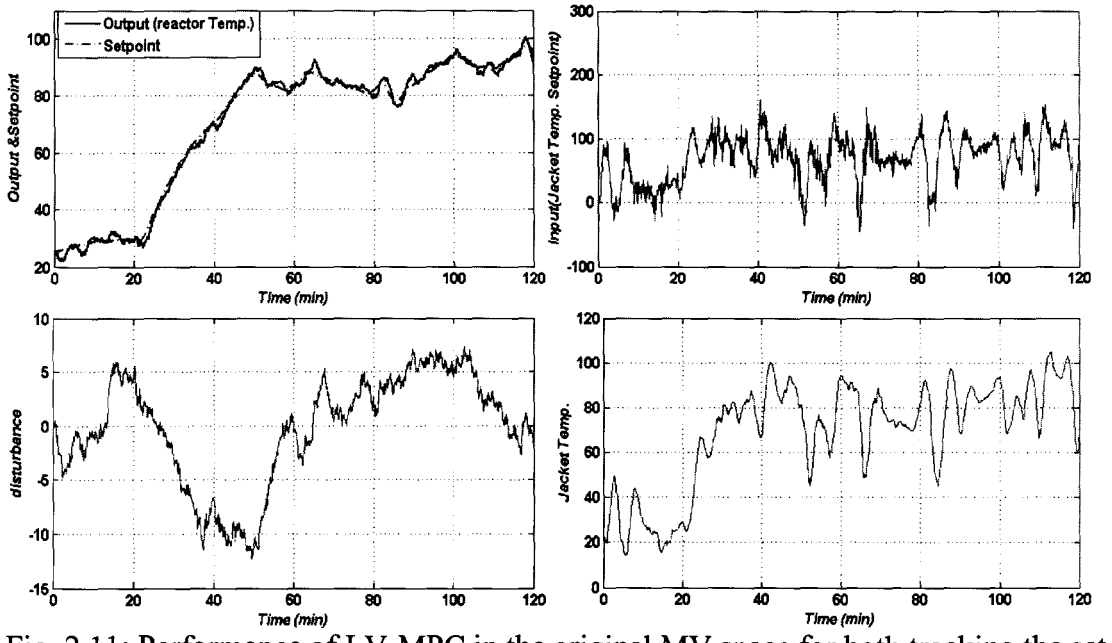


Fig. 2.11: Performance of LV-MPC in the original MV space for both tracking the set-point trajectory and rejecting a random walk disturbance occurring on top of the output. Control and prediction horizons *of CH*=20 and *PH*=60 are used along with multiphase PCA models based on the batch-wise unfolded dataset with 6 phases.

2.5. Conclusion

Two Latent variable Model Predictive Control (LV-MPC) methodologies are proposed for combined trajectory tracking and disturbance rejection in batch processes. The methods are based on latent variable models (developed using PCA in this chapter, but other latent variable modeling approaches could be used). These models are built from batch-wise unfolding of historical closed-loop batch data and from batch data collected from closed-loop designed experiments. Overlapping multi-models are proposed to be used for different phases of the batch (multiphase models) in order to focus more on the local behavior of the trajectories, improve conditioning of the matrices in the MPC solutions, and simplify the online computations. These batch-wise unfolded LV models are capable of capturing the time-varying and nonlinear behavior in batch processes.

The two control methodologies are based (i) on solving the control problem in the space of the latent variables to find the optimal latent variable scores and then computing (from the LV model) the manipulated variable trajectories corresponding to these optimal final LV scores (an "infinite horizon" controller), and (ii) on solving directly for the manipulated variable trajectories over a finite horizon ("finite horizon" controller). For the prediction step, missing data imputation methods for LV models are used, providing the equivalent of state estimators.

The methods are tested for both trajectory tracking and disturbance rejection using simulations on an exothermic batch reactor. Both control algorithms considerably

outperform the PI controller and give good offset free performance when tracking complex set-point temperature trajectories both with and without the presence of non-stationary disturbances.

References

[1] T.J. Crowley, E.S. Meadows, E. Kostoulas, and F.J. Doyle III, "Control of Particle Size Distribution described by a Population Balance Model of Semibatch Emulsion Polymerisation," *Journal of Process Control*, vol. 10, 2000, pp. 419-432.

[2] D.J. Kozub and J.F. MacGregor, "Feedback control of polymer quality in semi-batch copolymerization reactors," *Chemical Engineering Science*, vol. 47, 1992, pp. 929-942.

[3] D.J. Kozub and J.F. MacGregor, "State estimation for semi-batch polymerization reactors," *Chemical Engineering Science*, vol. 47, Apr. 1992, pp. 1047-1062.

[4] Y. Yabuki and J.F. MacGregor, "Product quality control in semibatch reactors using midcourse correction policies," *Industrial & Engineering Chemistry Research*, vol. 36, 1997, pp. 1268–1275.

[5] Y. Yabuki, T. Nagasawa, and J.F. MacGregor, "Industrial experiences with product quality control in semi-batch processes," *Computers & Chemical Engineering*, vol. 26, Feb. 2002, pp. 205-212.

[6] J. Flores-Cerrillo and J.F. MacGregor, "Control of particle size distributions in emulsion semibatch polymerization using mid-course correction policies," *Industrial and Engineering Chemistry Research*, vol. 41, 2002, pp. 1805-1814.

[7] J. Flores-Cerrillo and J.F. MacGregor, "Control of batch product quality by trajectory manipulation using latent variable models," *Journal of Process Control*, vol. 14, 2004, pp. 539-553.

[8] J. Flores-Cerrillo and J.F. MacGregor, "Latent variable MPC for trajectory tracking

in batch processes," *Journal of Process Control*, vol. 15, Sep. 2005, pp. 651-663.

[9] B. Joseph and F.W. Hanratty, "Predictive control of quality in a batch manufacturing process using artificial neural network models," *Industrial and Engineering Chemistry Research*, vol. 32, 1993, pp. 1951-1961.

[10] R. Cardello and K. San, "Application of gain scheduling to the control of batch bioreactors," *Proceedings of the 1987 American Control Conference, 10-12 June 1987*, Green Valley, AZ, USA: American Autom. Control Council, 1987, pp.682-6.

[11] M.R. Juba and J.W. Hamer, "Progress and challenges in batch process control," York, NY, USA: Elsevier Science Publ, Amsterdam, Neth & New, 1986, pp. 139-183.

[12] T. Clarke-Pringle and J.F. MacGregor, "Nonlinear adaptive temperature control of multi-product, semi-batch polymerization reactors," *Computers & Chemical Engineering*, vol. 21, 1997, pp. 1395–1409.

[13] C. Kravaris, R.A. Wright, and J.F. Carrier, "Nonlinear controllers for trajectory tracking in batch processes," *Computers & Chemical Engineering*, vol. 13, 1989, pp. 73-82.

[14] C.E. Garcia, M.D. Prett, and M. Morari, "Model predictive control: Theory and practice - A survey," *Automatica*, vol. 25, 1989, pp. 335-348.

[15] F. Martinsen, L. Biegler, and B. Foss, "A new optimization algorithm with application to nonlinear MPC," *Modeling, Identification and Control*, vol. 26, 2005, pp. 3-22.

[16] M. Cannon, Weiheng Liao, and B. Kouvaritakis, "Efficient MPC optimization using

Pontryagin's minimum principle," *International Journal of Robust and Nonlinear Control*, vol. 18, May. 2008, pp. 831-44.

[17] D.J. Louwerse and A.K. Smilde, "Multivariate statistical process control of batch processes based on three-way models," *Chemical Engineering Science*, vol. 55, Apr. 2000, pp. 1225-1235.

[18] J. Camacho and J. Picó, "Multi-phase principal component analysis for batch processes modelling," *Chemometrics and Intelligent Laboratory Systems*, vol. 81, Apr. 2006, pp. 127-136.

[19] P. Nomikos, "Multi-variate statistical process control of batch processes," Ph.D. Thesis, McMaster University, 1995.

[20] P. Nomikos and J.F. MacGregor, "Multivariate SPC charts for batch processes," *Technometrics*, vol. 37, 1995, pp. 41-59.

[21] P. Nomikos and J.F. MacGregor, "Multi-way partial least squares in monitoring batch processes," *Chemometrics and Intelligent Laboratory Systems*, vol. 30, Nov. 1995, pp. 97-108.

[22] P. Nomikos and J.F. MacGregor, "Monitoring batch processes using multiway principal component analysis," *AIChE Journal*, vol. 40, 1994, pp. 1361-1375.

[23] J. Camacho, J. Picó, and A. Ferrer, "The best approaches in the on-line monitoring of batch processes based on PCA: Does the modelling structure matter?," *Analytica Chimica Acta*, vol. 642, May. 2009, pp. 59-68.

[24] M. Golshan, J.F. MacGregor, M.J. Bruwer, and P. Mhaskar, "Latent Variable MPC for trajectory tracking in batch processes: role of model structure," *Proceedings of*

American Control Conference, St. Louis, USA: 2009, pp. 4779-4784.

[25] S. Wold, N. Kettaneh, H. Fridén, and A. Holmberg, "Modelling and diagnostics of batch processes and analogous kinetic experiments," *Chemometrics and Intelligent Laboratory Systems*, vol. 44, Dec. 1998, pp. 331-340.

[26] J.A. Westerhuis, T. Kourti, and J.F. MacGregor, "Comparing alternative approaches for multivariate statistical analysis of batch process data," *Journal of Chemometrics*, vol. 13, 1999, pp. 397-413.

[27] S. Garcia-Munoz, T. Kourti, J.F. MacGregor, A.G. Mateos, and G. Murphy, "Troubleshooting of an Industrial Batch Process Using Multivariate Methods," *Industrial & Engineering Chemistry Research*, vol. 42, Jul. 2003, pp. 3592-3601.

[28] P. Nomikos, "Detection and diagnosis of abnormal batch operations based on multi-way principal component analysis World Batch Forum, Toronto, May 1996," *ISA Transactions*, vol. 35, 1996, pp. 259-266.

[29] D. Neogi and C.E. Schlags, "Multivariate Statistical Analysis of an Emulsion Batch Process," *Industrial & Engineering Chemistry Research*, vol. 37, Oct. 1998, pp. 3971-3979.

[30] Y. Zhang and M.S. Dudzic, "Industrial application of multivariate SPC to continuous caster start-up operations for breakout prevention," *Control Engineering Practice*, vol. 14, Nov. 2006, pp. 1357-1375.

[31] T. Kourti and J.F. MacGregor, Technical Report, Department of Chemical Engineering, McMaster University, Hamilton, ON, Canada.

[32] P.R.C. Nelson, P.A. Taylor, and J.F. MacGregor, "Missing data methods in PCA

and PLS: Score calculations with incomplete observations," *Chemometrics and Intelligent Laboratory Systems*, vol. 35, Nov. 1996, pp. 45-65.

[33] F. Arteaga and A. Ferrer, "Dealing with missing data in MSPC: several methods, different interpretations, some examples," *Journal of Chemometrics*, vol. 16, 2002, pp. 408-418.

[34] C. Zhao, F. Wang, Z. Mao, N. Lu, and M. Jia, "Improved Batch Process Monitoring and Quality Prediction Based on Multiphase Statistical Analysis," *Industrial & Engineering Chemistry Research*, vol. 47, Feb. 2008, pp. 835-849.

[35] T. McAvoy, "Model predictive statistical process control of chemical plants," *Proceedings of the 15th IFAC World Congeress*, Barcelona, Spain: 2002, pp. 3876-3881.

[36] M.J. Bruwer, Internal Report, ProSensus Inc., Hamilton, ON, Canada 2007.

[37] N. Aziz, M.A. Hussain, and I.M. Mujtaba, "Performance of different types of controllers in tracking optimal temperature profiles in batch reactors," *Computers & Chemical Engineering*, vol. 24, Jul. 2000, pp. 1069-1075.

[38] B.J. Cott and S. Macchietto, "Temperature control of exothermic batch reactors using generic model control," *Industrial & Engineering Chemistry Research*, vol. 28, 1989, pp. 1177-1184.

[39] S.L. DeOliviera and M. Morari, "Robusr model predictive control for nonlinear systems," *Proceedings of 33rd IEEE Conference on Decision and Control*, Lake Buena Vista, FL, USA: 1994, pp. 3561-3566.

[40] T.J. Harris, F. Boudreau, and J.F. Macgregor, "Performance assessment of

multivariable feedback controllers," *Automatica*, vol. 32, Nov. 1996, pp. 1505-1518.

[41] G. Box, G.M. Jenkins, and G. Reinsel, *Time Series Analysis: Forecasting & Control*, Prentice Hall, 1994.

[42] J. MacGregor, T. Harris, and J. Wright, "Duality between the control of processes subject to randomly occuring deterministic disturbances and ARIMA stochastic disturbances," *Technometrics*, vol. 26, 1984, pp. 389-397.

CHAPTER 3

Latent Variable Model Predictive Control for Trajectory Tracking in Batch Processes: Alternative Modeling Approaches

Contents of this chapter have been published in two conference proceedings: American Control Conference (ACC), July 2009, and 9th International Symposium on Dynamics and Control of Process Systems (DYCOPS), July 2010. A journal paper containing the materials of this chapter will be submitted to the **Journal of Process Control**. Expected submission time is October 2010.

Citation:

M. Golshan, J.F. MacGregor, M.J. Bruwer, and P. Mhaskar, "Latent Variable MPC for trajectory tracking in batch processes: role of model structure," *Proceedings of American Control Conference*, St. Louis, USA: July 2009, pp. 4779-4784.

M. Golshan and J.F. MacGregor, "Latent Variable Modeling of Batch Processes for Trajectory Tracking Control," *Proceedings of 9th international Symposium on Dynamics and Control of Process Systems*, Leuven, Belgium: July 2010, pp. 13-18.

3.1. Introduction

The trajectory tracking control problem in batch processes involves the control over the local batch behavior at every sample time throughout the duration of the batch to make key process variables follow their corresponding set-point trajectories. One of the main bottlenecks in the application of advanced control algorithms for trajectory tracking in batch systems is the lack of a reliable process model. When available, nonlinear mechanistic models have been used for the trajectory tracking control of batch processes [1],[2]. However, due to the difficulties associated with the development of reliable mechanistic models for real batch processes, empirical models are often more appealing for practical situations.

Whether using mechanistic or empirical models, a candidate for trajectory tracking in batch processes is Model Predictive Control (MPC) [3-5]. MPC is one of the most appreciated control methodologies in the industry. Flores-Cerrillo and MacGregor [6] proposed the idea of Model Predictive Control over batch trajectories based on Latent Variable models. In this study an alternative approach for control and modeling methodologies for trajectory tracking control using MPC based on latent variable models is proposed [7]. The approaches mentioned above are based on using a Principal Component Analysis (PCA) model in the core of the prediction and control algorithms. Missing data imputation methods [8],[9] are used for model predictions in the batch LV-MPC.

In the current chapter different latent variable modeling alternatives for use with the LV-MPC methodology [7] are investigated and their performances on two case

studies are evaluated. The two previously proposed LV modeling approaches [6],[7],[10] are incorporated in the course of the new LV-MPC methodology, one based on the batch-wise unfolding (explained in chapter 2) and the other based on the observation-wise unfolding of the data array [6]. The advantages and disadvantages of these two modeling approaches are discussed. Then, a new modeling approach is proposed with the objective of avoiding the major problems of each of these modeling approaches while retaining their important benefits.

The structure of this chapter is as follows: Section 3.2 reviews the two LV modeling approaches presented in [10] and a new modeling alternative is proposed based on the benefits and drawbacks of each of them. A brief review of multiphase modeling for the different modeling alternatives and the identification conditions used to generate the training dataset for this study is also addressed in section 3.2. In Section 3.3 the LV-MPC control methodologies proposed in chapter 2 [7] are referenced to maintain continuity. Section 3.4 contains the studies on implementation of the LV-MPC using three different modeling approaches on two batch case studies (a Single Input Single Output (SISO) process, and a Multi Input Multi Output (MIMO) process). The conclusion and recommendations are presented in Section 3.5.

3.2. Latent Variable Modeling of Batch Processes

In the current section, two previously proposed alternatives for LV modeling of batch processes from the identification (training) dataset is reviewed and compared.

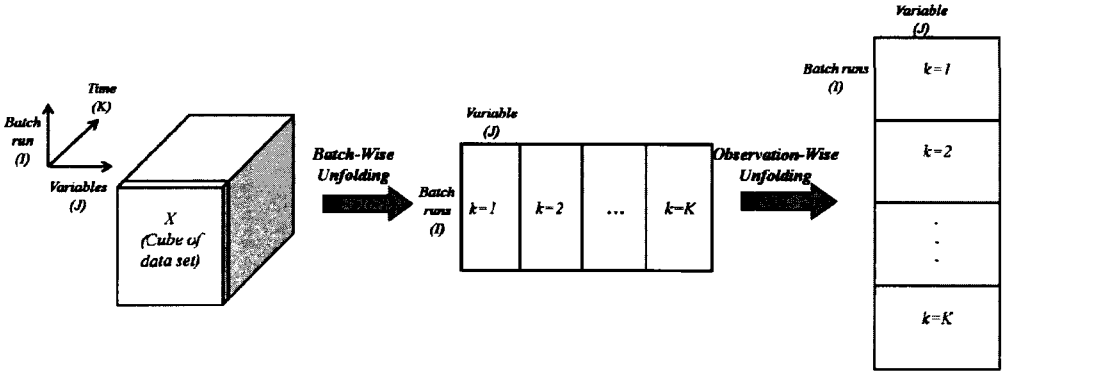
Then, the new LV alternative proposed in this study is presented. The multiphase modeling concept presented in chapter 2 is applicable to all modeling alternatives.

3.2.1. Rearrangement of batch dataset

The structure of the data collected from a batch process is a cube as shown in Fig.3.1a where for "*I* " batch runs, the trajectories of "*J*" variables are measured over "*K*" time intervals. Latent variable modeling of these data involves unfolding this data array into a two dimensional matrix. The main difference among the approaches [11-13] stems from the way they construct the 2-dimensional array (a matrix) from the cube of the data array. Each method of unfolding the data array and performing PCA involves decomposing a different covariance matrix.

Batch-Wise Unfolding (BWU)

Nomikos and McGregor [11],[14] suggested many unfolding possibilities, but proposed the batch-wise unfolding approach (shown in Fig.3.1b) as the most logical way for modeling the differences among batches. In this approach all the variables at different sample times are put beside each other and each batch history constitutes one observation or row in the unfolded matrix (Fig. 3.1b). In this way the rows correspond to individual batches and the latent variable scores summarize the differences among the batches. Applying PCA or PLS to this unfolded matrix allows for modeling the time-varying behavior of the batches as a locally linear model at every sample time. The application of the batch-wise unfolding in the course of LV-MPC is investigated in chapter 2.



(a) $X_{cube(I\times J\times N)}$ (b) Batch-wise Unfolding (c) Observation-Wise Unfolding

Fig. 3.1: batch process dataset, batch-wise unfolding, and observation-wise unfolding

Observation-Wise unfolding (OWU)

Nomikos [15], Nomikos and MacGregor [16] and Wold et al. [17] proposed the observation-wise unfolding approach in which observations of the variables at each sample time are the rows of the matrix. Hence, the latent variables model differences among observations over time. A schematic diagram is shown in Fig. 3.1c. The underlying assumption behind this approach is the fact that the correlation structure among the dataset does not vary with time and a static average model is enough to explain the process. Hence, an LV model on the OWU matrix can be built using as few as 1-3 batch runs by considering each time step during a batch as an observation. However, the nonlinear and time varying behavior which is typical for most batch processes is ignored by this approach.

Observation-Wise with Time-lag Unfolding (OWTU)

To overcome the lack of dynamic modeling ability of the OWU approach, a modification inspired by finite time series modeling has been proposed [6],[18] to include

time lags in the observation-wise unfolded batch dataset. This approach is similar to using an ARX model at all time periods during the batch or each phase of the batch. The resulting model is a dynamic model but still an average model over the whole batch or over each phase. A schematic of the OWTU is illustrated in Fig. 3.2. In this approach the current time is considered to be the middle column and *ph* and *fh* are respectively the past and future number of sample times considered about each time point.

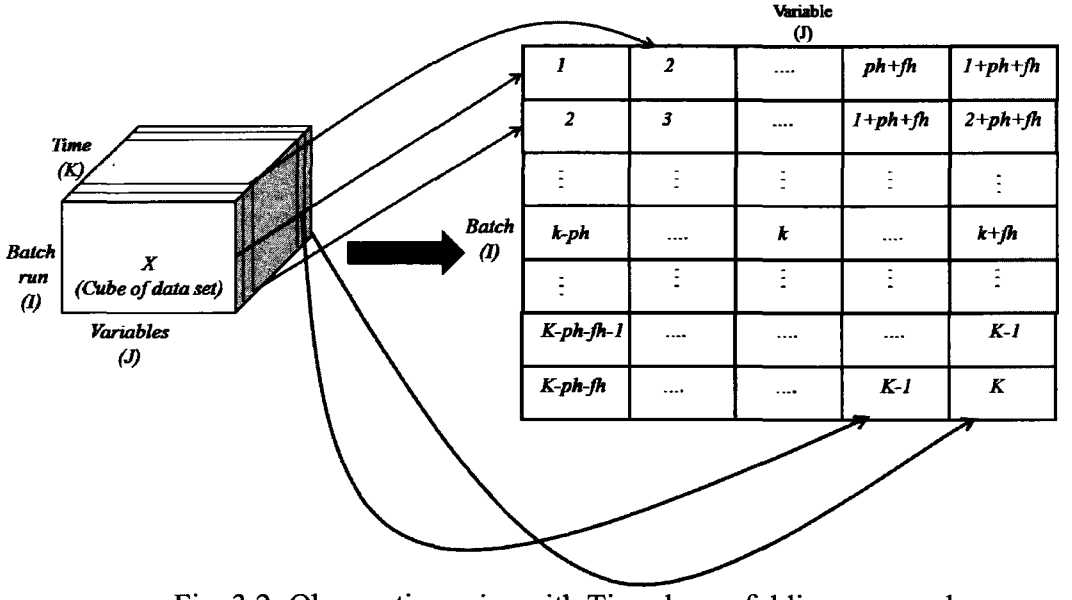


Fig. 3.2: Observation-wise with Time-lag unfolding approach

3.2.2 Comparison of BWU and OWTU approaches

Batch-wise unfolding puts all the variables at all time lags into one row and then mean centers to remove the average trajectories. A PCA or PLS model then provides different loadings or weights for every deviation variable at every time point throughout the duration of the batch. As a result, the latent variable model captures the time-varying

properties throughout the batch as a locally linear model at every time point. Therefore, this BWU approach offers the considerable advantage of capturing the time varying characteristics that can be useful for batch process control. However, at the high sampling frequencies required for trajectory control, the data requirements to identify the LVMs based on the BWU data present a problem. First, BWU becomes a short, fat matrix and one often needs many Principal Components (PCs) to model the BWU matrix which means many batch runs are needed in the identification dataset. Second, the number of loading parameters required to capture the locally varying dynamic effects is large and the number of multivariate observations (batches or rows) available for the model building is usually not large. Therefore, the variance of the resulting loadings is large. This causes the PCA model to be non-smooth (see Fig. 3.4 below). Although the use of many observations (batches) smoothens the loading matrix and minimizes the aforementioned problem, with minimum number of observations required for modeling this problem exists. Thus, the batch-wise unfolding approach needs a large number of batches to build a PCA model. This requirement is the most critical bottleneck in modeling batch processes using batch-wise unfolding approach for trajectory tracking control. (Note that this is usually not a problem for the traditional batch data analysis, monitoring and end-point control problems where the sampling rates and control frequencies are much lower).

On the other hand, in the observation-wise with time-lag unfolding (OWTU) approach one gets a huge number of observations (rows) using even 1 batch and only has to identify an average model for the whole batch or for each batch phase. As shown in

Fig. 3.2, the total number of observations resulting from each batch is *K-ph-fh*. For example, for a batch or a batch phase with 300 sample times and using a typical value of 20 for each *ph* and *fh*, one batch results in 260 observations that are much more than enough for building a PCA model. The main drawback of this algorithm is that it is an average model for the batch or for each phase and cannot handle time varying and nonlinear behavior.

3.2.3. Regularized Batch-Wise Unfolding (RBWU)

In this section a third modeling approach that is proposed in this study is presented that aims to capture the major benefits of the two previously mentioned modeling approaches, while avoiding the problems related to each one. The new unfolding approach uses elements from both of the preceding approaches. It unfolds batch-wise but also repeats each batch row L times, each time shifted by one additional sampling interval. A schematic of the batch-wise with time-shifting approach is shown in Fig. 3.3. The parameter L is the number of time shifts used which can be thought of in two ways: One can start with BWU shown in Fig. 3.1 and then replicate each row L times while shifting it by one interval in each case. Alternatively one can start with OWTU shown in Fig. 3.2 and use the past and future horizons (*ph*, *fh*) to cover the (K-L) time steps of the batch in each row and use only L block rows.

If L=0 (no shifting) this unfolding is simply BWU. But if a small number of shifts (e.g. L/K<0.05) are used, this approach will retain most of the advantages of the BWU approach (capturing time-varying non-linear behavior), but at each time interval

the model will be averaged over L consecutive time periods thereby restoring some of the advantages of the OWTU. For example, if L is small (e.g. L=5) and the number of time intervals is large (e.g. K>300), this will not seriously affect the capturing of any time-varying behavior, but it will reduce the variance of the latent variable model loadings since these loadings at each time point will now be averaged over L consecutive time periods (see equation 3.4 and Fig.3.4). The time shifting effectively provides a regularized BWU latent variable model where the loading estimates are effectively averaged over a window of L local time periods. The resulting model will therefore have a significantly smoother variation in the loading coefficients with time. A similar regularization of the latent variable model could be achieved by performing PCA or PLS with constraints on the rate of change or smoothness of the loadings from one time interval to the next. However, this would lead to a non-linear modification of the latent variable estimation algorithms with essentially the same result as achieved by using this simple time shifting with the standard algorithms.

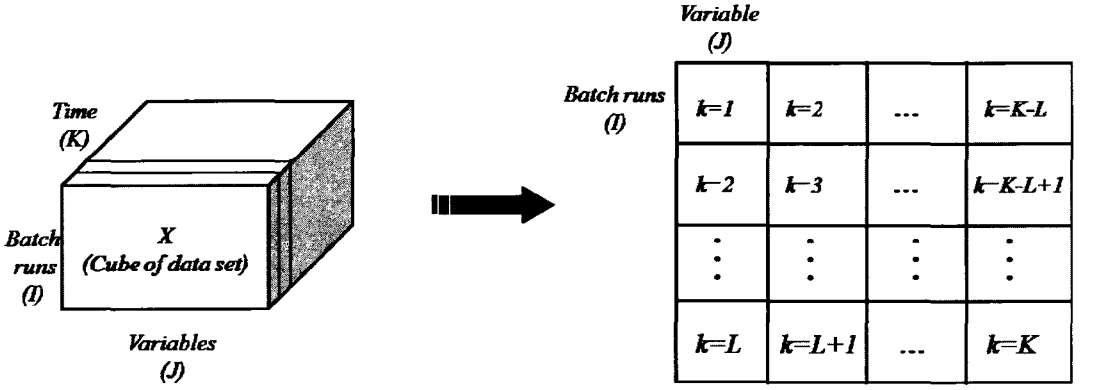


Fig. 3.3: Schematic of the Regularized Batch-Wise Unfolding (RBWU)

To show how the batch-wise with limited time-lag unfolding approach leads to a regularized PCA, the batch-wise unfolded matrix in the Fig.3.1 is defined as:

$$X_1 = [a_1, a_2, ..., a_K] \tag{3.1}$$

where $a_1,..,a_K$ are matrices of J measured variables for all batches ($1:I$) at sample times $1,...K$ (blocks in Figs. 3.1b and 3.3, $a_i \in \mathbb{R}^{I \times J}$). Then, the corresponding batch-wise with time-lag unfolding approach can be shown as:

$$X_2 = \begin{bmatrix} a_1 & a_2 & \cdots & a_{K-L} \\ a_2 & a_3 & \cdots & a_{K-L+1} \\ & \vdots & & \\ a_L & a_{L+1} & \cdots & a_K \end{bmatrix} \tag{3.2}$$

$$Cov(X_2) = \frac{1}{LI} X_2^T X_2 = \frac{1}{LI} \begin{bmatrix} a_1^T & a_2^T & \cdots & a_L^T \\ a_2^T & a_3^T & & a_{L+1}^T \\ \vdots & \vdots & & \vdots \\ a_{K-L}^T & a_{K-L+1}^T & & a_K^T \end{bmatrix} \begin{bmatrix} a_1 & a_2 & \cdots & a_{K-L} \\ a_2 & a_3 & \cdots & a_{K-L+1} \\ & \vdots & & \\ a_L & a_{L+1} & \cdots & a_K \end{bmatrix} \tag{3.3}$$

Using the outer product definition of the matrices:

$$Cov(X_2) = \frac{1}{LI} \left\{ \begin{bmatrix} a_1^T \\ a_2^T \\ \vdots \\ a_{K-L}^T \end{bmatrix} \begin{bmatrix} a_1 & a_2 & \cdots & a_{K-L} \end{bmatrix} + \begin{bmatrix} a_2^T \\ a_3^T \\ \vdots \\ a_{K-L+1}^T \end{bmatrix} \begin{bmatrix} a_2 & a_3 & \cdots & a_{K-L+1} \end{bmatrix} + \cdots + \begin{bmatrix} a_L^T \\ a_{L+1}^T \\ \vdots \\ a_K^T \end{bmatrix} \begin{bmatrix} a_L & a_{L+1} & \cdots & a_K \end{bmatrix} \right\}$$

Hence,

$$Cov(X_2) = \frac{1}{L} \left(Cov(X_{1,1:K-L}) + Cov(X_{1,2:K-L+1}) + ... + Cov(X_{1,L:K}) \right) \tag{3.4}$$

where $X_{1,i:K-j}$ means the blocks "*i*" to "*K-j*" of matrix X_1. The equation (3.4) shows that the covariance of the resulting matrix from the batch-wise with time-shifting approach (X_2) is the average of the original covariance matrix (X_1) over L consecutive sample times. Thus, the covariance of the resulting matrix is a regularized version of the covariance of the original matrix. Note that the loading matrix of the PCA model built for the X_2 matrix is the right singular matrix of the X_2 matrix or the eigen vector of the covariance matrix: $cov(X_2)$. This is a regularized version of the loading vector for the X_1 matrix. Fig. 3.4 shows a typical regularization obtained by performing this approach to a BWU dataset.

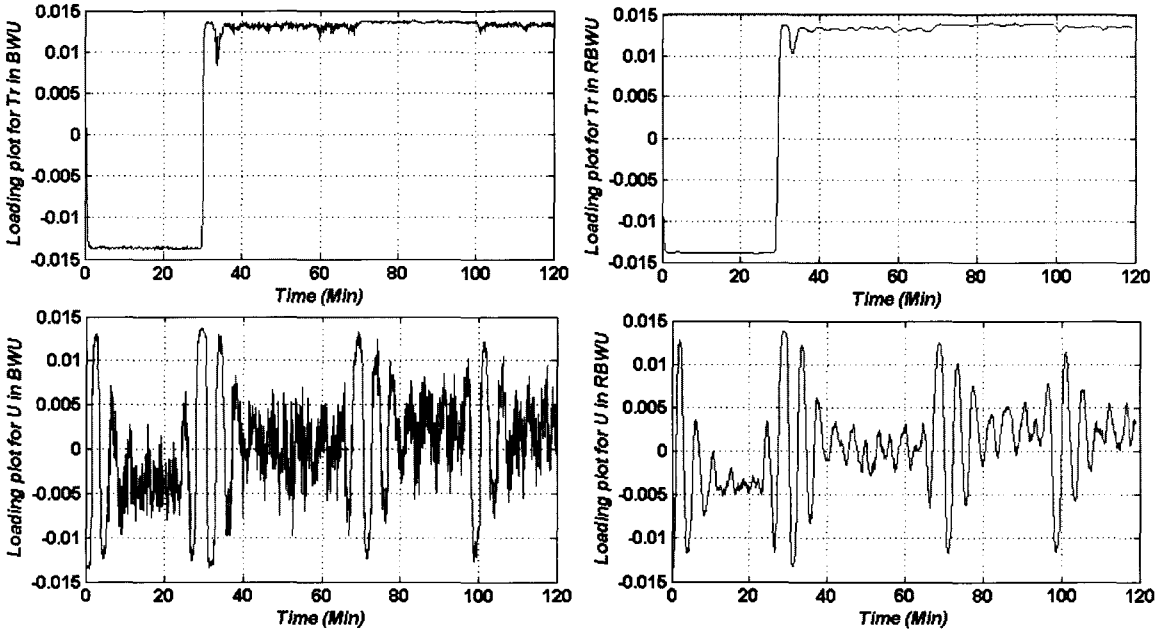


Fig. 3.4: The left figures show the loading plots corresponding to the first principal component obtained from a batch-wise unfolded dataset for the controlled (T_r) and manipulated (U) variables with 20 batches. Right figures show the same plot but for a RBWU data derived from the same BWU dataset by considering L=5. The first case study presented in section 3.4.1 is used to generate this figure.

The main benefit of using the observation-wise with time-lag unfolding (OWTU) approach is that the batch can be modeled using only 1-3 batch runs (although one only obtains an average dynamic model for the batch). However, with this regularized approach one gets some of the same benefit without the liability. For example, by using 10 batches and L= 5-10, one can get 50-100 observations (rows) in the unfolded matrix which are adequate for implementing Multiphase LV-MPC (MLV-MPC) as proposed in [7]. In general the choice of the number of time shifts (L , the regularization parameter) will depend upon the rapidity of the time varying behavior of the batch and the number of batches available for the model identification.

3.2.4. Multi-phase Modeling

LV modeling of batch processes for the trajectory tracking control using the BWU approach leads to a very large global LV model because of the large number of time intervals over the batch duration. This is not desirable as explained in chapter 2 because it requires many latent variables (which implies many batches may be required in the training set), leads to ill-conditioned matrices in the model used during the control computations and does not focus enough on the local behavior of the trajectories. Therefore, utilization of multi-phase LV (MLV) models, as presented in chapter 2 is necessary or at least preferred. The multiphase modeling approach in the BWU modeling approach is based on identifying multiple phases within the batch according to the discussions presented in section 2.3.1, partitioning of the dataset according to the phases, considering overlap between adjacent phases, and building PCA models for each phase.

The same multiphase modeling approach is applicable to a regularized batch-wise unfolded dataset (RBWU dataset) and the same problems exist if only one phase is used throughout the batch. The resulting matrix from a batch-wise unfolded dataset with time-shifting is considered as a new regularized batch-wise unfolded matrix and the multiphase modeling is performed in the same manner as described in [7].

However, for the OWTU approach, multiphase modeling is more critical. The observation-wise with time-lag unfolding approach by its definition is applicable to the processes that are not highly nonlinear or time-varying and whose behavior can be modeled by an average linear dynamic model. When a batch process is time-varying, it can be broken up into multiple phases and a time-invariant OWTU PCA model can be built for each phase. The phases should be selected such that within each phase there is minimum variation in the process characteristics (i.e., the correlation structure among the time dependent variables is as constant as possible). Thus, poor phase selection will result in loss of model accuracy. Furthermore, multi-phase approach in OWTU modeling results in switching between different local models at the transition between phases. This switching manner may result in performance deterioration for a transient period because of inconsistencies between the two adjacent models. Thus, selection of multiple phases in OWTU approach carries both desirable and undesirable effects and may not always improve the performance of the LV-MPC.

To implement the MLV-MPC on a batch-wise unfolded dataset or an observation-wise with time-lag unfolded dataset, one needs to run a conventional PI controller or the LV-MPC based on a moving window PCA model discussed in chapter 2 for the first few

sample times at the beginning (e.g. *ph* sample times) and end (e.g. *fh* sample times) of the batch where there is not enough past or future horizon to initiate the missing data imputation and MPC algorithm respectively (see section 2.3.2). However, using the RBWU approach, the size of the blocks in the batch dataset decreases to *K-L* which means there are dataset for *K-L* time steps. Thus, one should add another *L* sample times to the period of applying the PI controller at the beginning or the end transition time and increase either of those sample times to *(ph +L)* and *(fh +L)* respectively. This is the only penalty one has to accept to use the new unfolding approach which is negligible.

3.2.5. Identification Experiments

The direct identification approach based on closed-loop data is used in this study. Closed-loop identification is preferred over open-loop identification for batch processes in order to maintain the process close to its desired trajectories and to minimize the final product quality variations. The closed-loop design of identification experiments for identifying models of time varying, finite duration batch systems is discussed in the next chapter. Therefore, we do not discuss the identification issues in this chapter except to note a few issues and some observations from the simulation studies presented below. In this study it was found that models were identifiable from historical batches run under pure feedback control with inclusion of a few batches run using different set-point trajectories in the dataset. In other words, there is no need for additional excitation of the closed loop system by a Random Binary Sequence (RBS) signal on top of the controller output. This somewhat surprising result can be explained by analogy with the closed-loop identification of linear time-invariant systems [19],[20]. There it has been shown that the

closed-loop identifiability is satisfied if one switches between a sufficient number of linear controllers or if the control is nonlinear. Here the batch system is non-linear and time varying and so with a fixed PID controller it is analogous to the controller being time varying or nonlinear for a linear system. Furthermore, there are time-varying set-points in the training data generation that also helps to satisfy the identifiability conditions. The historical batch data are also very important for providing models for the effects of inherent disturbances in the batch process and their influence on the behavior of the evolving trajectories. This information is essential for the prediction of the future trajectories and ensuring no steady-state offset in the control as discussed in the next section.

3.3. Control Methodology

Two control formulations for Multiphase LV-MPC (MLV-MPC) methodology were proposed and studied in chapter 2 [7]. The difference is that one is formulated in the latent variable space while the other is formulated in the manipulated variable space. The MLV-MPC formulation is used in this chapter to evaluate the performance of different LV modeling alternatives in the course of batch trajectory tracking control. In this chapter we mainly focus on using the formulation in the latent variable space. Details of the LV-MPC methodology is presented in section 2.3.2.

3.4. Simulation Studies

This section investigates the LV modeling alternatives and evaluates them via the performance of the MLV-MPC controller obtained using each alternative modeling approach. In section 3.4.1 the same Single Input Single Output (SISO) example as used in the previous chapter [7] is selected and the proposed LV-MPC is tested on the modeling approaches using BWU, OWTU and RBWU data matrices. In section 3.4.2 the LV-MPC is tested on a Multi-Input Multi-Output (MIMO) batch reactor example based on the three aforementioned modeling approaches (BWU, OWTU, RBWU).

The main objective in the following tests is set-point tracking. However, it is of great importance to check the power of disturbance rejection for the proposed control methodologies. It is shown that if appropriate nonstationary disturbance models are used in designing the controller, then offset free tracking should be achieved [21-23]. This integral action comes from the inclusion of nonstationary disturbance states into the model thereby giving rise to the ability of the state estimator to predict future offsets resulting from the appearance of these disturbances. It is observed in this study that if nonstationary effects are present, then some of the latent variables will be nonstationary in nature. As a result the future predictions of the trajectories using the missing data imputation algorithms (the equivalent of state estimators in this methodology) will appropriately capture any drifts due to the disturbances that have entered the current batch. In the set-point tracking studies shown in the figures of this section no offset is apparent. The PCA model developed from closed-loop data effectively models the persistent offsets remaining from the PI controller as nonstationary states (latent

variables) and these drifts were then properly predicted over the future horizon in any new batch and eliminated by the MPC.

If the dataset includes some measured variables (e.g. x_{me}) as well as the controlled variables (y) in them, the effects (including nonstationary effects) of all disturbances showing up in any of these measurements are expected to be modeled more explicitly. However, in this study it is shown that this is not a limitation and the LV-MPC is capable of tracking the set point and rejecting the load even with the minimum number of measurements. This is illustrated by considering extra measurements in the first case study, but minimum number of measurements (only y, u, y_{sp}) in the second case study.

Most of the disturbances in batch chemical processes occur in the initial conditions and these will evolve through the process with a slow and predictable dynamics. However, to provide a more severe test of the disturbance rejection ability of the batch LV-MPC, a very large additional random walk disturbance [24],[25] is superimposed upon the Controlled Variable (the CV) for several simulation runs as well as different initial conditions for the batch and the ability of the LV-MPC to eliminate the large offsets coming from these disturbances is investigated. This disturbance rejection test may not represent reality in many chemical processes, but it does provide a good test of the disturbance rejection ability of the LV-MPCs in the simulations.

The three modeling approaches presented in section 3.2 (BWU, OWTU, RBWU) are studied on two case studies for set-point trajectory tracking as well as disturbance rejection for each modeling approach under the two control formulations presented in chapter 2. However, it should be noted that the two control formulations are thoroughly

compared in chapter 2 and shown to be similar. Furthermore, the objective of the current chapter is not to compare the performance of the different control methodologies. Thus, for the sake of conciseness in the current chapter the results are only presented for one control methodology (Control in latent variable space). However, as the complementary work to chapter 2, the study on the BWU for the second case study (section 3.4.2) is presented for both control methodologies. The reader can interpret the results in the line of discussions presented in chapter 2.

3.4.1. SISO Process

A batch reactor model is used by Aziz et al. [26] as a case study for temperature control problem. This case study was originally proposed by Cott and Macchieto [27]. The schematic of the reactor is shown in Fig. 2.5. This process is explained in chapter 2 (section 2.4). The detailed model for this process is presented in [26] and appendix C.

Batch-Wise Unfolding (BWU)

The performance of MLV-MPC using control formulation in the latent variable space on batch-wise unfolding approach is shown Figs. 3.5-3.6. In these figures the LV-MPC is performed in the multiphase framework (MLV-MPC) based on the 6 phases along the batch. These phases are constructed on the intervals of: 1-30, 30-50, 50-60, 60-80, 80-100, and 100-120. For this control study data from 45 historical batches run under closed-loop control with a PI controller (40 batches with similar set-point trajectories plus 5 batches with slightly different set-points trajectories) were used to develop the PCA

models (see section 4.2). The control is executed in the mean-centered and auto-scaled space, and y is mean-centered by y_{sp} as discussed in chapter 2.

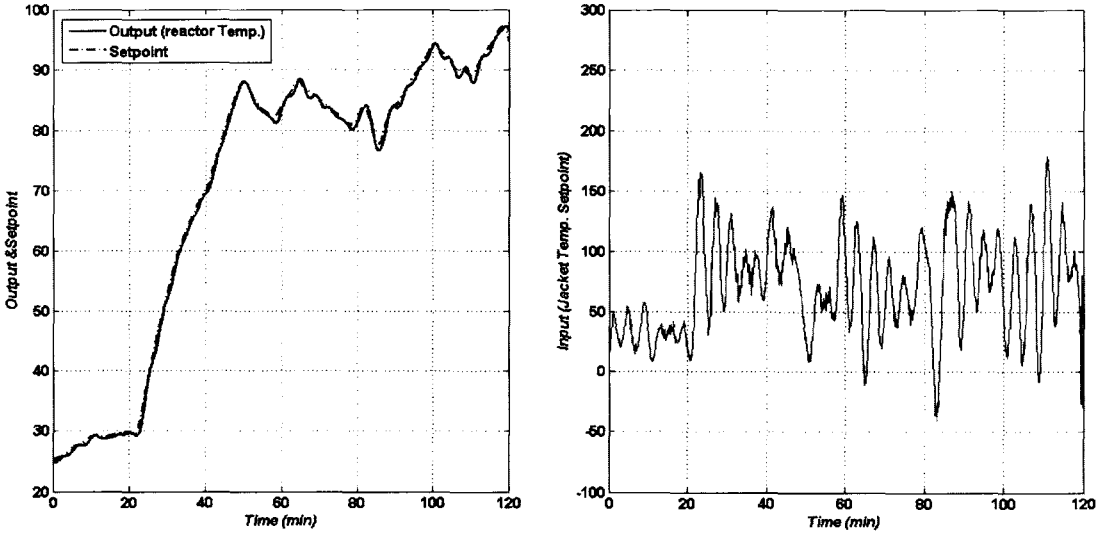


Fig. 3.5: MLV-MPC for set-point tracking using PCA model on batch-wise unfolded dataset

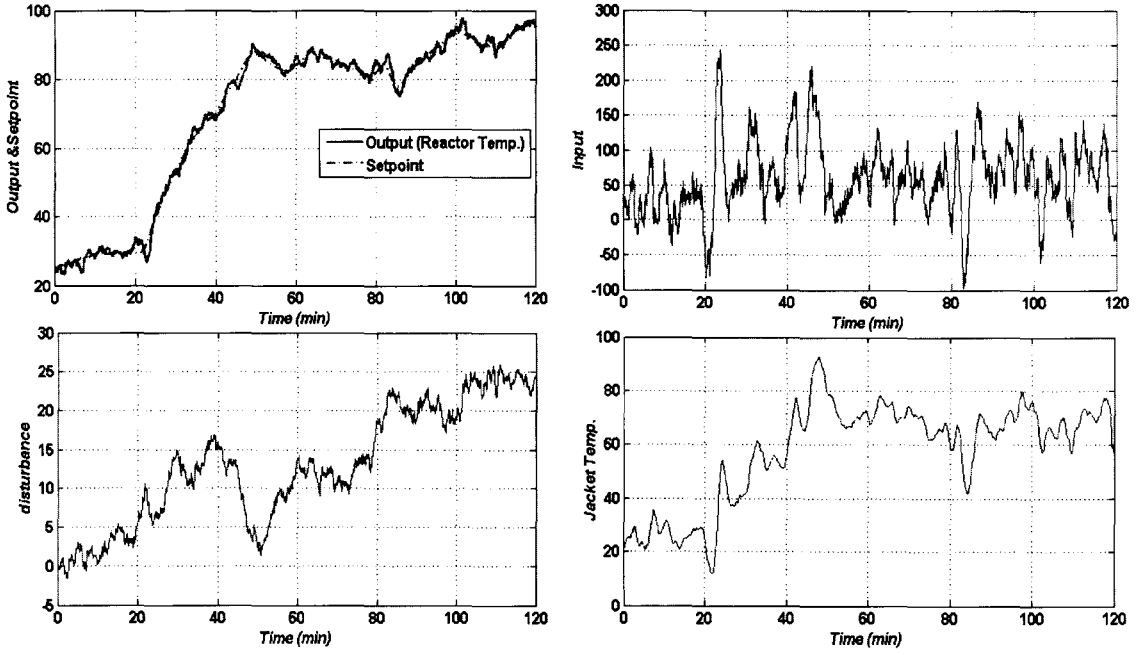


Fig. 3.6: Random Walk disturbance rejection of MLV-MPC using PCA model on batch-wise unfolded dataset

It is seen that the MLV-MPC methodology based on batch-wise unfolding is capable of tracking the set point trajectory as well as rejecting a significant amount of a nonstationary disturbance. The above figures are comprehensively discussed in chapter 2 [7].

Observation-Wise with Time-lag Unfolding (OWTU)

The next modeling candidate that is investigated is the Observation-wise with Time-lag unfolding approach. It should be noted that the OWTU approach, has good performance as long as correlation structure among the process variables do not change considerably over the time of the phase. In this case an average model is enough to represent the batch phase adequately. Figs. 3.7 and 3.8 show the results of the LV-MPC on observation-wise with time-lag unfolding approach. The same phases as are presented in BWU approach are applicable to the following figures. Only two historical batch runs executed on one type of set-point trajectory are used to model the process using OWTU approach.

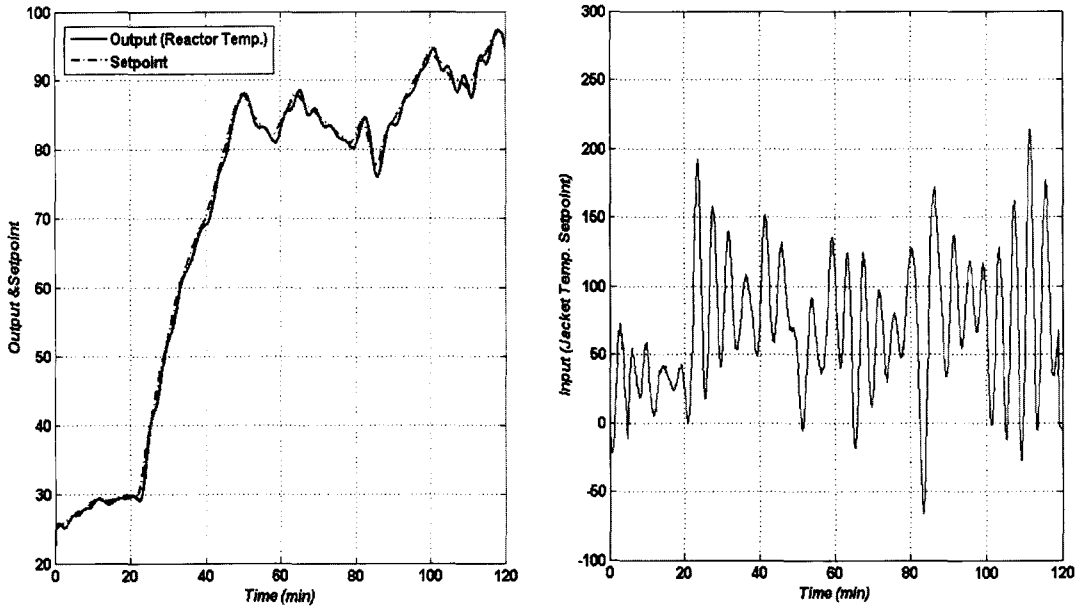


Fig. 3.7: MLV-MPC for set-point tracking using PCA model based on the OWTU dataset

Fig. 3.7 shows that the LV-MPC algorithm has a good tracking performance. However, its performance is slightly inferior to that of the batch-wise unfolding approach (see Table 3.1). However, this modeling approach results in a smoother manipulated variable behavior. This property was predictable according to the discussions in section 3.2. One loses the model accuracy by using an average model but gains smoothness in the manipulated variable. Fig. 3.8 shows that the LV-MPC algorithm based on OWTU rejects the random walk disturbance leading to an offset-free trajectory tracking. The input smoothing using OWTU approach becomes more important in this situation. Fig. 3.6 shows that in the batch-wise unfolding approach one can reject the disturbance, but the resulting aggressive input may not be possible in a realistic condition. In the observation-wise with time-lag unfolding approach, the disturbance is rejected using a

more realistic input. This property along with the fact that one can build this model using only 1-3 batches makes this approach very practically attractive.

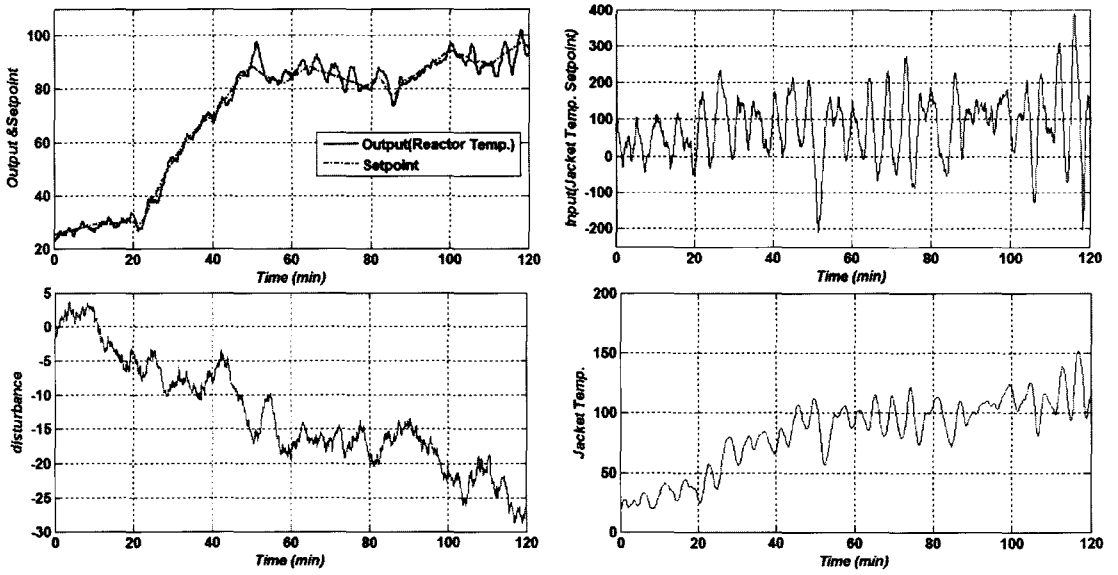


Fig. 3.8: Random Walk disturbance rejection of MLV-MPC using PCA model on OWTU dataset

Regularized Batch-Wise Unfolding (RBWU)

Since the batch takes 2 hours, using sample time of 0.1 min the number of control intervals comes up to 1200 sample times. However, in this batch process the correlation structure does not change very fast. Thus, Instead of using 45 batch runs as is used in BWU, it is possible to use 20 historical batches (17 batches with similar set-point trajectories plus 3 batches with slightly different set-point trajectories) and repeat those batches 5 times (L=5) according to the RBWU algorithm presented in section 3.2. Thus, one gets 100 observations to model the batch (any phase of the batch). Figs. 3.9 and 3.10

show the performance of MLV-MPC based on the RBWU modeling approach using this reduced number of batches. Both set point tracking and disturbance rejection are good.

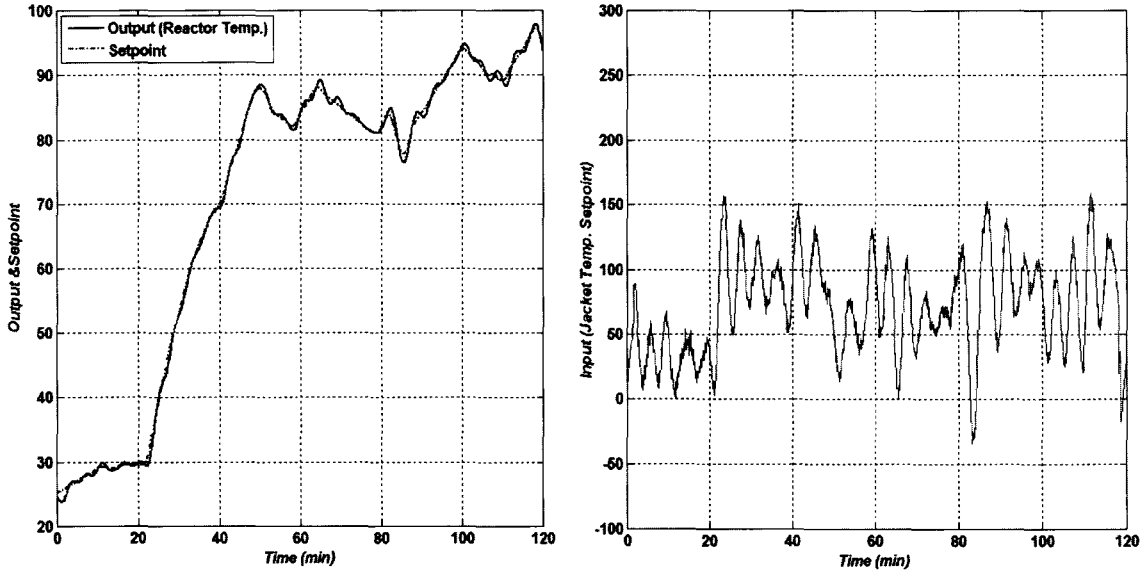


Fig. 3.9: MLV-MPC for set-point tracking using PCA model on RBWU dataset

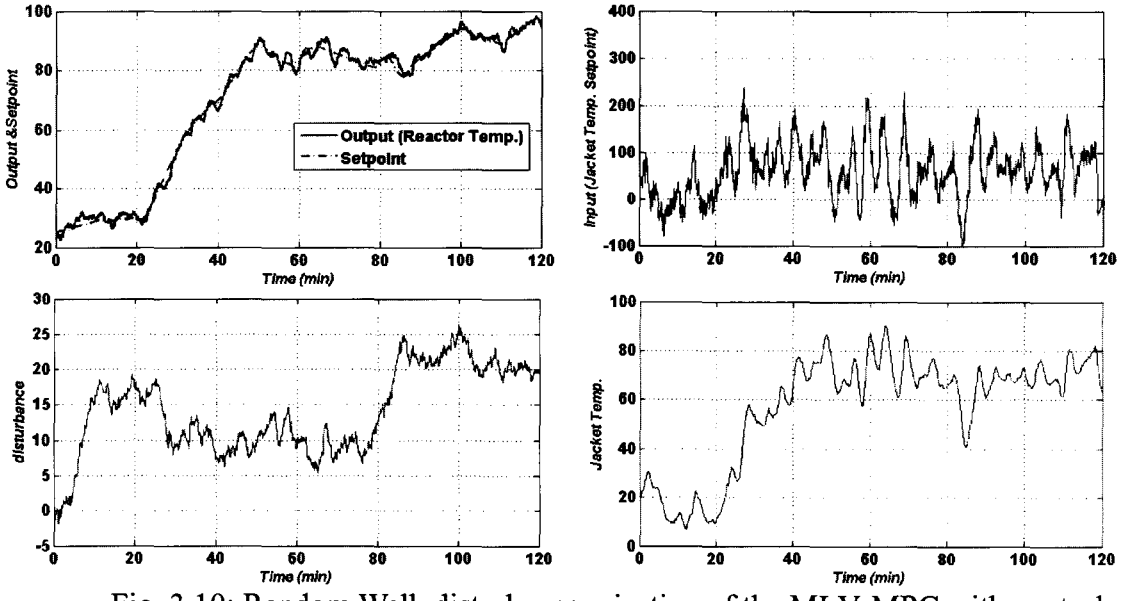


Fig. 3.10: Random Walk disturbance rejection of the MLV-MPC with control formulation in LV space using PCA model on RBWU dataset

Table 3.1 summarizes the results of implementing the LV-MPC algorithm on the SISO temperature control problem comparing performance indices obtained by testing different modeling approaches.

Table 3.1- Comparison of the effect of different modeling alternatives for the Temperature set-point tracking control by LC-MPC

Performance criteria	BWU approach	OWTU approach	RBWU approach
RMSE of "y-ysp"	0.6910	0.9938	0.6838
STD of "Δu"	6.7085	6.4766	6.5809

where *RMSE* is the acronym for "Root Mean Square Error" and *STD* is the short for "STandard Deviation". Table 3.1 shows that the RBWU approach (last column) slightly outperforms the BWU approach and at the same time, leads to a slightly smoother manipulated variable. However, the major advantage of the RBWU over BWU is that only 20 batches are used to build the PCA model on RBWU dataset while the BWU approach uses 45 batches in the training dataset. The OWTU leads to a smoother control action but results in a higher value for the RMSE index.

Table 3.1 shows that the standard deviation of change of the manipulated variable in OWTU is slightly less than the other two modeling approaches. Nonetheless, looking at Figs. 3.5, 3.7, 3.9 shows that the manipulated variable in the LV-MPC based on OWTU approach is considerably smoother than that of the BWU and RBWU approaches. But, there are wider oscillations of input variable in the OWTU approach compared to other two modeling approaches and this contributes to the "*STD of Δu*" index and increases it. This wider range of change of manipulated variable in OWTU approach

comes from the fact that the model is inferior to the models built on BWU and RBWU dataset. Thus, more changes in the control actions (feedback actions) are needed to compensate for the modeling errors.

3.4.2. MIMO Process

The example considered in this study is a Nylon 6, 6 polymerization process. The case study is introduced by Russell et al. [28]. The polymerization reaction happens in an autoclave. A reversible reaction occurs between Amine end groups (A) on Hexa-methylene Diamine (HMD) monomer or polymer chain and Carboxylic end groups (C) on adipic acid monomer or polymer chain ends with the condensation reaction shown in equation 3.7. As a result the polymer chain links (L) and water (W) are produced. Besides, carboxylic end groups and polymer chains may decompose and stabilized end groups (SE) are formed according to equation 3.5 and 3.6.

$$C \rightarrow SE + W \tag{3.5}$$

$$L \rightarrow SE + A \tag{3.6}$$

$$A + C \rightleftharpoons L + W \tag{3.7}$$

This process is modeled and investigated for the end product quality by Russell et al. [28]. They proposed different control configurations among which the control of reactor temperature and pressure configuration is concluded among the best possible control configurations. The schematic of the reactor is shown in Fig. 3.11.

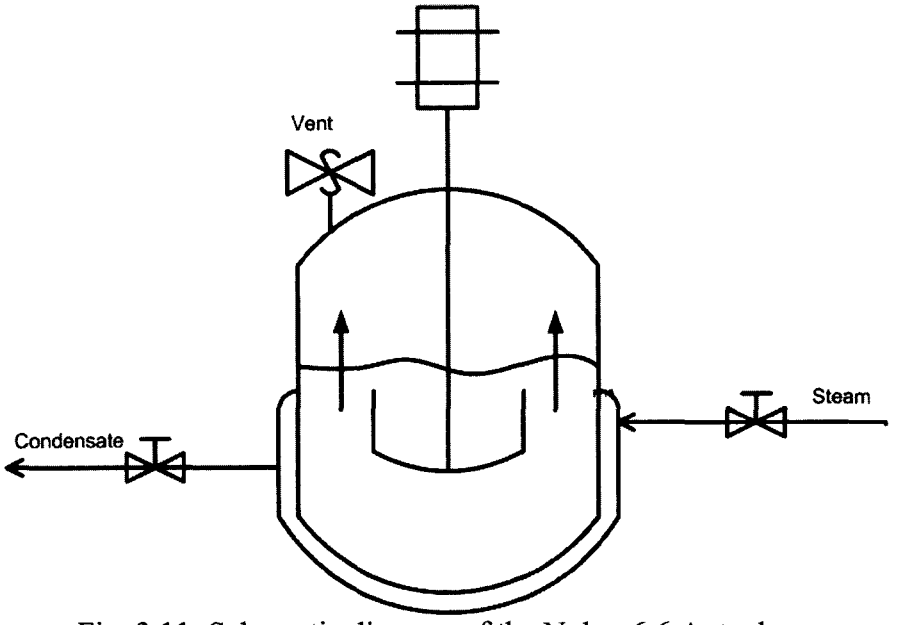


Fig. 3.11: Schematic diagram of the Nylon 6,6 Autoclave

This is a MIMO system where the reactor temperature is controlled using the pressure of steam flow in the jacket and the reactor pressure is controlled using the vent rate through the valve on top of the reactor. It is a constrained problem where vent rate cannot be less than zero and the steam pressure cannot be less than 4 psi or more than 52 psi. Once again, the trajectory selected in this study may not be consistent with the objectives presented in [28]. The main objective of this study is to test the LV-MPC for trajectory tracking problem in a MIMO case study.

This case study is a difficult control problem for several reasons. There is a considerable amount of lack of information in the Nylon process as compared to the first temperature control problem. There is minimum number of measurements (only manipulated variables and controlled variables are measured, and no additional variables that might help with disturbance estimation are available). Furthermore, in a considerable

portion of batch time the manipulated variable is saturated. Moreover, there is a strong interaction between the two control loops. The manipulated variable for the temperature control problem, the steam pressure, has a direct effect on the rate of liquid vaporization which directly affects the reactor pressure. On the other hand, change of reactor pressure by manipulating the vent rate, affects reactor temperature according to thermodynamic principles.

Batch-Wise Unfolding (BWU)

Figs. 3.12 and 3.13 show the application of LVMPC for set point tracking of the nylon 6,6 process using batch-wise unfolding approach. 35 historical batches (30 batches with similar set-point trajectories plus 5 batches with a slightly different set-point trajectories) divided up into 10 phases of equal size were used for identification. There exists a lack of tracking performance at the beginning of the batch. This originates from the use of different initial conditions and the absence of significant reaction in this early period. Furthermore, there is an offset at the end of the batch for the pressure control loop which is due to the lack of volatile materials in the batch reactor at the end of the batch to increase the pressure and although the control is working properly (vent rate is zero) at those time steps, the process is not able to increase the pressure.

As mentioned before, the performance of the LV-MPC formulation in the original variable space of the manipulated variables is also presented here, but only for the BWU approach to allow for a performance analysis of this control formulation in a MIMO system as well.

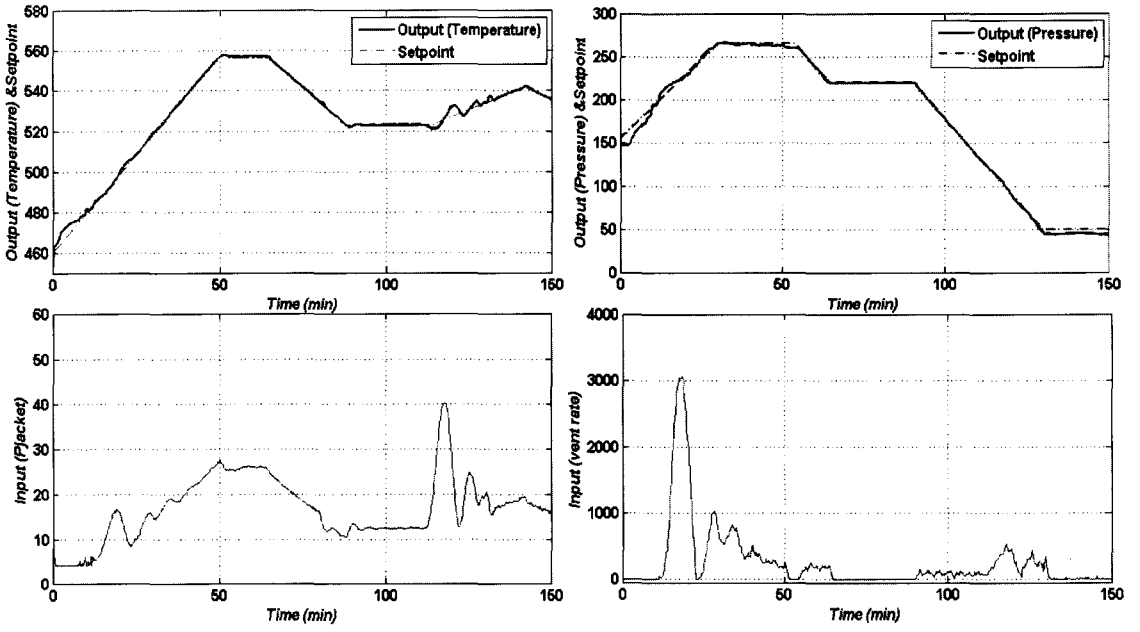


Fig. 3.12: MLV-MPC based on batch-wise unfolding modeling approach on the nylon 6,6 autoclave using control formulation in LV space

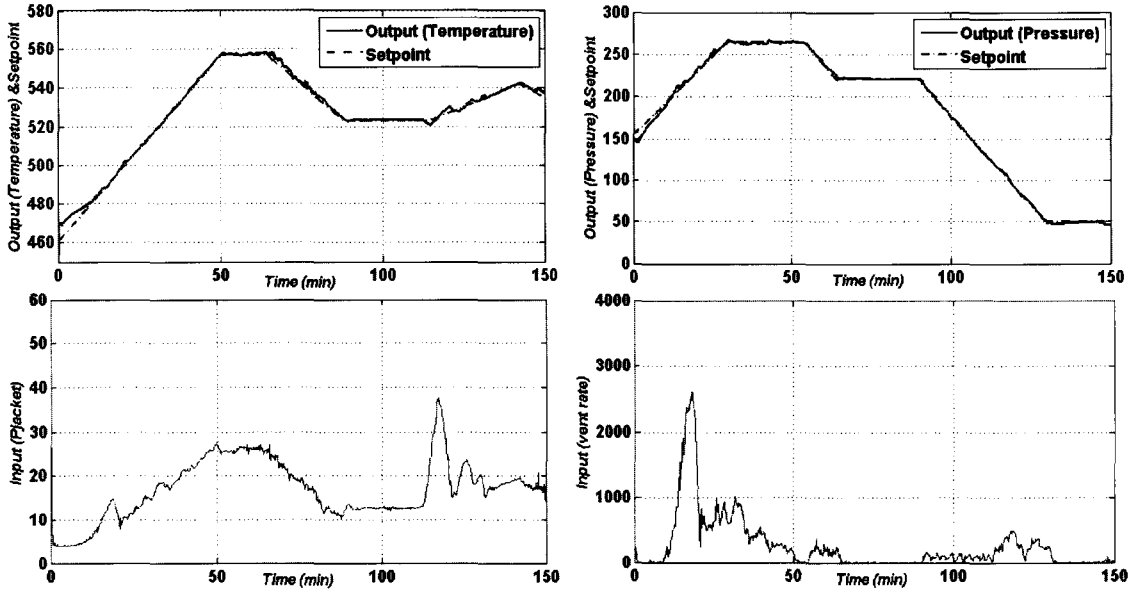


Fig. 3.13: MLV-MPC based on batch-wise unfolding modeling approach on the nylon 6,6 autoclave using control formulation in the original variable space. *RMSE* of T/P= 1.4150/ 2.2523 , "*STD of* Δu" for T/P loop= 0.5876/ 65.4617

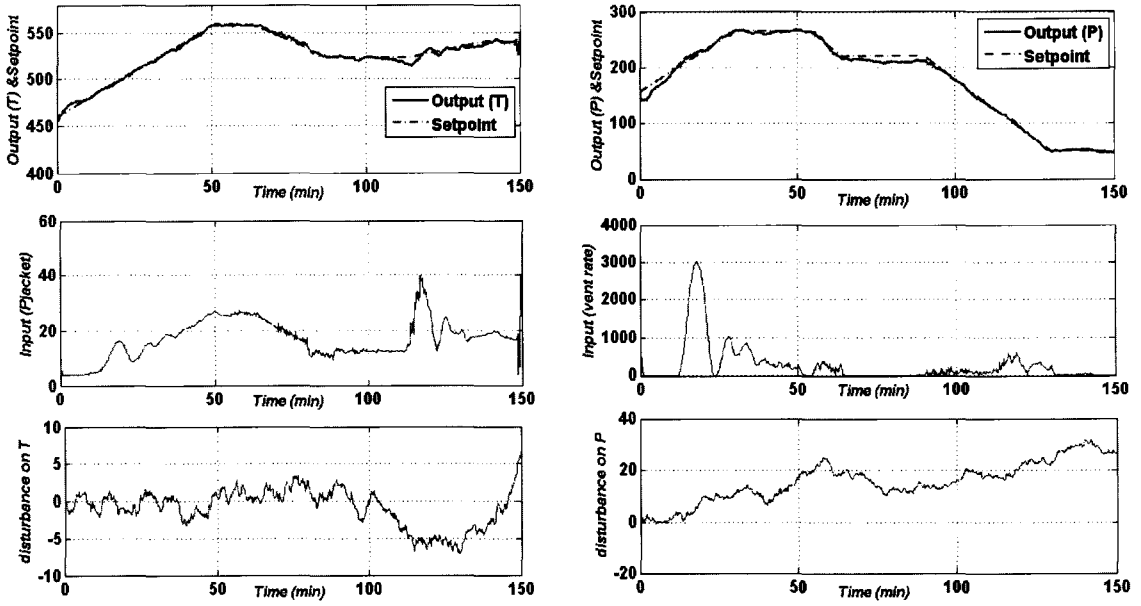


Fig. 3.14: Random Walk disturbance rejection by MLV-MPC based on batch-wise unfolding modeling approach on the nylon 6,6 autoclave using control formulation in LV space

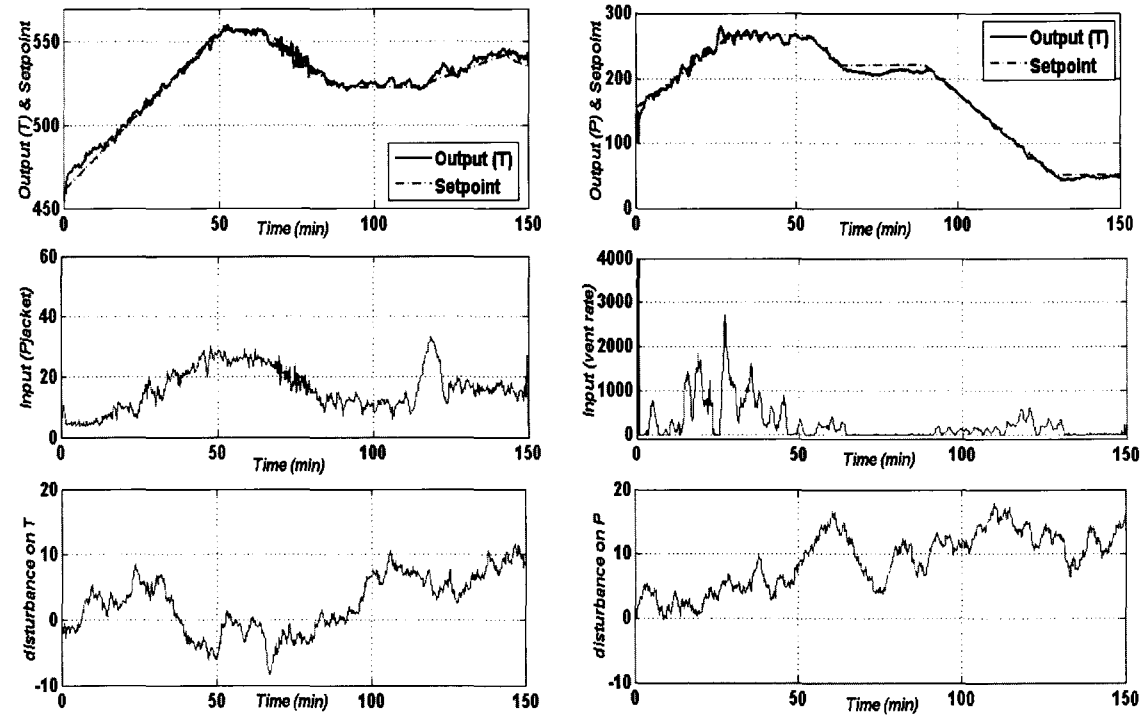


Fig. 3.15: Random Walk disturbance rejection of MLV-MPC based on batch-wise unfolding modeling approach on the nylon 6,6 autoclave using control formulation in original variable space

Figs. 3.14 and 3.15 show the disturbance rejection test. Once again the Random walk disturbance on top of the output variables is tested. It is seen that the algorithm is able to reject the nonstationary disturbance.

Observation-Wise with Time-lag Unfolding (OWTU)

Inadequate number of batches may lead the practitioner to use the observation-wise with time-lag unfolding approach. The performance of the observation-wise with time-lag unfolding approach using one phase throughout the batch and 2 batch histories in the training dataset is shown in Fig.3.16. The tracking quality is a bit worse than that in Fig. 3.12 (see Table 3.2). More oscillations in tracking may result because of the average time invariant model for the whole batch that this method leads to. The case study is a time varying process, and one phase may not be sufficient to model this process. Though, as explained before, the inconsistencies caused by switching between different models on this specific process decrease the quality of the LV-MPC performance. After running a few batches, one may switch to BWU or regularized BWU. However, this modeling approach has the least data requirements for modeling. Moreover, it requires only the data of the current PI controller that is running the process without any additional excitation by dither signal (only historical batches). Fig. 3.17 shows the results of imposing a random walk disturbance on the controlled variable. It is seen that the LV-MPC based on OWTU is able to perform and offset-free tracking in the existence of a nonstationary disturbance.

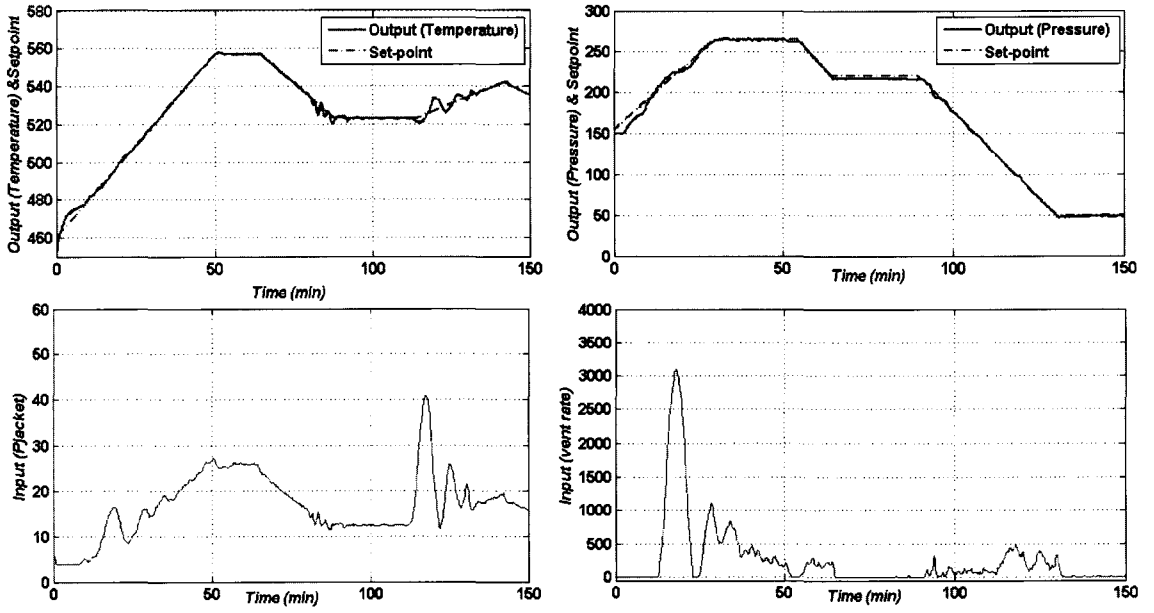


Fig. 3.16: LV-MPC based on OWTU modeling approach on the nylon 6,6 autoclave

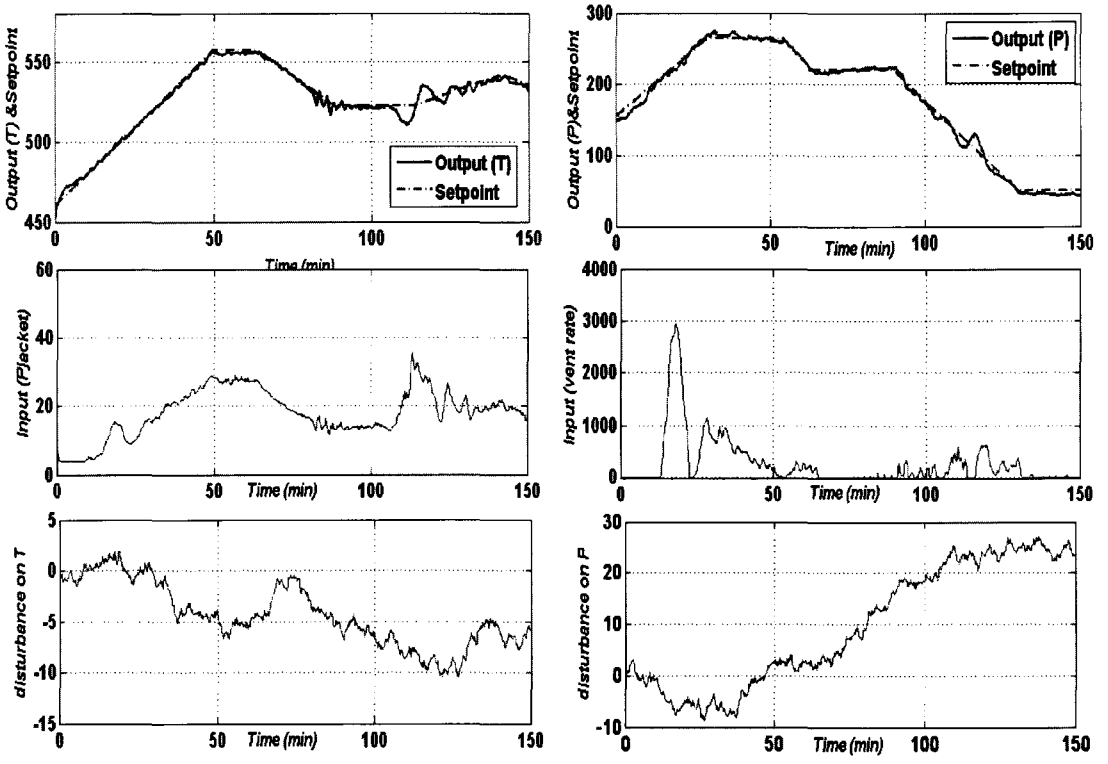


Fig. 3.17: Random Walk disturbance rejection by LV-MPC based on OWTU modeling approach on the nylon 6,6 autoclave

Regularized Batch-Wise Unfolding (RBWU)

Fig. 3.18 illustrates the performance of the LV-MPC based on the batch-wise with time-lag unfolding approach using 15 historical batches (12 similar batches plus 3 batches run by a slightly different set-point trajectory) for identification and 10 phases per batch and 5 time shift units (L=5). It tries to compensate the shortcomings of both previous modeling approaches. In fact, when the number of observations is large or the process does not have large noises (as is the case for this process) the difference between the batch-wise and batch-wise with time-shift unfolding approaches becomes minimal as explained in section 3.2. However, in the above examples, RBWU needs less than half of the observations (batch runs) used in the training dataset of BWU, but gives similar trajectory tracking. The disturbance rejection figure is omitted for the sake of briefness as it would be similar to Fig. 3.14.

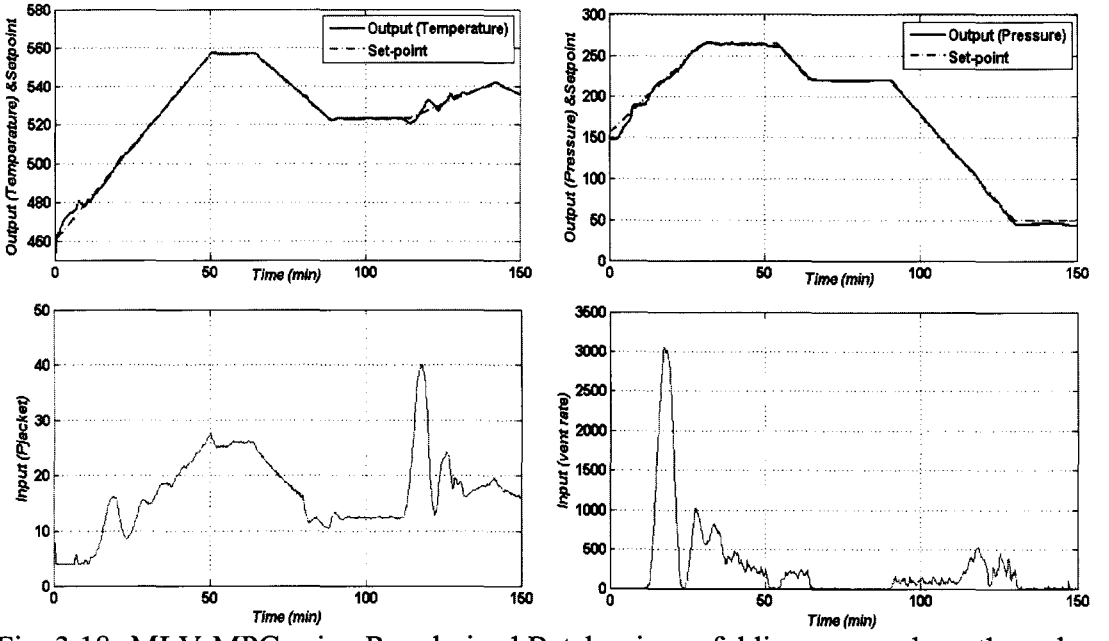


Fig. 3.18- MLV-MPC using Regularized Batch-wise unfolding approach on the nylon 6,6 autoclave using control formulation in LV space

Table 3.2- Comparison of the effect of different modeling alternatives on the set-point tracking control of the Nylon 6,6 process by LV-MPC

Performance criteria	BWU approach (T/P)	OWTU approach (T/P)	RBWU approach (T/P)
RMSE of "y-ysp"	1.0417/2.9918	1.2043/2.8300	1.0982/2.8214
STD of "Δu"	0.4667/43.9659	0.4709/43.2539	0.4425/43.3021

Table 3.2 summarizes the performance indices of testing LV-MPC on the Nylon 6,6 process for different modeling approaches used in the course of LV-MPC based on the control formulation in the latent variable space. Considering the indices for both control loops, it is seen than the RBWU approach outperforms the BWU in terms of smoothness of manipulated variable as well as the quality of tracking. The overall performance of OWTU is again inferior to the other two modeling approaches.

3.5. Conclusion

In this chapter, different alternatives for LV modeling of batch processes are scrutinized from the view point of their application for trajectory tracking in batch processes. The previously proposed two modeling approaches (BWU and OWTU) [7],[10],[6] are investigated in more details. Then, in order to receive the benefits of each modeling approach while compensating for the drawbacks of each a new LV modeling approach (RBWU) is proposed. All modeling alternatives are implemented on two case studies (a SISO and a MIMO batch process) in the course of LV-MPC.

BWU is more suitable than RBWU for modeling the nonlinearity and time-varying characteristics of batches, but needs a large number of batch runs in the training

dataset. Compared to BWU, RBWU requires fewer batch runs in the dataset, has a smoother loading matrix, and produces a smoother control action. The BWU modeling approach is suitable for the situations where there are large number of observations and the process correlation structure changes very fast over time. However, in most situations especially for chemical processes, the correlation structure in the dataset does not change so fast and RBWU approach yields almost as accurate modeling of time-varying behaviors as the BWU approach, while needing less number of observations. Thus, the newly proposed RBWU modeling approach should be preferred over BWU in general.

On the other hand, the OWTU approach requires as few as 2 batches in the training dataset and yields a smooth PCA model. However, it leads to modeling an average process dynamics. The OWTU is perfect when there are few observations in hand to build a model and the process correlation structure does not change fast over time. In the case of lack of number of observations, it is always possible to start implementing the LV-MPC using observation-wise with time-lag unfolding approach. After completing enough number of batches, one can switch to either batch-wise or batch-wise with time-lag unfolding approach.

References:

[1] T. Clarke-Pringle and J.F. MacGregor, "Nonlinear adaptive temperature control of multi-product, semi-batch polymerization reactors," *Computers & Chemical Engineering*, vol. 21, 1997, pp. 1395–1409.

[2] C. Kravaris, R.A. Wright, and J.F. Carrier, "Nonlinear controllers for trajectory tracking in batch processes," *Computers & Chemical Engineering*, vol. 13, 1989, pp. 73-82.

[3] C. Cutler and B. Ramaker, "Dynamic matrix control-A computer control algorithm," *1980 Joint Automatic Control Conference, 13-15 Aug. 1980*, New York, NY, USA: IEEE, 1980, pp. WP5-B/6 pp.

[4] J. Richalet, A. Rault, J. Testud, and J. Papon, "Model algorithmic control of industrial processes," *Digital computer applications to process control, 14-17 June 1977*, Amsterdam, Netherlands: North-Holland, 1977, pp. 103-20.

[5] C.E. Garcia, M.D. Prett, and M. Morari, "Model predictive control: Theory and practice - A survey," *Automatica*, vol. 25, 1989, pp. 335-348.

[6] J. Flores-Cerrillo and J.F. MacGregor, "Latent variable MPC for trajectory tracking in batch processes," *Journal of Process Control*, vol. 15, Sep. 2005, pp. 651-663.

[7] M. Golshan, J.F. MacGregor, M. Bruwer, and P. Mhaskar, "Latent Variable Model Predictive Control (LV-MPC) for trajectory tracking in batch processes," *Journal of Process Control*, vol. 20, Apr. 2010, pp. 538-550.

[8] F. Arteaga and A. Ferrer, "Dealing with missing data in MSPC: several methods, different interpretations, some examples," *Journal of Chemometrics*, vol. 16, 2002,

pp. 408-418.

[9] P.R.C. Nelson, P.A. Taylor, and J.F. MacGregor, "Missing data methods in PCA and PLS: Score calculations with incomplete observations," *Chemometrics and Intelligent Laboratory Systems*, vol. 35, Nov. 1996, pp. 45-65.

[10] M. Golshan, J.F. MacGregor, M.J. Bruwer, and P. Mhaskar, "Latent Variable MPC for trajectory tracking in batch processes: role of model structure," *Proceedings of American Control Conference*, St. Louis, USA: 2009, pp. 4779-4784.

[11] P. Nomikos and J.F. MacGregor, "Multi-way partial least squares in monitoring batch processes," *Chemometrics and Intelligent Laboratory Systems*, vol. 30, Nov. 1995, pp. 97-108.

[12] D.J. Louwerse and A.K. Smilde, "Multivariate statistical process control of batch processes based on three-way models," *Chemical Engineering Science*, vol. 55, Apr. 2000, pp. 1225-1235.

[13] J. Camacho and J. Picó, "Multi-phase principal component analysis for batch processes modelling," *Chemometrics and Intelligent Laboratory Systems*, vol. 81, Apr. 2006, pp. 127-136.

[14] P. Nomikos and J.F. MacGregor, "Monitoring batch processes using multiway principal component analysis," *AIChE Journal*, vol. 40, 1994, pp. 1361-1375.

[15] P. Nomikos, "Multi-variate statistical process control of batch processes," Ph.D. Thesis, McMaster University, 1995.

[16] P. Nomikos and J.F. MacGregor, "Multivariate SPC charts for batch processes," *Technometrics*, vol. 37, 1995, pp. 41-59.

[17] S. Wold, N. Kettaneh, H. Fridén, and A. Holmberg, "Modelling and diagnostics of batch processes and analogous kinetic experiments," *Chemometrics and Intelligent Laboratory Systems*, vol. 44, Dec. 1998, pp. 331-340.

[18] A. Ferrer, D. Aguado, S. Vidal-Puig, J.M. Prats, and M. Zarzo, "PLS: A versatile tool for industrial process improvement and optimization," *Applied Stochastic Models in Business and Industry*, vol. 24, 2008, pp. 551-567.

[19] T. Soderstrom and P. Stoica, *System Identification*, Prentice-Hall, 1989.

[20] L. Ljung, *System Identification: Theory for the User*, Prentice Hall PTR, 1999.

[21] J.F. MacGregor, M.J. Bruwer, and M. Golshan, "System and Methodology for the Model Predictive Control of Batch Processes using Latent Variable Dynamic Models," U.S. Patent 61/052,992, 2008.

[22] T.J. Harris, F. Boudreau, and J.F. Macgregor, "Performance assessment of multivariable feedback controllers," *Automatica*, vol. 32, Nov. 1996, pp. 1505-1518.

[23] D.J. Kozub and J.F. MacGregor, "Feedback control of polymer quality in semi-batch copolymerization reactors," *Chemical Engineering Science*, vol. 47, 1992, pp. 929-942.

[24] G. Box, G.M. Jenkins, and G. Reinsel, *Time Series Analysis: Forecasting & Control*, Prentice Hall, 1994.

[25] D.J. Kozub and J.F. MacGregor, "State estimation for semi-batch polymerization reactors," *Chemical Engineering Science*, vol. 47, Apr. 1992, pp. 1047-1062.

[26] N. Aziz, M.A. Hussain, and I.M. Mujtaba, "Performance of different types of

controllers in tracking optimal temperature profiles in batch reactors," *Computers & Chemical Engineering*, vol. 24, Jul. 2000, pp. 1069-1075.

[27] B.J. Cott and S. Macchietto, "Temperature control of exothermic batch reactors using generic model control," *Industrial & Engineering Chemistry Research*, vol. 28, 1989, pp. 1177-1184.

[28] S.A. Russell, D.G. Robertson, J.H. Lee, and B.A. Ogunnaike, "Control of product quality for batch nylon 6,6 autoclaves," *Chemical Engineering Science*, vol. 53, Nov. 1998, pp. 3685-3702.

CHAPTER 4

Identification for Control of Batch Processes Using Latent Variable Models

The contents of this chapter are accepted for presentation at the **AIChE annual meeting, 2010.** A journal paper containing the materials of this chapter will be submitted to the **Journal of Industrial & Engineering Chemistry Research**. Expected submission date is November 2010.

4.1. Introduction

Process system identification has been extensively studied since the 1970s. However, most of the results are interpreted in the context of application to time-invariant continuous processes. There are several differences between continuous and batch processes that make the identification problem of these two cases substantially different. Major differences are the non-linear, time-varying dynamics of batch systems and the finite time for each batch and the presence of finite batch runs for identification.

There are two major steps in solving a system identification problem: (1) Design of Experiment (DOE) to generate the training dataset, and (2) building a model based on the available dataset. Design of identification experiments plays an important role in satisfying the identifiability conditions and improving the quality of the identified model. References [1-4] discuss the effect of information content of the training dataset on the performance of system identification and present some optimal DOEs based on different objective functions, such as minimizing the bias in the identified model, and handling constraints. For many processes and particularly for batch processes a main interest is to solve the identification problem using the dataset collected under feedback conditions due to safety and economic reasons. Necessary and sufficient conditions for the identifiability of Linear Time Invariant (LTI) continuous systems operating in closed-loop are derived in [5-8]. The extension of identifiability conditions for nonlinear systems is presented in [9] where necessary and sufficient identifiability conditions are derived locally using locally linearized models. The identifiability (or more correctly parameter estimability) conditions for nonlinear systems where the true underlying model structure

is known are studied in references [10],[11]. However, the concepts of identifiability or estimability are not clearly defined for batch systems where a true underlying LTI empirical model or theoretical cannot be assumed. Furthermore, most identifiability concepts are asymptotic in nature and with finite time batch processes and a finite number of batch runs, asymptotic properties are not achieved.

Approaches to tackle the second step of the system identification problem, the model building step, can be categorized into two major categories: parametric and nonparametric identification methods [12],[13]. Parametric methods include approaches that impose a structure on the process model and include parsimonious models with a small number of parameters. On the other hand, nonparametric methods consist of approaches in which no specific model structure is presumed and essentially imply an infinite number of parameters. In practice one often uses models intermediate between these extremes which are parametric but have a large number of parameters and some structure. Parametric methods consist of Prediction Error Methods (PEMs) and Instrumental Variable Methods (IVMs) and can be solved within three frameworks: Direct, Indirect, and Joint input-output [14-17]. Ljung et al. and Gustavsson et al., [18],[19] proved that if a system is identifiable, the above three approaches are equivalent under asymptotic conditions. Esmaili et al. [17] show that for finite number of observations in the training dataset, direct identification approach gives better or equal performance as compared to the other two parametric approaches. Thus, using the direct approach is preferred in general. There are also several nonparametric methods to solve the identification problem; the most popular ones are Correlation Analysis Methods

(CAM) and Spectral Analysis Methods (SAM). They can also be used in the above three frameworks (Direct, Indirect, and Joint Input-Output) [12],[13],[20],[21].

There are few papers discussing the identification problems in batch processes. Shen et al. [22] tries the application of a PID controller on an empirical LTI transfer function built for a batch process case study. This approach is just a direct application of system identification for time invariant continuous system to a batch process. It does not consider the nonlinearity and time-varying effects which are very likely to occur in batch processes due to major changes of operating conditions throughout the duration of the batch processes. Ma and Braatz [23] assume that a mechanistic model is available, and investigate the effect of parameter uncertainty in the identification DOE and optimal control in batch processes. However, in many practical situations, a reliable mechanistic model is not available.

Several studies have been performed on the latent variable modeling of batch processes under non-designed experiments for the purpose of passive monitoring of the batches [24],[25]. However, these studies were aimed only at analyzing for differences among batches and predicting final product quality, whereas the current chapter is concerned with models for the set-point tracking of batch trajectories. There are no papers discussing the identifiability conditions for obtaining models for batch processes from closed-loop data nor any on the effect of the training data generation step on the model accuracy or the controller performance. In chapter 2, a Model Predictive Control (MPC) approach based on LV modeling of batch processes, called LV-MPC is proposed and some intuitive closed-loop experiments for the identification of LV models are

performed [26]. Chapter 3 investigated the properties of three different latent variable modeling approaches that can be used in the course of LV-MPC [27]. Some authors have tried to model the batch processes by Subspace Identification Methods (SIMs) [28],[29] that are closely related to latent variable models [30]. SIMs have received considerable attention in the literature, yet the studies on SIMs are mostly confined to developing LTI models which may not be satisfactory for batch process modeling and control. Verhaegen and Yu [31] proposed a new version of SIMs for modeling the Linear Time Varying (LTV) processes, however, it entails a lot of effort as it performs the local modeling at every sample time, while the approach proposed by Golshan et al.[26] models the LTV processes with much less effort because it finds local models at all sample times in one modeling step.

In the previous chapters, the LV-MPC methodologies based on different LV modeling alternatives are proposed and investigated. As explained above, the generated training dataset from the identification experiments has a substantial effect on the quality of the developed model. In the current chapter the system identification problem for batch processes are explored by emphasizing on the identification for control using LVM approaches. The focus of this study is on the DOE to satisfy pseudo identifiability conditions (see section 4.2.3) for batch processes in order to get as informative data as possible and to reduce the bias in the identified model. However, an accurate identifiability test is not possible since this study is dealing with models obtained for the batch dataset and assumes that the true underlying dynamic model is not available. Note that neither good prediction, nor good control proves identifiability. However, the

ultimate objective of this study is to find an LV model that results in the best control using LV-MPC. Thus, our ultimate test of the model would be to use it in the course of the proposed LV-MPC (to find an adequate model for control of the process). [26].

The remainder of this chapter is organized as follows: In Section 4.2, the concept of identifiability is investigated and conditions to generate an informative training dataset for batch process identification are proposed. In Section 4.3, the bias issue in closed-loop identification is studied. Then, the controller characteristics to be used for generating the closed-loop identification dataset in order to reduce the bias in the identified model by LV modeling approach are proposed. In Section 4.4, the simulation studies to illustrate the theoretical results are demonstrated. The conclusion is presented in section 4.5.

4.2. Closed-loop identifiability conditions in batch processes

The identifiability conditions refer to the situations under which the process model parameters can be estimated. This topic is widely investigated in the literature [5-7],[18],[19],[32],[33]. In all of these papers, the system is assumed to be LTI. References [9],[34],[35] have investigated the identifiability of nonlinear systems in the sense of locally linearized models. References [10],[11] review identifiability and distinguishability for nonlinear systems where the true underlying process model structure is available. In this research, the batch processes are considered to be nonlinear and time-varying systems in general and the true model structure are assumed to be unavailable in general. They are also assumed to be finite time processes and only a finite number of batch observations are available in the identification dataset. Therefore, the

formal definitions of identifiability conditions do not directly apply to the batch processes in general. However, the conventional definition for identifiability conditions can serve this study by providing the desirable conditions to be considered in the training data generation step as explained later in this section.

In the current section, the conventional definitions of identifiability conditions for LTI systems are reviewed. Most papers discuss the identifiability conditions in the context of parametric methods, and more specifically Prediction Error Methods (PEMs). However, the comparison of different identification methods is presented in this section which shows that different identification approaches are just different representation of the same approach of solving the identification problem. Thus, the identifiability conditions are the same for all identification approaches. The modeling approaches considered in this study are the latent variable modeling approaches discussed in [36-38] and the general results for identifiability conditions are interpreted in the context of batch processes.

4.2.1. Identifiability conditions for LTI systems operating in closed-loop

The most common definition of closed-loop identifiability is presented in Ljung et al.[18] as follows. Assume the data are collected from a closed-loop system as shown in Fig. 4.1, where G is the open-loop process, F is the controller, H is the noise dynamics, L is the set-point filter, e is the white noise, v is the set-point of the closed-loop system, u and y are the input and output respectively, and d is an additional dither signal that might be necessary to excite the closed-loop system.

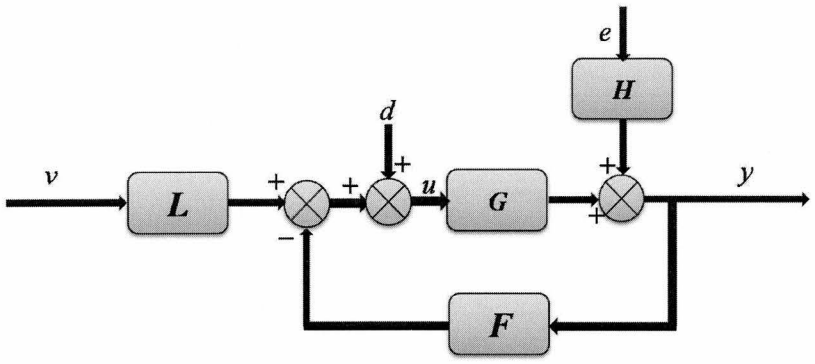


Fig.4.1: Schematic diagram of the closed-loop system in the training data generation step

Assume the system is represented as:

$$\mathcal{S}: y(t)= G_{\mathcal{S}}(q^{-1})u(t)+ H_{\mathcal{S}}(q^{-1})\, e(t) \tag{4.1}$$

and the model is assumed to be:

$$\mathcal{M}: y(t)= G\ (q^{-1})u(t)+ H\ (q^{-1})\varepsilon(t) \tag{4.2}$$

where $\mathcal{S}$ denotes System and $\mathcal{M}$ denotes Model structure. $e(t)$ and $\varepsilon(t)$ are white noises of different characteristics in general. The following definitions are necessary:

Definition 4.1[19]: $D_T(\mathcal{S},\mathcal{M})=\{\quad G\ (z)=G_{\mathcal{S}}(z)$ *and* $H\ (z)=H_{\mathcal{S}}(z)$ *a.e.* $z\}$

The set D_T consists of model parameter values that result in a model with the same process and noise characteristics as the system. Assume $\mathcal{I}$ denotes the identification method, and $\mathcal{X}$ denotes the identification experiment. Then,

Definition 4.2 [19]: The system $\mathcal{S}$ is said to be System Identifiable (SI) under $\mathcal{M}$, $\mathcal{I}$, and $\mathcal{X}$, SI$(\mathcal{M},\mathcal{I},\mathcal{X})$, if $(n;\mathcal{S},\mathcal{M},\mathcal{I},\mathcal{X})\rightarrow D_T(\mathcal{S},\mathcal{M})$ with probability 1 as $n\rightarrow\infty$.

Where is the vector of model parameters and n is the total number of observations in the training dataset.

Definition 4.3[19]: The System is said to be Strongly System Identifiable (SSI) under $\mathcal{I}$, and $\mathcal{X}$, SSI$(\mathcal{I},\mathcal{X})$, if it is SI$(\mathcal{M},\mathcal{I},\mathcal{X})$ for all $\mathcal{M}$ such that $D_T(\mathcal{S},\mathcal{M})$ is non-empty.

Definition 4.4[12]: a signal $\eta(k)$, with spectrum $S_{\eta\eta}(\omega)$, is Persistently Exciting (PE) of order "*np*" if $S_{\eta\eta}(\omega)\neq 0$ for at least "*np*" frequencies in the range $-\pi<\omega<\pi$.

This is a sufficient condition for identifying a model with "*np*" zeros and poles. However, it is preferred in practice to over specify the order of PE.

Assume that the training dataset, $\mathcal{X}$, is generated by the process input that is obtained from a set of controllers switching among "*r*" different settings:

$$u_i = L_i v - F_i y \qquad i = 1,\ldots,r \tag{4.3}$$

Based on the *Definitions 4.1-4.4* and equation (4.3), the following theorem applies [19].

Theorem 4.1[13]: for the model set defined in (4.2) and controller structure defined in (4.3) and assuming a time delay either in the system (and model) or in the controller, i.e. $G(0)F(0)=0$ and $L=I$ without loss of generality, and there is an external signal (either v or d) which is Persistently Exciting (PE) of any finite order, the necessary

and sufficient conditions to satisfy SSI in the Prediction Error Method (PEM) is as follows:

$$n_y + n_u \leq r(n_y + n_\mu) \tag{4.4}$$

where n_u and n_y are dimensions of input and output respectively and n_μ is the summation of dimensions of set-point and dither signal (v+d). The proof of the above theorem is presented in [5]. The following conclusions can be mined from the theorem 4.1[12],[13]:

1. SSI cannot be guaranteed if u is determined through a noise-free linear low order pure feedback from the output (v,d=0).
2. Simple ways to guarantee SSI for a SISO system:
   a) Use a controller that shifts between different settings during the generation of identification experiments (or equivalently use a nonlinear or time varying controller)
   b) Add a PE external signal (dither signal, d, or time-varying and PE set-point signal, v)

4.2.2. Connections between different identification approaches

Brillinger and Krishnaiah [39] show that the spectral analysis method (SAM) and Prediction Error Method (PEM) are different representation of the same approach of solving the identification problem:

$$\min_{\hat{G}} (Y - \hat{G}U) \overset{Least\ Squares}{\Longrightarrow} \begin{cases} PEM : \hat{G} = Y^T U (U^T U)^{-1} = \dfrac{\sum_k \hat{\alpha}_k e^{-i\omega k}}{\sum_j \hat{\beta}_j e^{-i\omega j}} \\ SAM : \hat{G} = \dfrac{\Phi_{yu}(\omega)}{\Phi_u(\omega)} \end{cases} \tag{4.5}$$

where Φ_u and Φ_{yu} are spectral density functions of the input and the cross-spectral density function of the input and output variables respectively. The case of " $\min H^{-1}(Y - \hat{G}U)$ " (the more frequently considered case for PEMs) can be cast as the above problem by considering the auxiliary variables $Y'=H^{-1}Y$ and $U'=H^{-1}U$ in the above analysis. Brillinger and Krishnaiah [39] claim that the spectral analysis approach conveys stronger numerical stability than PEMs because PEMs are more vulnerable in the cases where the LS assumptions are not satisfied which is likely to happen in practical situations. They also propose an algorithm that combines the linear regression approach (considered in PEMs) with the spectral analysis approach as follows. First, use the spectral analysis approach to compute a nonparametric estimation of a finite order parametric process transfer function ((ω)). Then, use the linear regression approach to determine which coefficients are important to retain and come up with another estimation of the parametric process transfer function ($\tilde{G}(\omega)$). Then based on a graph of comparison of (ω) with $\tilde{G}(\omega)$ find the best estimation of the process transfer function in an iterative procedure.

On the other hand, the basic concepts behind the LVMs and spectral analysis approach for system identification are the same. In the spectral analysis method, one performs the Fourier transform on the correlation function (an orthogonal transformation), while in LVMs one performs the Singular Value Decomposition (SVD) on the correlation function (another orthogonal transformation on the same function). Both transformations try to decompose the correlation function in several orthogonal components. The only difference is that in Fourier transformation the kernel function is

the "Complex Exponentials" function, while the kernel function in the LVMs are the Eigen Vectors of the correlation matrix of the training dataset. In fact LVM is a subdivision of the correlation analysis method. In order to obtain the transfer function in the frequency domain using spectral analysis method one uses equation (4.6), while equation (4.7) is to obtain the transfer function in the form of ARX (transfer function in the time domain) using correlation analysis method:

$$\hat{G}(e^{-i\omega}) = \frac{\Phi_{yu}(\omega)}{\Phi_{u}(\omega)} \tag{4.6}$$

$$\hat{h} = r_u^{-1} r_{yu} \tag{4.7}$$

where r_u and r_{yu} are the correlation matrix of the input variable and the cross-correlation matrix between input and output variables, and and are the estimated transfer function from SAM and CAM respectively.

The LVMs obtained from Partial Least Squares (PLS) are generalized versions of the ARX model obtained by equation (4.7) in which output is expressed as a function of transformed version (through principal components) of the past inputs, outputs, and possible extra measured variables. In Principal Component Analysis (PCA) the same concept applies with the modification of modeling the effects of future variables as well as past variables.

Brillinger [40] presented the PCA concept in the frequency domain. It is shown that the same concept for PCA applies in the frequency domain as follows:

$$X = \xi A \quad , A = (2\pi)^{-1} \int_0^{2\pi} V(\alpha) e^{it\alpha} d\alpha \tag{4.8}$$

$$\xi = XB \quad , B = (2\pi)^{-1} \int_0^{2\pi} V^*(\alpha) e^{it\alpha} d\alpha \tag{4.9}$$

where V is the matrix of Eigen vectors of the spectral density function of the time series "X", V^* is the complex conjugate of V, and ξ are the Principal Components (PCs) in the frequency domain.

The above discussion is an overview of different identification approaches and connections among them. The discussion shows that the identification algorithms are different in the sense of the space they represent the process model in. In another words they are different in the model format they produce. Thus, the results of theorem 4.1 are applicable to all modeling approaches as under SSI condition considered in the DOE step, the system is identifiable regardless of the type of model, $\mathcal{M}$, used unless the set D_T is empty for such model which means the model is structurally not able to mimic the process behavior.

4.2.3. Pseudo Identifiability conditions for batch processes operating in closed-loop

In the current study the batch processes are assumed to be nonlinear and time-varying processes and the true model structure is not available. Furthermore, they are finite time processes and only a finite number of batch observations are available for model identification, while the identifiability conditions are asymptotic properties. Thus, there is no formal definition for identifiability conditions of batch processes in general. However, it is proposed in this study to satisfy the SSI condition as the minimum requirement for the batch process identification dataset (named pseudo identifiability condition). Under such condition, it is assured that the dataset contains enough

information to find a linear process model. If the batch process happens to be a LTI system, then the formal definition for identifiability condition applies and SSI condition guarantees the identifiability of the process. On the other hand, if the batch process presents nonlinear and/or time-varying behavior, one loses the formal definition of identifiability condition. However, this time-varying behavior of naturally occurring batch operating data actually helps in satisfying these pseudo-identifiability conditions. In particular, the condition 2(a) arising from identifiability condition (4.4) states that one simple way to guarantee SSI, is to use a time-varying or nonlinear controller on a linear process. An analogous condition can exist by having a linear controller acting on a nonlinear and/or time-varying process. Hence, it can be argued that the normal operating data with no external excitation satisfies the conditions for LTI model identification under close-loop. Note that all modeling approaches including LVMs presented in this study are deviation models where the mean or set-point trajectories are removed from the batch dataset before performing the LV modeling. However, the time-varying set-point moves the operating conditions of the batch to different levels, thereby making the deviation form of the model more nonlinear and time-varying.

Another common approach to satisfy the SSI conditions for dataset obtained from close-loop system is to add an external excitation in the form of a Random Binary Sequence (RBS) signal on top of the controller output or into the set-point (condition 2(b) above). As explained above, batch processes are often identifiable from historical data without the need for additional excitation of the closed-loop system by an external signal which is an attractive result for practitioners. However, additional identifiability

conditions can be ensured by adding set-point changes. These changes should also improve the quality of the model estimation (in terms of better estimates of the model parameters).

In the rest of this section, the necessary and sufficient conditions to satisfy the SSI conditions (pseudo-identifiability) for three LV modeling alternatives incorporated in the course of the LV-MPC [27] is proposed as different observations from the theorem 4.1.

Observation-Wise with Time-lag Unfolding (OWTU)

The dataset of a batch process can be collected in a cube since there are three dimensions in the batch dataset as shown in Fig.4.2a. In order to apply a LVM such as Principal Component Analysis (PCA) that is proposed to be used in the course of LV-MPC, one needs to unfold the three dimensional dataset into a two dimensional matrix. Different unfolding approaches lead to different modeling properties. OWTU is one of the LV modeling approaches studied in chapter 3. It is illustrated in Fig. 3.2 and is repeated here in Fig.4.2 for the sake of convenience. Parameters "*ph*" and "*fh*" are past and future lags respectively considered in the OWTU approach as explained in chapter 3. This unfolding approach is a similar approach, in terms of data arrangement and modeling properties, to conventional time-series analysis and system identification studies. It leads to a LTI dynamic model of the batch process over each phase that captures the average dynamics of the batch over the considered phase. However, the main difference between LV modeling on OWTU dataset and conventional time-series analysis is that LVMs try to find a model for the complete batch (phase) considering all variables

together in the multivariate model format. To ensure identifiability condition for the PCA model built from OWTU, one needs to satisfy either of conditions 2(a) or 2(b) arising out of theorem 4.1in section 4.2.1.

Assessing the theorem 4.1 for the dataset resulted from this modeling approach, the following DOE will satisfy the SSI conditions.

Observation 4.1: "If the system is square ($n_y=n_u=n_v$), the SSI condition is satisfied if the set-point is different (time-varying) in more than "1+ph+fh" sample times along the batch (phase) completion or from batch (phase) to batch (phase)."

In this study the system is assumed to be square (the number of manipulated variables and controlled variables are assumed to be equal and there is a set-point for every output). Thus, as long as the set-point (and hence operating region) is time-varying, the non-linear and time-varying effects are included in the dataset which satisfy the pseudo identifiability condition according to condition 2(a) of theorem 4.1. However, inclusion of extra set-point trajectories that are different from the main set-point trajectory at several sample times during each modeling phase will further improve the quality of the indentified model.

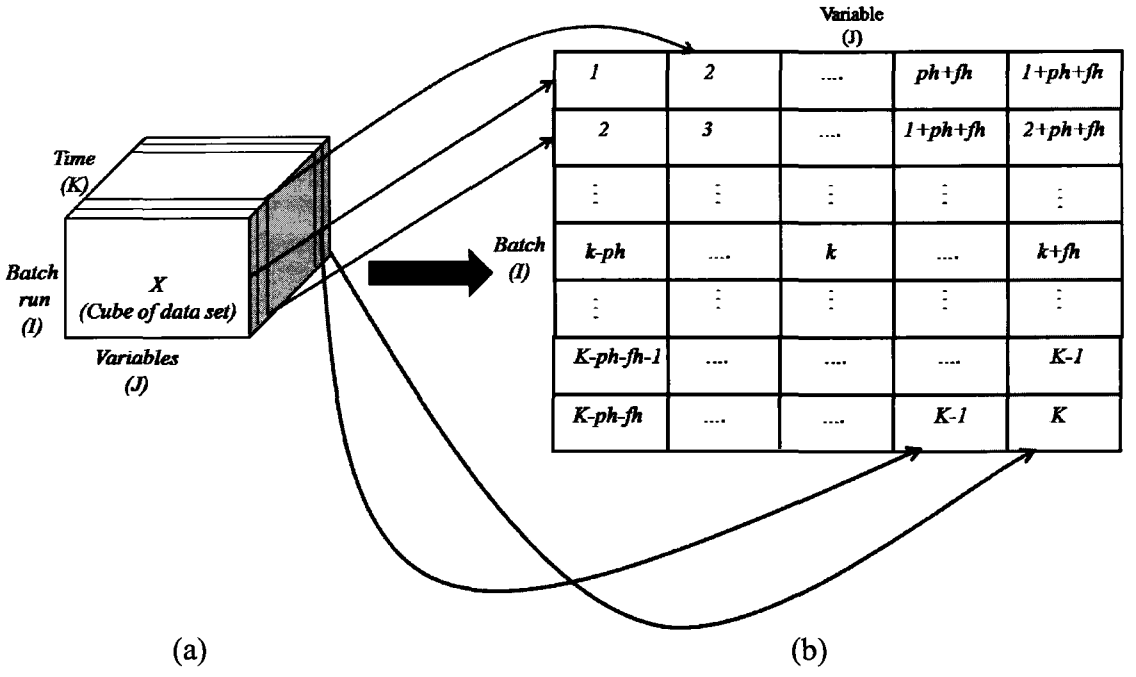


Fig. 4.2: (a) Cube of a batch process dataset (b) Observation-Wise with Time-lag Unfolded of the dataset

Note that the LVMs model the dataset by considering all columns of the matrix and extracting fewer LVs to explain the major variations in the matrix. The maximum number of LVs is equal to the number of columns. The different (time-varying) set-point in more than "*1+ph+fh*" sample times ensures that the columns of the OWTU matrix are linearly independent (OWTU is a full rank matrix). However, in order to better estimate the model parameters, it would be desirable to have the set-point trajectory different in significantly more than "*1+ph+fh*" sample times.

Another alternative to ensure identifiabilty for an LTI process is to satisfy condition 2(a) of theorem 4.1– namely switch between different controller settings during each phase (or have a time varying controller). But, as pointed out earlier, this identifiability condition would equivalently be satisfied if the batch process is a nonlinear

or time-varying process and the model is a LTI one, as it is with a PCA model on the OWTU data.

In practice, it would generally be prudent not to rely only on one of these identifiability conditions, but to satisfy both by not only relying upon the nonlinear time-varying behavior of the batch process but also by introducing one or more set-point deviations from the nominal set-point trajectories from batch (phase) to batch (phase).

Batch-Wise Unfolding (BWU) and Regularized Batch-Wise Unfolding (RBWU)

BWU and RBWU are two of the three candidates studied in chapter 3. They are shown in Figs. 4.3 and 4.4 respectively. Parameter "*L*" is the number of repetition of each batch in the RBWU as explained in chapter 3. Since these two modeling approaches share most of the same properties, they are studied together.

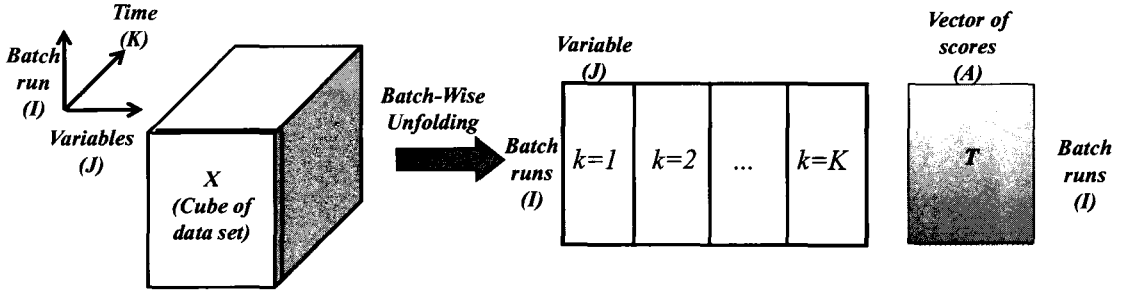


Fig. 4.3: (a) Cube of a batch process dataset (b) Batch-Wise Unfolding of a batch process dataset (c) matrix of LV scores

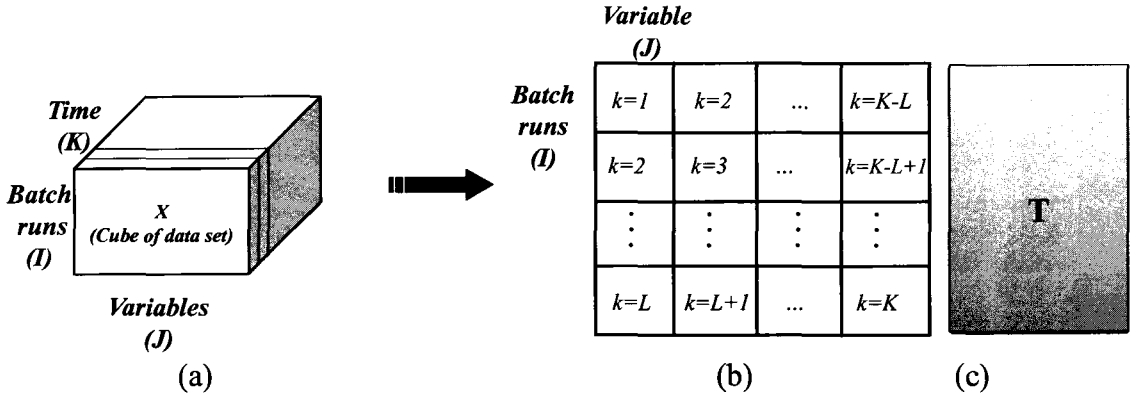


Fig. 4.4: (a) Cube of a batch process dataset (b) Regularized Batch-Wise Unfolding of a batch process dataset (c) matrix of LV scores

As explained in chapter 3, using the BWU or RBWU modeling approaches, the PCA method models the time-varying and nonlinear properties of a batch process by modeling all local variations at every sample time. As a result, the concept of identifiability condition applies only locally to the BWU and RBWU modeling approaches. This means that the SSI condition should be satisfied at every sample time for different observations (from batch to batch). Thus, assessing the theorem 4.1, the DOE to satisfy the SSI condition for BWU and RBWU datasets is as follows:

Observation 4.2: "If the system is square ($n_y=n_u=n_v$) the SSI condition is satisfied if the set-point signal is time-varying at every sample time from batch to batch in the training dataset."

In LV modeling by BWU and RBWU approaches the process is modeled locally at every sample time throughout the batch (phase). However, it is different from local

modeling of processes by conventional modeling approaches where every local model is developed regardless of other local models and thus nonlinear behavior of the batch process does not help for modeling at every sample time. The multivariate modeling approaches such as LVMs consider variables of all sample times in one model by summarizing the fat matrices in Figs. 4.3b and 4.4b in a smaller matrices of LV scores (T) (Figs. 4.3c and 4.4.c). Therefore, the nonlinear properties still help in satisfying the identifiability conditions. However, this effect cannot be quantified theoretically at this time. Thus, in order to ensure the pseudo-identifiability conditions, at least one different set-point should be considered in the dataset in the way that the different set-point has a deviation from the original set-point at all sample times so that each column contains at least one set-point change (PE of order 1 at each time step). If not (i.e. if the different set point trajectory overlapped the nominal one over a certain period), then one might lose identifiability condition during that period.

This can be achieved by running a batch using a shifted nominal set-point trajectory. However, such a set-point trajectory may result in a slightly different final product quality which may not be desirable for industrial applications. An alternative approach would be to ask an expert to help to select a modified set-point trajectory in a way that the final product quality should stay in the specified range.

4.3. Design Of Experiment (DOE) to reduce the bias in the closed-loop identification by LVMs

Closed-loop identification can often result in bias in the identified models. For parametric models such as PEM, identifiability conditions such as discussed in section 4.2.1 define the necessary and sufficient conditions for asymptotic identifiability (and hence lack of bias for large data sets) of LTI systems. However, for finite data sets as available in batch phases, one can still see bias even if the LTI identifiability conditions are satisfied. Furthermore, if the model structure is not defined correctly, the model will be a biased one. Therefore, with fitting linear models to time varying and nonlinear systems some model bias from the true system is inevitable. This problem is extensively studied for parametric methods such as PEMs [12],[13],[41],[42].

With nonparametric modeling approaches such as CAM and SAM, one avoids the problem of model structure definition, but, even for LTI systems, the method inherently includes bias under closed-loop identification [12],[13],[19].

It is shown that if one applies the nonparametric modeling approaches such as spectral density and correlation based methods to closed-loop data, a biased estimate of process transfer function is obtained [12],[13]. For a closed loop system, Fig.4.1, (assuming $L(q^{-1})=1$ without loss of generality) the following equations exist:

$$y(t)=\frac{1}{1+G_s(q^{-1})F(q^{-1})}\left[G_s(q^{-1})v(t)+\frac{1}{F(q^{-1})}z(t)\right],\ z(t)=F(q^{-1})H_S(q^{-1})e(t) \quad (4.10)$$

$$u(t)=\frac{1}{1+G_s(q^{-1})F(q^{-1})}\left[v(t)-z(t)\right] \quad (4.11)$$

Hence,

$$\Phi_u = \frac{1}{\left|1+G_s(e^{i\omega})F(e^{i\omega})\right|^2}\left[\Phi_v(\omega)+\Phi_z(\omega)\right] \tag{4.12}$$

$$\Phi_{yu} = \frac{1}{\left|1+G_s(e^{i\omega})F(e^{i\omega})\right|^2}\left[G_s(e^{-i\omega})\Phi_v - \frac{1}{F(e^{-i\omega})}\Phi_z\right] \tag{4.13}$$

Using the equations (4.10) and (4.11), the auto and cross-correlation functions are obtained for "*u*" and "*yu*" respectively as follows:

$$\gamma_u = \frac{1}{\left(1+G_s(q^{-1})F(q^{-1})\right)^2}\left[\gamma_v+\gamma_z\right] \tag{4.14}$$

$$\gamma_{yu} = \frac{1}{\left(1+G_s(q^{-1})F(q^{-1})\right)^2}\left[G_s(q^{-1})\gamma_v - \frac{1}{F(q^{-1})}\gamma_z\right] \tag{4.15}$$

As a result, the estimated transfer functions from the spectral analysis approach and the correlation analysis approach respectively are obtained by equations (4.16) and (4.17) [13]:

$$\hat{G}(e^{-i\omega}) = \frac{\Phi_{yu}}{\Phi_u} = \frac{G_s(e^{-i\omega})\Phi_v - 1/F(e^{-i\omega})\,\Phi_z}{\Phi_v(\omega)+\Phi_z(\omega)} \tag{4.16}$$

$$\hat{h}(q^{-1}) = r_u^{-1}r_{yu} = \frac{G_s(q^{-1})\gamma_v - 1/F(q^{-1})\,\gamma_z}{\gamma_v+\gamma_z} \tag{4.17}$$

Clearly, the estimated frequency response ( ) is a biased estimate of the true response (G_s) and (q^{-1}) is a biased estimate of $G_s(q^{-1})$.

The direct consequence of the above discussions is that the identified latent variable models will exhibit some bias under the proposed closed-loop identification

approach. These latent variable models are non-parsimonious models that might be expected to exhibit some characteristics of non-parametric models and their bias relationships shown in equations (4.16) and (4.17). They are also based on finite data sets from the batch phases and the true process is time-varying and nonlinear. The objective of this section and the simulation section to follow is to investigate this bias and to identify conditions under which the bias can be minimized.

Equations (4.16) and (4.17) are valid for nonparametric models of LTI systems. There is no analytical expression for bias distribution if nonparametric models apply to nonlinear time-varying systems. However, they can provide some justifications and guidelines for DOE in batch process modeling using LVMs.

From the above equations one can see that if the ratio Φ_z/Φ_v is small, the bias term is small and a close approximation of the open-loop transfer function is obtained using nonparametric methods [39]. Therefore, as long as the noise level (e) is small compared to the external excitation (v) at any frequency, there will be small bias at those frequencies. The inclusion of different set-point trajectories in the training dataset decreases the Φ_z/Φ_v ratio leading to a smaller bias.

The controller and disturbance dynamics also will have an important effect on the bias in the identified model. To better see this, equation (4.16) can be rewritten as:

$$\hat{G}(e^{-i\omega}) = \frac{\Phi_{yu}}{\Phi_u} = \frac{G_s(e^{-i\omega})\Phi_v - F(e^{i\omega})H(e^{-i\omega})\Phi_e H^T(e^{i\omega})}{\Phi_v(\omega)+\Phi_z(\omega)} \tag{4.18}$$

Equation (4.18) shows that a high magnitude of disturbance ($H(e^{i\omega})$) and controller ($F(e^{i\omega})$) at certain frequencies will increase the bias at those frequencies. The

nature of the disturbance in most batch processes is not clear due to the time-varying nonlinear behavior of the process and generally poor knowledge of how the disturbances enter and propagate. However, some general comments on disturbances and their effects according to equation (4.18) can be made for the identification studies used in this thesis:

Effect of disturbances ($H(e^{i\omega})$) on possible model bias:

The latent variable models in this study are expressed in terms of deviations of the output variables (y's) from their set-point trajectories and in terms of the deviations of other variables from their mean trajectories (see section 2.4.1). Under these conditions, any deviation of the output variables from their set point trajectories that cannot be explained by the input (u) trajectories is absorbed into the disturbance term ($H(q^{-1})e$). This disturbance term would include real disturbances arising from the propagation of different initial conditions (recipe, raw material and impurity variations) for each batch, and from disturbances entering during the progress of the batch (e.g. impurities in a feed stream entering a semi-batch process). However, under closed-loop identification with mean centering "y" by "y_{sp}", the disturbance term ($H(q^{-1})e$) would also have to absorb any offset in the y's from their set-point trajectories that arises from the inability of the existing controller to track the set-points during the runs used for closed-loop identification (a pseudo disturbance). This latter contribution can be quite large in the case where a PI controller is being used to track ramp set-point trajectories since a PI controller cannot keep up with ramp set-points. As a result this deviation will appear as a

nonstationary disturbance and will inflate $H(e^{i\omega})$ particularly at low frequencies. According to equation (4.18) this would lead to an apparent model bias in the model gain.

Effect of the controller tuning ($F(e^{i\omega})$) on possible model bias:

The nature and tuning of the controller used during the closed-loop identification studies will affect the model bias. A tightly tuned PI controller (large PI gains) will result in a large magnitude of the controller frequency response (term $F(e^{i\omega})$ in equation (4.18)), hence, increasing the magnitude of bias. However, a tightly tuned PI controller will reduce the magnitude of the pseudo disturbance term ($H(e^{i\omega})e$) discussed above by decreasing the offset in the y's from their corresponding set-points. The latter effect has a stronger impact on the bias than the magnitude of $F(e^{i\omega})$ as according to equation (4.18) the square factor of the $H(e^{i\omega})$ appears in the bias term. Thus, a tightly tuned PI controller should be preferred over a sluggishly tuned PI controller in general because it reduces the magnitude of the low frequency components in the pseudo disturbance term as discussed above. This would imply that the better the controller used during the identification experiments, the better the identified model will be and hence the better the LV-MPC designed from it will be. This raises a problem, because, the better the existing controller, the less need for a LV-MPC. In practice this probably means that one might best iterate on the model building, first using batch data collected from the existing controller, then once the improved LV-MPC is designed and implemented, collect more closed-loop data to add to the earlier data and then re-identify the model. This will also

enhance the identifiability conditions described in section 4.2 by using more than one controller for collecting identification data.

4.4. Simulation studies

To illustrate the theoretical aspects investigated in sections 4.2 and 4.3 the following simulation studies are conducted. The case study is the temperature control problem in a batch reactor studied in previous chapters. The detailed process model is presented in [26],[43],[44] and Appendix C. The schematic of the reactor is shown in Fig. 2.5.

In section 4.4.1 the simulation studies confirming the identifiability of batch processes using only historical batches in the training dataset as discussed in section 4.2 are presented. Section 4.4.2 addresses the discussions stated in section 4.3 by focusing on the PI controller in the identification data generation step. This section includes 3 simulation examples of performing the LV-MPC algorithm based on three different PI tunings to show the effect of the controller tuning in the data generation on the performance of the resultant LV-MPC.

4.4.1. Identifiability tests

As explained in Chapter 3 (Section 3.2.5) the simulation studies showed that batch processes are identifiable only from historical batches without the need for RBS excitation on top of the controller output. This observation motivated the study of current chapter. Theoretical studies presented in section 4.2 confirm that not only there is no need to RBS excitation, but also inclusion of only one different batch is often enough to get a

reasonable model based on BWU and RBWU. For the OWTU, the only requirement is the historical dataset and inclusion of the different set-point trajectory is most of the time unnecessary but helpful as explained in section 4.2.

Figs. 4.5-4.6 show the performance of the LV-MPC methodology based on the BWU modeling approach using 40 historical batches plus one batch run with a slightly different set-point trajectory. Two alternatives for considering the different set-point trajectory are tested. In Fig.4.5 the one different set-point in the training dataset is similar to the original set-point but slightly shifted, while in Fig.4.6 the different set-point is similar to the original set-point trajectory with a positive shift during first half of the batch and a negative shift in the rest of the batch as a possible example of changing the set-point without changing the final product quality. Fig.4.7 shows the shape of the different set-point trajectories studied in Figs. 4.5 and 4.6 as well as the performance of the PI controller on the two different set-point alternatives.

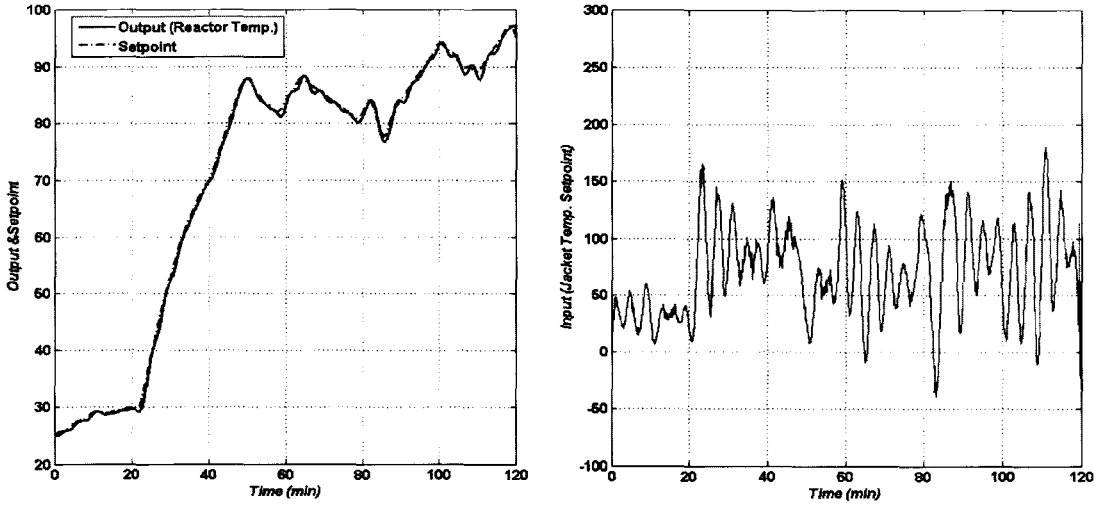


Fig.4.5: LV-MPC based on BWU modeling approach, using 40 historical batches (with PI controller) plus one extra batch run (with PI controller) on a similar set-point trajectory with a small different level throughout the batch (shifted set-point trajectory).

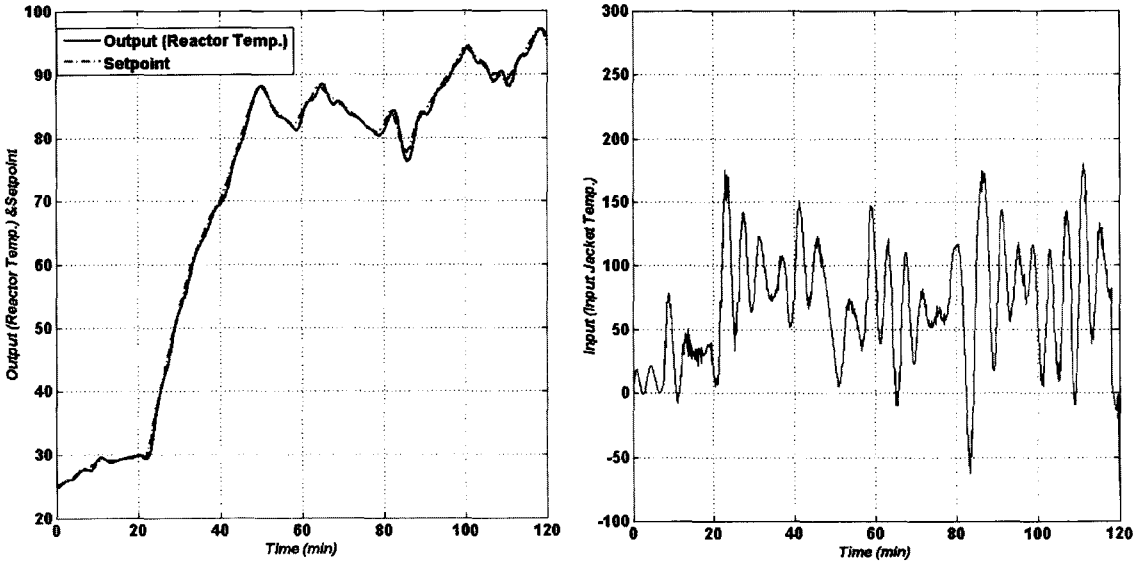
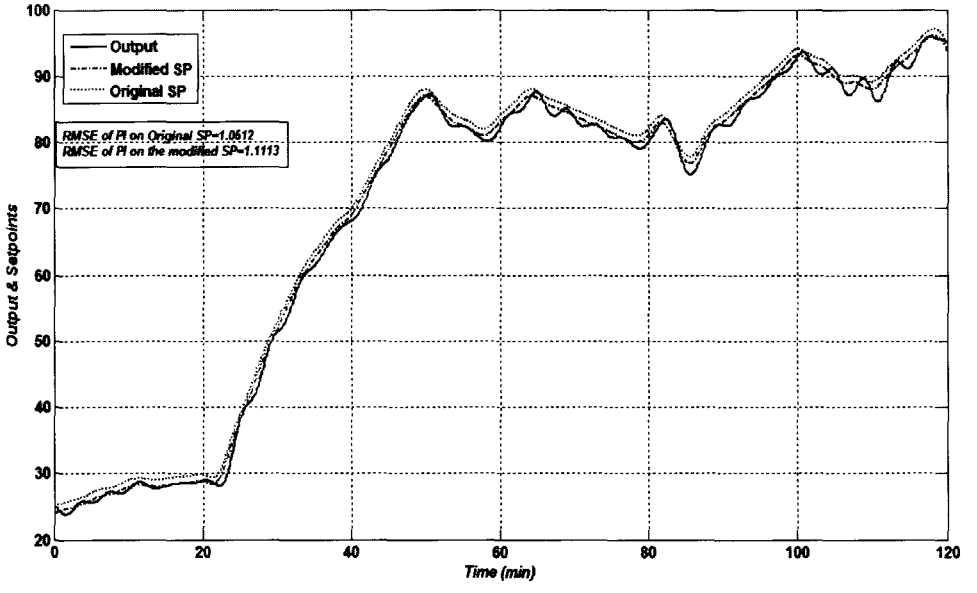
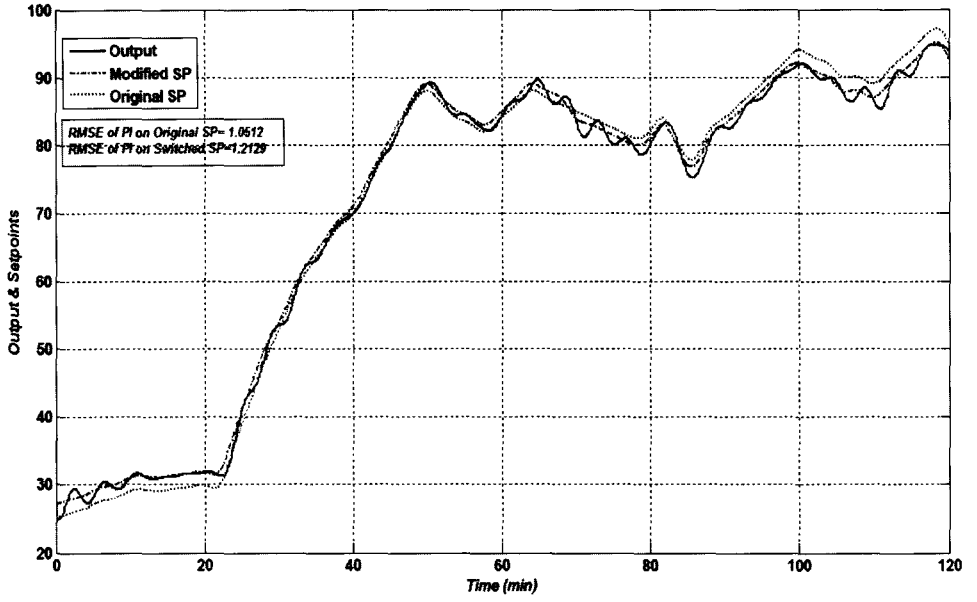


Fig.4.6: LV-MPC based on BWU modeling approach, using 40 similar batches plus one extra batch run on a similar set-point trajectory with a small different level throughout the batch. However, the level difference is positive in the first half time of the batch and negative in the second half time of the batch as a possible way to expect the final product quality to stay within the desired specification tolerance



(a)



(b)

Fig. 4.7: (a) Performance of the PI controller on the similar but slightly shifted set-point trajectory,(b) Performance of the PI controller on the set-point which is above the original set-point for the first half of the batch and then smoothly switches to below the original set-point trajectory for the rest of the batch

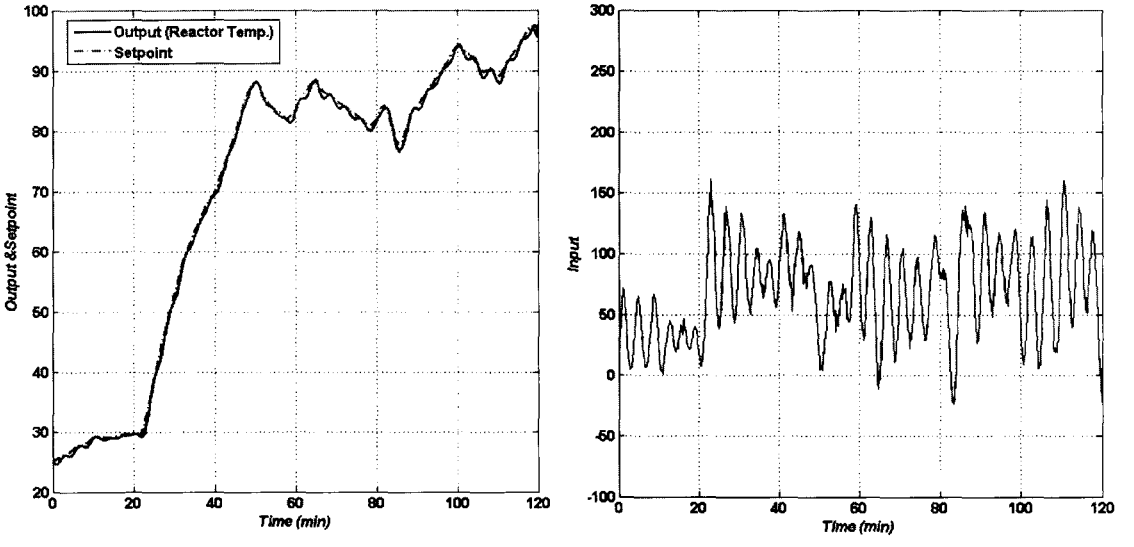


Fig.4.8: LV-MPC based on BWU modeling approach, using 40 batches with similar set-point plus one extra batch run based on a similar but slightly shifted set-point trajectory. The RBS signal is also delivered on top of the controller output for all batch runs

Fig. 4.8 is the same as Fig. 4.5 but the RBS signal is also delivered on top of the controller output in the training data generation step to excite the closed-loop system. This is a common method to satisfy the identifiability conditions [13],[45]. Table 4.1 compares the performances of Figs. 4.5-4.6 and Fig.4.8 to the noise free Nonlinear MPC (NMPC) based on a perfect mechanistic model. It shows that the qualities of all above alternatives (Figs. 4.5-4.6 and Fig.4.8) are in the same range. Specifically, Table 4.1 shows that RBS signal is not improving the performance of the LV-MPC which is an interesting finding. Industries often do not like the use of a RBS signal in the data generation step since it introduces constant variation and the resultant batch may result in off-spec product. This study shows that one can get similar results using historical batches with a modest requirement of having two different product grades or one slightly different set-point trajectory for the same product.

Table 4.1: Comparison of the LV-MPC performance based on different alternative identification dataset with the NMPC

Performance Index	Fig.4.5	Fig. 4.6	Fig. 4.8	NMPC
RMSE of $(y-y_{sp})$	0.6944	0.7041	0.6925	0.1168
STD of Δu	6.5792	6.8985	6.6046	4.5246

Fig.4.9 shows the performance of the LV-MPC based on RBWU modeling approach. It is seen that this approach leads to a similar model quality (even slightly better) to the BWU with the same DOE requirements as in the BWU, but with fewer number of batch runs in the training dataset.

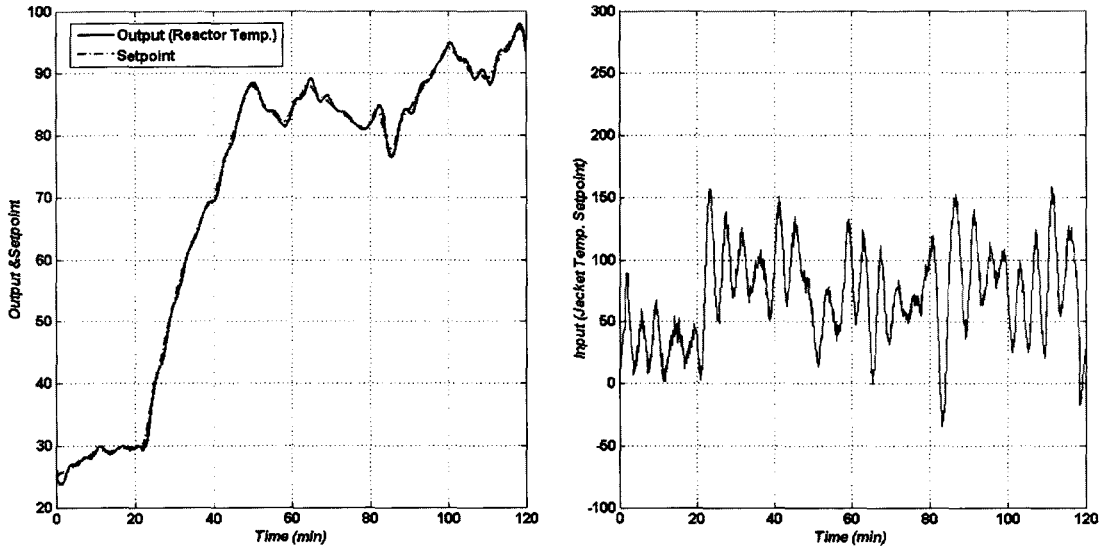


Fig.4.9: LV-MPC based on RBWU modeling approach, using 20 similar batches plus one similar but shifted set-point trajectory, *STD of* Δu*=6.5075, RMSE of* $(y-y_{sp})$*=0.6807*

Fig. 4.10 shows the performance of the LV-MPC based on OWTU using only 2 historical batches on one set-point trajectory in the dataset.

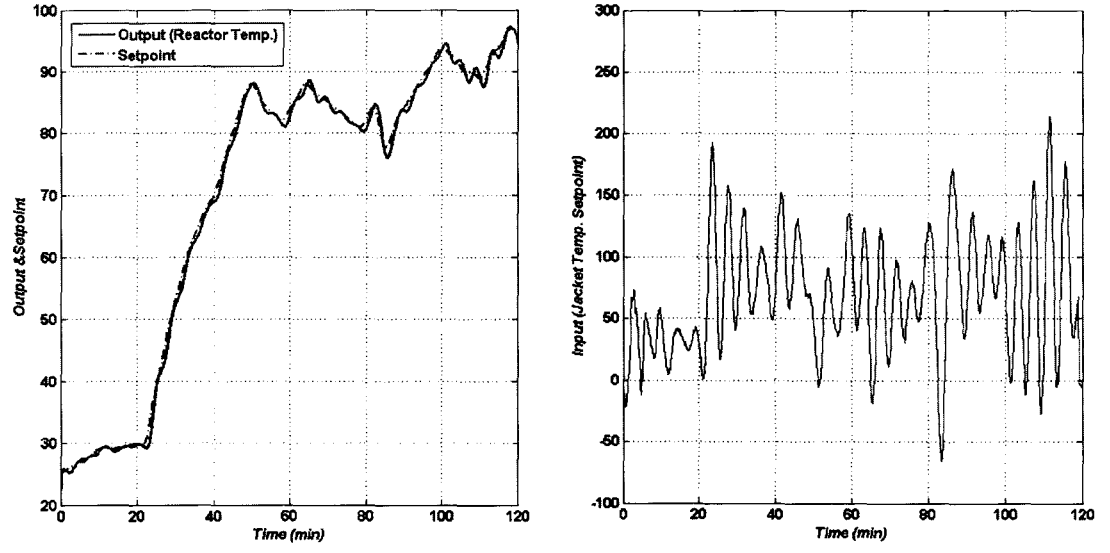
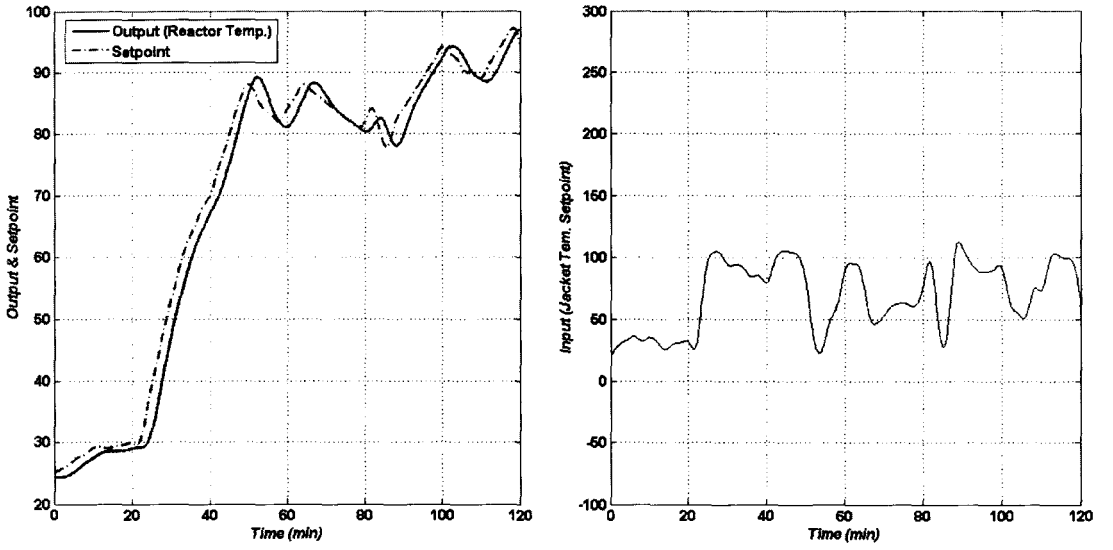


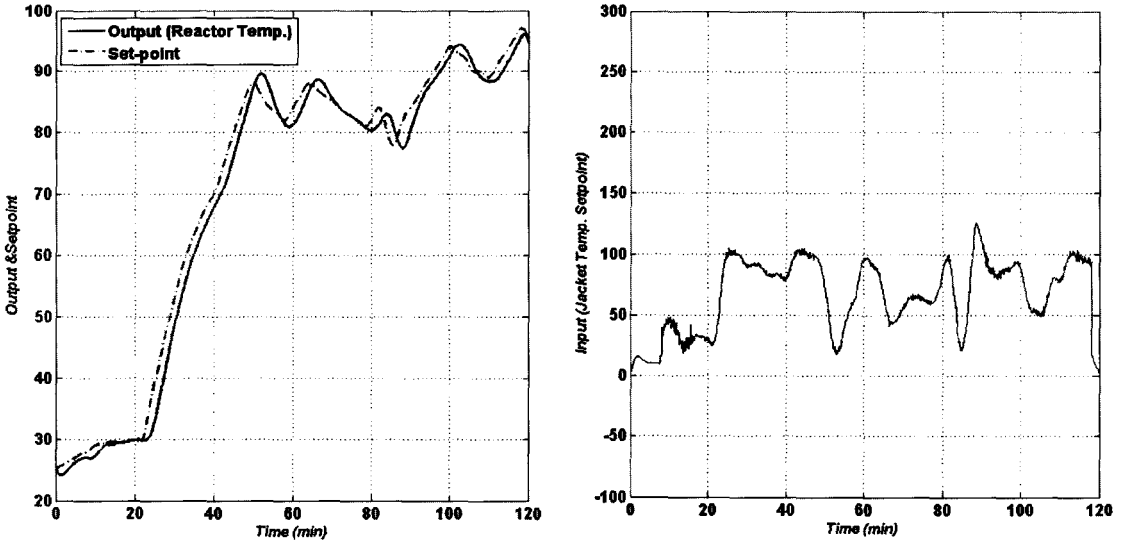
Fig.4.10: LV-MPC based on OWTU modeling approach, using 2 historical batches in the dataset run with PI controller, *STD of* Δu*=6.4766, RMSE of (y-ysp)= 0.9938*

4.4.2. Investigation of controller characteristics

In this section, the rationalization presented in section 4.3 about the effect of controller and disturbance dynamics on the magnitude of bias in the identified model will be tested by using different PI controller tunings in the training data generation step; Then, assessing the identified LV model in the course of the LV-MPC algorithm based on the BWU modeling approach using 40 batches run on similar set-point trajectory plus one batch run with slightly different set-point trajectory (as in Fig. 4.5) in the identification dataset. The noise level on top of the controlled variable (T_r) is considered to be small which is because of the accurate temperature sensors available (noise level $\approx$ 0.1-0.2 C). The LV-MPC methodology utilizes a PCA model of the batch process for the purpose of prediction. It is built based on the dataset collected under the closed-loop condition by a PI controller in the data generation step. Figs. 4.11-4.13 show the effect of the PI tuning used in the training data generation step on the performance of the resultant LV-MPC. Table 4.2 summarizes performances of the LV-MPC and PI for trajectory tracking corresponding to Figs. 4.11-4.13.

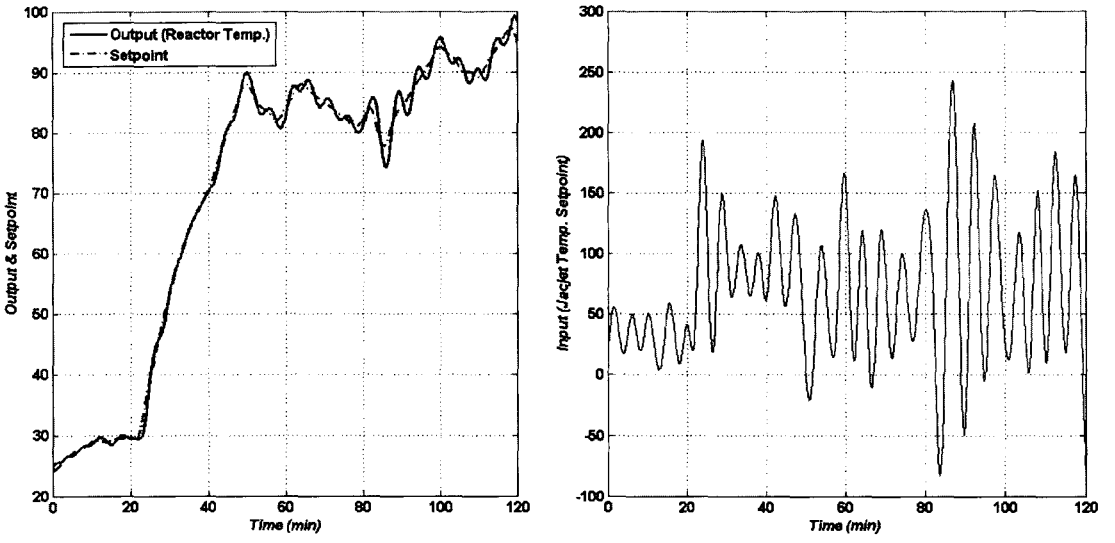


(a)

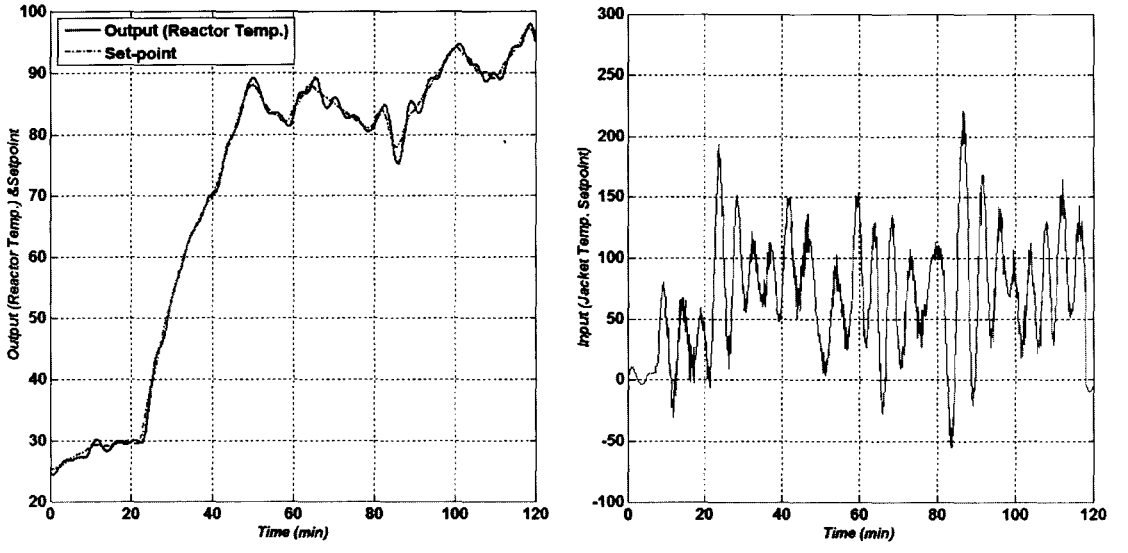


(b)

Fig.4.11: Trajectory tracking by (a) Loose PI controller used in the identification data generation step, *(Kc=10, τ_i=30)*, (b) LV-MPC based on a PCA model obtained from batch-wise unfolded closed-loop data which is generated by the loose PI controller

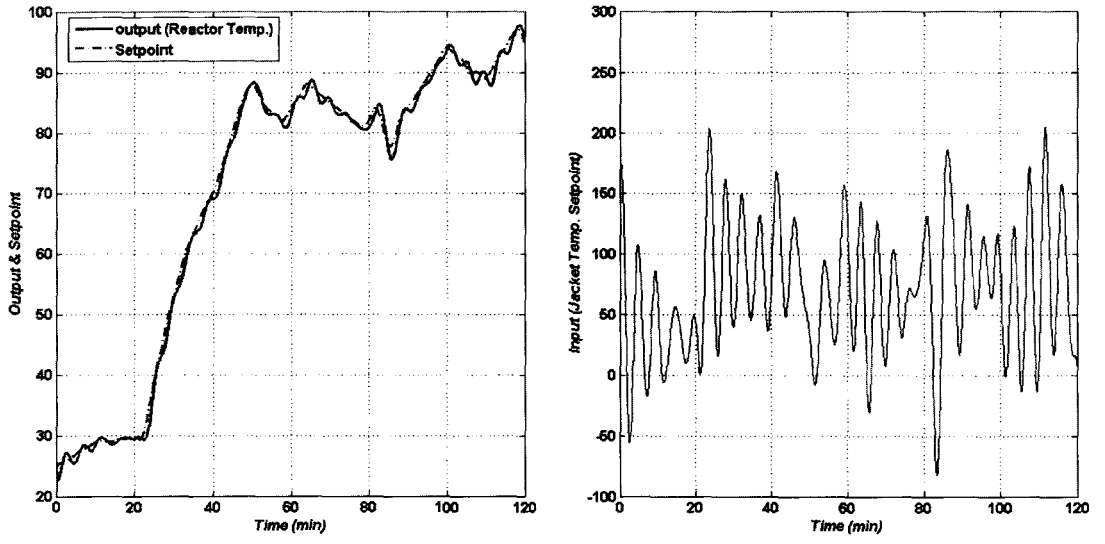


(a)

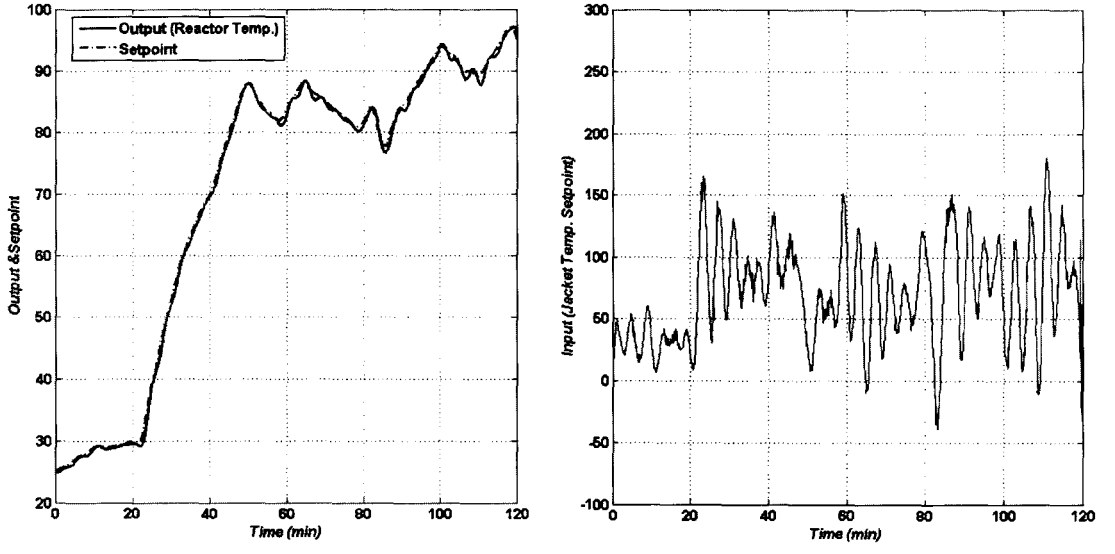


(b)

Fig.4.12: Trajectory tracking by (a) tight PI controller used in the identification data generation step with smaller Kc and larger K_i *(*$Kc=40$, $\tau_i=4$*), (b)* LV-MPC based on a PCA model obtained from batch-wise unfolded closed-loop data which is generated by the tight PI controller



(a)



(b)

Fig.4.13: Trajectory tracking by (a) tight PI controller used in the identification data generation step with larger *Kc* and smaller K_i *(Kc=60,* τ_i*=120), (b)* LV-MPC based on a PCA model obtained from batch-wise unfolded closed-loop data which is generated by the tight PI controller

Table 4.2: summary of performance of the LV-MPC based on a PCA model built on closed-loop data generated by different tunings of a PI controller

	Loose PI in the data generation step	Tight PI with smaller Kc and larger K_i in the data generation	Tight PI with larger Kc and smaller K_i in the data generation
RMSE of $(y\text{-}y_{sp})$ from PI	2.9875	1.2251	1.0512
RMSE of $(y\text{-}y_{sp})$ from LV-MPC	2.4820	0.8897	0.6944

The above simulations, show that the tightly tuned PI controller in the identification dataset with large K_c and small K_i leads to the best trajectory tracking performance by the LV-MPC and the loose PI leads to a biased model that results in an obvious bias in the trajectory tracking when the model is used in the course of LV-MPC. Flores-Cerrillo and MacGregor [46] also claimed that a fast PI controller in the identification experiment leads to a better trajectory tracking by the LV-MPC as compared to a sluggish controller in the previous version of LV-MPC methodology without further explanation of this observation. An explanation for the improved results from using a tightly tuned controller was provided in section 4.3. These simulation studies provide a confirmation of this effect. Moreover, Bakke et al. [47] investigated the effect of different PI tunings in the closed-loop training data generation on the identified model for continuous processes and also concluded that large controller gain leads to a more accurate model.

In order to better explain the results of Figs. 4.11-4.13, the frequency responses of different PI controllers used in these figures are illustrated by plotting their Bode diagram in Fig. 4.14. The ramp in the low frequency ranges of Fig. 4.14 represents the integral term. If the integral gain is large, the magnitude of the controller frequency response is large in the low to moderate frequency ranges. The horizontal lines correspond to the

proportional gain of the PI controllers which is an all frequency pass filter. Thus, a small integral gain results in fast disappearance of this term in the Bode diagram and smaller magnitude for the PI controller in the low frequency ranges of the Bode diagram. The frequency response of the transfer function of the batch process under study which is approximated by a first order process with its dominant time constant is also shown in Fig. 4.14. Fig. 4.15 shows the pseudo disturbance dynamics explained in section 4.3 resulted from the different PI controllers used for tracking ramps in Fig. 4.11-4.13 as well as the spectrum of the pseudo disturbance for the three PI examples.

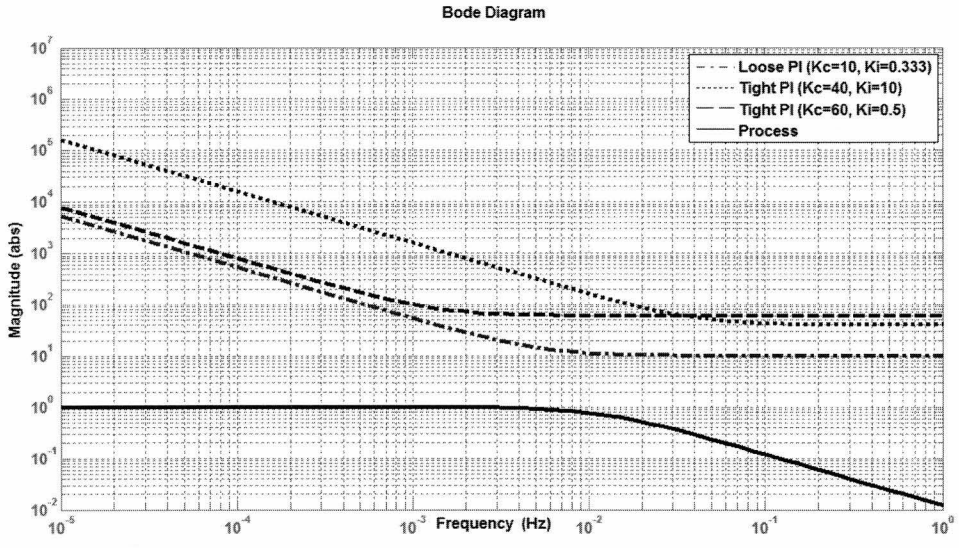
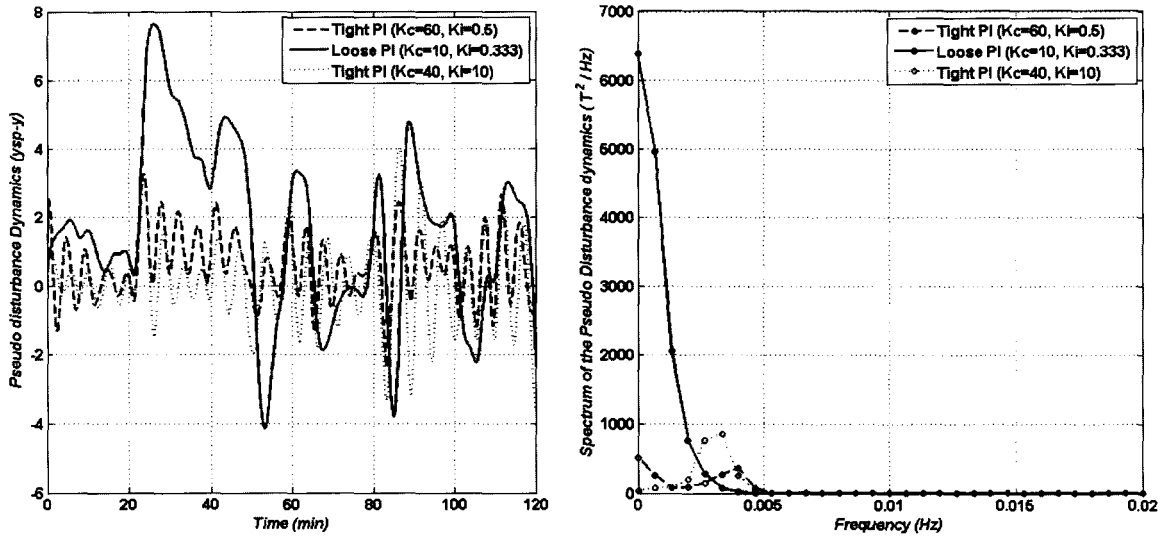


Fig. 4.14: Bode diagram for different PI controllers used in the data generation step and an approximated process transfer function



(a) (b)

Fig. 4.15: a) The pseudo disturbance resulting from mean centering y with y_{sp} b) the spectrum of the pseudo disturbance

In the process operating region which is low to moderate frequencies, the slow PI has the smallest magnitude in the Bode plot. However, according to Figs.4.11a and 4.15a, the slow PI operating on a set of ramps results in a large persistent offset in the trajectory tracking that the PI is unable to remove it. As explained in section 4.3, this persistent offset will have to be absorbed into the disturbance term $(H(e^{i\omega})e)$ when y is mean centered by y_{sp} in the modeling step (pseudo disturbance). Fig.4.15b shows that the magnitude of the pseudo disturbance resulting from the loose PI controller is significantly larger than that resulting from tight PI controllers in the low frequencies. As the square factor of the pseudo disturbance appears in the bias term (see equation 4.18), it dominates the effect of "F" and leads to a large bias. Thus a tightly tuned PI controller is preferred over a Loose PI controller.

Comparison of the two tight PI controllers according to Figs.4.14 and 4.15 shows that the tight controller with large K_i results in a larger magnitude in the low frequency region for both controller and pseudo disturbance resulting in a larger bias. Thus a tightly tuned PI controller with larger proportional gain and smaller integral gain results in the smallest bias in the identified model.

If one can use accurate feed-forward information for trajectory tracking, it is possible to decrease the gain of the feed-back controller while avoiding the large offset in the trajectory tracking. Hence the bias becomes smaller as both magnitudes of "F " and "H" get smaller in the frequency space.

It is seen in Figs. 4.11-4.13 that the LV-MPC produces a better performance for the trajectory tacking (smaller pseudo disturbance effect) as compared to the PI controller. Thus, one should use the best possible PI controller according to the above guidelines for generating the identification dataset to build the LV model and re-identify the PCA model after collecting more closed-loop data from the improved control by the LV-MPC.

4.5. Conclusion

Closed-loop identification of Batch processes has to be carried out with special considerations. The identification experiments have strong impact on satisfying the identifiability conditions for batch processes and the quality of the identified model. It is shown that although the conventional definition of identifiability condition does not

apply batch processes, they provide a set of desirable requirements for assuring the feasibility of model building task for such systems.

In this study, the identifiability of batch processes operating in closed-loop is investigated by focusing on LVMs. It is shown in section 4.2 that the strong system identifiability conditions are the same for all types of models since the SSI conditions are not dependent on the type of model and also there is no philosophical difference among different identification techniques. A set of conditions that satisfy the strong system identifiability conditions for different LVM alternatives studied in this research are proposed as observations from the general identifiability theorem. Under such conditions, the training dataset is informative enough for finding an adequate linear model for the batch process to be used in the course of the LV-MPC. It is shown that most of the batch processes are identifiable from the historical dataset and there is no need for addition of external RBS dither signal to the closed-loop system during the training data generation step. The maximum requirement would be to have an identification dataset that includes data on more than one set-point trajectories in the training dataset which is a modest requirement.

The bias issue in closed-loop identification using LVMs is also studied. Brillinger and Krishnaiah [39] showed that if the spectrum of the external noise is considerably smaller than the spectrum of the external signal (set-point), the bias will be small if the spectral analysis method is applied to closed-loop data. In this study, based on the analogy of LVMs and spectral analysis approaches, it is shown that the bias is small in LVMs under the same condition that it would be small for spectral analysis method.

Furthermore, the effect of controller tuning and disturbance dynamics on the bias term is investigated. A simulation study on different PI tunings shows that a tight PI with small integral gain and large proportional gain is suitable for the SISO temperature tracking control problem.

References

[1] G.E.P. Box and J.F. MacGregor, "Parameter Estimation with Closed-Loop Operating Data," *Technometrics*, vol. 18, Nov. 1976, pp. 371-380.

[2] L. Miskovic, A. Karimi, D. Bonvin, and M. Gevers, "Closed-loop identification of multivariable systems: with or without excitation of all references?," *Automatica*, vol. 44, 2008, pp. 2048-56.

[3] L. Ljung, "Information contents in identification data from closed-loop operation," *Proceedings of the 32nd IEEE Conference on Decision and Control. Part 3 (of 4), December 15, 1993 - December 17, 1993*, San Antonio, TX, USA: Publ by IEEE, 1993, pp. 2248-2252.

[4] U. Forssell and L. Ljung, "Identification for control: some results on optimal experiment design," *Proceedings of the 37th IEEE Conference on Decision and Control, 16-18 Dec. 1998*, Piscataway, NJ, USA: IEEE, 1998, pp. 3384-9.

[5] T. Soderstrom, L. Ljung, and I. gustavsson, "Identifiability conditions for linear multivariable systems operating under feedback," *IEEE Transactions on Automatic Control*, vol. 21, Dec. 1976, pp. 837-840.

[6] T. Ng, G. Goodwin, and B. Anderson, "Identifiability of MIMO linear dynamic systems operating in closed loop," *Automatica*, vol. 13, Sep. 1977, pp. 477-85.

[7] R. Lamanna deR, H. Geffner, and D. Cadorin, "On linear stochastic systems closed loop identifiability," *Proceedings of the 25th IEEE Conference on Decision and Control (Cat. No.86CH2344-0), 10-12 Dec. 1986*, New York, NY, USA: IEEE, 1986, pp. 2031-2.

[8] G.E.P. Box and J.F. MacGregor, "The Analysis of Closed-Loop Dynamic-Stochastic Systems," *Technometrics*, vol. 16, Aug. 1974, pp. 391-398.

[9] M. Grewal and K. Glover, "Identifiability of linear and nonlinear dynamical systems," *IEEE Transactions on Automatic Control*, vol. AC-21, Dec. 1976, pp. 883-7.

[10] K.R. Godfrey, M.J. Chapman, and S. Vajda, "Identifiability and indistinguishability of nonlinear pharmacokinetic models," *Journal of Pharmacokinetics and Biopharmaceutics*, vol. 22, Jun. 1994, pp. 229-251.

[11] E. Walter and L. Pronzato, "On the identifiability and distinguishability of nonlinear parametric models," *Mathematics and Computers in Simulation*, vol. 42, Oct. 1996, pp. 125-134.

[12] L. Ljung, *System Identification: Theory for the User*, Prentice Hall PTR, 1999.

[13] T. Soderstrom and P. Stoica, *System Identification*, Prentice-Hall, 1989.

[14] U. Forssell and L. Ljung, "Closed-loop identification revisited," *Automatica*, vol. 35, 1999, pp. 1215-1241.

[15] P. Van den Hof, R. Schrama, and O. Bosgra, "An indirect method for transfer function estimation from closed loop data," *Proceedings of 1992 31st IEEE Conference on Decision and Control, 16-18 Dec. 1992*, New York, NY, USA: IEEE, 1992, pp. 1702-6.

[16] P. Van den Hof, "Closed-loop issues in system identification," *Annual Reviews in Control*, vol. 22, 1998, pp. 173-86.

[17] A. Esmaili, J. MacGregor, and P. Taylor, "Direct and two-step methods for closed-

loop identification: a comparison of asymptotic and finite data set performance," *Journal of Process Control*, vol. 10, Dec. 2000, pp. 525-37.

[18] L. Ljung, I. Gustavsson, and T. Soderstrom, "Identification of linear multivariable systems operating under linear feedback control," *IEEE Transactions on Automatic Control*, vol. AC-19, Dec. 1974, pp. 836-40.

[19] I. Gustavsson, L. Ljung, and T. Söderström, "Identification of processes in closed loop--identifiability and accuracy aspects," *Automatica*, vol. 13, Jan. 1977, pp. 59-75.

[20] J. Welsh and G. Goodwin, "Finite sample properties of indirect nonparametric closed-loop identification," *IEEE Transactions on Automatic Control*, vol. 47, 2002, pp. 1277-92.

[21] W. Heath, "Probability density function of indirect non-parametric transfer function estimates for plants in closed-loop," *12th Symposium on System Identification, 21-23 June 2000*, Kidlington, UK: Elsevier Science, 2001, pp. 709-14.

[22] G. Shen, C. Cordeiro, and D. Vinson, "Model identification of batch processes for process control," *Proceedings of 1995 American Control Conference - ACC'95, 21-23 June 1995*, Evanston, IL, USA: American Autom Control Council, 1995, pp. 1751-5.

[23] D. Ma and R. Braatz, "Robust identification and control of batch processes," *Computers & Chemical Engineering*, vol. 27, Sep. 2003, pp. 1175-84.

[24] P. Nomikos, "Multi-variate statistical process control of batch processes," Ph.D. Thesis, McMaster University, 1995.

[25] P. Nomikos and J.F. MacGregor, "Multi-way partial least squares in monitoring batch processes," *Chemometrics and Intelligent Laboratory Systems*, vol. 30, Nov. 1995, pp. 97-108.

[26] M. Golshan, J.F. MacGregor, M.J. Bruwer, and P. Mhaskar, "Latent Variable Model Predictive Control (LV-MPC) for Trajectory Tracking in Batch Processes," vol. 20, Apr. 2010, pp. 538-550.

[27] M. Golshan and J.F. MacGregor, "Latent Variable Modeling of Batch Processes for Trajectory Tracking Control," *9th international Symposium on Dynamics and Control of Process Systems*, Leuven, Belgium: 2010.

[28] A. Dorsey and J. Lee, "Building inferential prediction models of batch processes using subspace identification," *Journal of Process Control*, vol. 13, 2003, pp. 397-406.

[29] J.H. Lee and A.W. Dorsey, "Monitoring of batch processes through state-space models," *AIChE Journal*, vol. 50, 2004, pp. 1198-1210.

[30] R. Shi and J. MacGregor, "A framework for subspace identification methods," *2001 American Control Conference, June 25, 2001 - June 27, 2001*, Arlington, VA, United states: Institute of Electrical and Electronics Engineers Inc., 2001, pp. 3678-3683.

[31] M. Verhaegen and X. Yu, "A class of subspace model identification algorithms to identify periodically and arbitrarily time-varying systems," *Automatica*, vol. 31, 1995, pp. 201-216.

[32] H. Akaike, "On the use of a linear model for the identification of feedback

systems," vol. 20, 1968, pp. 425-439.

[33] M. Gevers, "Identification for control," *Annual Reviews in Control*, vol. 20, 1996, pp. 95-106.

[34] E. Tse, "information matrix and local identifiability of parameters," *Proceedings of Joint Automation and Control Conference*, 1973, pp. 611-619.

[35] T.J. Rothenberg, "Identification in Parametric Models," *Econometrica*, vol. 39, May. 1971, pp. 577-591.

[36] E. Johansson, N. Kettaneh-wold, S. Wold, and L. Ericksson, *Multi- and Megavariate Data Analysis Principles and Applications*, Umetrics Academy, 2001.

[37] I.T. Jolliffe, *Principal Component Analysis*, Springer-Verlag, 1986.

[38] T. Kourti, "Application of latent variable methods to process control and multivariate statistical process control in industry," *International Journal of Adaptive Control and Signal Processing*, vol. 19, 2005, pp. 213-246.

[39] D.R. Brillinger and P.R. Krishnaiah, *Handbook of Statistics 3: Time Series in the Frequency Domain*, Elsevier, 1984.

[40] D.R. Brillinger, *Time Series: Data Analysis and Theory.*, Holt, Rinehart, and Winston, 1974.

[41] P.M. Van Den Hof and R.J. Schrama, "Identification and control - closed-loop issues," *Automatica*, vol. 31, 1995, pp. 1751-1770.

[42] L. Ljung and U. Forsell, *bias, variance and optimal experimetal design: Some comments on closed-loop*, Department of Electrical Engineering: Linkoping University, Sweden, 1999.

[43] N. Aziz, M.A. Hussain, and I.M. Mujtaba, "Performance of different types of controllers in tracking optimal temperature profiles in batch reactors," *Computers & Chemical Engineering*, vol. 24, Jul. 2000, pp. 1069-1075.

[44] B.J. Cott and S. Macchietto, "Temperature control of exothermic batch reactors using generic model control," *Industrial & Engineering Chemistry Research*, vol. 28, 1989, pp. 1177-1184.

[45] G. Box, G.M. Jenkins, and G. Reinsel, *Time Series Analysis: Forecasting & Control*, Prentice Hall, 1994.

[46] J. Flores-Cerrillo and J.F. MacGregor, "Latent variable MPC for trajectory tracking in batch processes," *Journal of Process Control*, vol. 15, Sep. 2005, pp. 651-663.

[47] M. Bakke, T.A. Johansen, and S. Skogestad, "Effect of Varying Controller Parameters in Closed-Loop Subspace Identification," *Proceedings of the 9th International Symposium on Dynamics and Control of Process Systems*, Leuven, Belgium: 2010, pp. 581-586.

CHAPTER 5

Conclusions and Future Work

This thesis has addressed several questions about the trajectory tracking control problem in batch processes. The main theme of this thesis is based on the synergies between the latent variable theories and the conventional process control theories. Specifically, the incorporation of LV models into the MPC algorithm is studied and then the identifiability issues are discussed for the LVMs. The current chapter is organized as follows: Section 5.1 includes the conclusions of this research. Possible future directions of this research are presented in Section 5.2.

5.1. Conclusions

In this study it is shown that the Latent Variable Models (LVMs) have excellent abilities to represent batch processes. The LVMs are efficient and less technically involved than other identification approaches such as PEMs, IVMs, and even SIMs. Moreover, the data requirements to build the LV models are modest. In this study LVMs are connected to the Model Predictive Control (MPC) algorithm through the LV-MPC methodology and it is shown that the proposed LV-MPC yields superb performance for

trajectory tracking control in batch processes. The proposed LV-MPC is also computationally faster than the NMPC which is an important property for online applications. Multiphase PCA models with overlaps are used to represent different phases of the batch rather than a single model for the entire batch in order to more focus on modeling the local variations and simplify the online computations.

Several alternatives for the LV-MPC are proposed in this research. The finalized version consists of two control formulation choices and three modeling alternatives. Consequently, there are six combinations of the control and modeling approaches. One can mix and match from the six combinations according to the practical considerations.

Two control formulation options are incorporated in the LV-MPC methodology. In the first control formulation, the optimal control problem is solved in the latent variable space to find the optimal latent variable scores of the batch (phase). Then, the vector of corresponding manipulated variables till the end of the batch (phase) is computed from the PCA model. This control formulation is called the "Infinite Horizon" control approach in the context of the batch process as explained in section 2.4. The second control algorithm, which was originally suggested by ProSensus Inc. [1] and elaborated in this study, solves the optimal control problem in the original variable space and directly finds a finite horizon of the vector of optimal manipulated variables ("Finite Horizon" control algorithm). The proposed control methodologies are tested on two simulated batch reactor case studies (a SISO example and a MIMO example) using the multiphase PCA model built on a BWU dataset. Both control formulations perform

similar with justifiable small differences as discussed in chapter 2. They yield an offset-free trajectory tracking with or without the presence of nonstationary disturbances.

Furthermore, several modeling alternatives are incorporated into the LV-MPC methodology. The first version of the LV-MPC was developed using the BWU modeling approach. BWU modeling approach is an accurate approach for modeling the nonlinear and time-varying properties of the batch process [2],[3]. However, the BWU modeling approach needs a large number of batch runs to be used in the identification dataset. The next modeling alternative proposed to be used in the course of the LV-MPC methodology is the OWTU modeling approach [4],[5]. This modeling approach enables the modeling of batch processes using as few as 2 batch runs in the training dataset. However, the PCA model based on OWTU leads to a LTI dynamic model that models the average dynamics of the batch process over the batch (phase) operating region. Then, in order to take the advantages of both of the abovementioned modeling approaches while compensating for the drawbacks of each one, a new LV modeling approach (the RBWU modeling approach) is proposed in this study. The concept of multiphase modeling is applicable to all three modeling alternatives as explained in chapter 3. All modeling alternatives are implemented on two case studies (a SISO and a MIMO batch process) in the course of LV-MPC using the control formulation in the LV space. The results are compared in chapter 3 leading to the following recommendations for practitioners to choose among the different modeling alternatives.

BWU is more suitable than RBWU for modeling the nonlinearity and time-varying characteristics of batches. However, compared to BWU, RBWU requires fewer

batch runs in the dataset, has a smoother loading matrix, and produces a smoother control action. The BWU modeling approach is suitable for the situations where there is a large number of observations and the process correlation structure changes very fast over time. However, as discussed in chapter 3 in most situations especially for chemical processes, the correlation structure in the dataset does not change very fast and the RBWU approach yields almost as accurate modeling of time-varying behaviors as the BWU approach, while needing less number of observations. Thus, the newly proposed RBWU modeling approach should be preferred over the BWU modeling approach in general.

On the other hand, the OWTU approach requires as few as 2 batches in the training dataset and yields a smooth PCA model. However, it leads to modeling an average process dynamics. The OWTU is desirable when there are few observations in hand to build a model and the process correlation structure does not change rapidly over time. In the case of insufficient number of observations, it is always possible to start implementing the LV-MPC using OWTU approach. After completing enough number of batches, one can switch to either BWU or RBWU approach. Furthermore, as explained in [4], due to the fact that in the OWTU approach, variables of different sample times are considered as different observations, one indeed has an average model valid for the entire operating region within the phase. Thus, this modeling approach can be used for trajectory tracking of a new set-point trajectory that is not even used in the dataset but is in the same range as the set-point trajectories used in the identification dataset.

In the last stage of this research, the effect of the identification dataset on the accuracy of the identified LV model is investigated. In this research, the LV models are

obtained using the direct identification approach from closed-loop data. The closed-loop data needs to satisfy certain conditions called SSI conditions in order to be informative enough for obtaining an adequate process model for the purpose of control. It is shown that although the conventional definitions of identifiability conditions do not apply to nonlinear and time-varying systems such as batch processes, they provide a set of desirable requirements for assuring the feasibility of obtaining a linear model for such systems (pseudo identifiability conditions).

A set of conditions that satisfy the strong system identifiability conditions for different LVM alternatives studied in this research are proposed in chapter 4. Under such conditions, the training dataset is informative enough for finding a linear model for the batch process. It is shown that most of the batch processes are identifiable from the historical dataset and there is no need for addition of an external RBS dither signal to the closed-loop system during the training data generation step. The maximum requirement would be to have an identification dataset that includes data on more than one set-point trajectories in the training dataset which is a modest requirement.

The bias issue in closed-loop identification using LVMs is also studied. Brillinger and Krishnaiah [6] showed that if the spectrum of the external noise is considerably smaller than the spectrum of the external signal (set-point), the bias will be small in the model identified by spectral analysis method applied to closed-loop data. In this study, based on the analogy of LVMs and spectral analysis approach, it is shown that the bias is small in LVMs under the same condition that it would be small for spectral analysis method. Furthermore, the effect of controller tuning and disturbance dynamics on the bias

term is investigated. The simulation study on different PI tunings shows that a tightly tuned PI controller with small integral gain and large proportional gain is suitable for the identification data generation from the SISO temperature tracking control problem to be used for the LV modeling.

5.2. Future Possible Directions of the Research

This thesis addresses many questions about the trajectory tracking in batch processes. However, there are still many possible research opportunities related to this work. These opportunities are briefly reviewed in this section as follows:

A) Different LV Modeling Methods: The control formulations proposed in this study are based on the PCA model as a representative of the LVMs. In this approach one considers all the variables in a training dataset and models the variations in the dataset. This approach does not give to the output variables any more attention than other variables. Shi [7] showed that there are other LVMs that focus more on the output prediction such as Canonical Correlation Analysis (CCA) and Reduced Rank Analysis (RRA). One opportunity would be to replace the PCA model in the current work by either of these two modeling alternatives.

B) Comparison of the LVMs and SIMs: Subspace Identification Methods (SIMs) are extensively studied for modeling of dynamic systems [8-12]. They have also been used in the course of model-based controllers such as LQG[13] and MPC[14]. The data arrangements considered in LVMs and SIMs are similar in many cases and they are

substantially different from what is considered in conventional system identification approaches such as Prediction Error Methods (PEMs) and Instrumental Variable Methods (IVMs). Furthermore, the main identification step in SIMs is accomplished by an LVM. In fact, SIM is basically a LVM approach followed by fitting the LV's to a state space structure. The major benefit of SIMs and LVMs over conventional identification approaches is the fact that they can handle the MIMO systems almost as simply as SISO systems. Shi [7] shows the connections between the LVMs and SIMs. The data requirements for LVMs and SIMs are also equal. Thus, it is important to investigate which model outperforms the other one in order to find the optimal LV model to be incorporated into the LV-MPC. The current PCA model as well as the modeling alternatives proposed in section (A) above (CCA and RRA) are candidates to be compared with SIMs.

C) Identification of Batch Processes: As discussed in chapter 4, the study on the identification problem for batch processes is rare in the literature. This problem is addressed in chapter 4 of this thesis. However, since this problem has not been well defined for a batch process, the scope of this thesis was to first define the problem and then address few preliminary questions related to this topic. As discussed in chapter 4, there is no identifiability condition defined for batch processes, but the pseudo identifiability condition introduced in chapter 4 will provide the desirable condition to guarantee the existence of a satisfactory model. It was shown in chapter 4 that batch processes can be modeled only from historical batches under mild conditions without the

need for external dither excitation on top of the controller output in the closed-loop system. However, the data on more than one set-point trajectory might be needed. A study on how the nature of the set-point trajectory variations improves the model identification is potentially a constructive research.

D) Systematic Selection of the number of Principal Components (PCs): the number of PCs to be selected for a specific process is a case dependent parameter in the LV-MPC algorithm. A well established approach for selection of the number of PCs is the cross validation criterion [15-17]. This approach keeps adding the PCs until degradation in the prediction power of the model appears. However, it is shown in the literature [18-20] that such stop point is suitable for building inferential models such as in soft sensor developments, but is early for the purpose of using the model for control applications. In the current research the number of PCs in the PCA model is manually tuned to produce the best trajectory tracking control by the LV-MPC. However, it is also possible to systematically determine the number of PCs. Duchesne and Macgregor [21] proposed to use the largest number of latent variables to capture the underlying model structure and stop when there is evidence that adding more PCs does not model any process structure but captures the noise. The proposed method in [21] is to continue adding PCs as long as the Sum of Square Errors (SSE) and the total variance of the parameters are both decreasing or stable and to stop when the variance of the parameters start to increase. Jackknife and bootstrap methods are metrics for the above criteria.

E) Addressing More Practical Needs: Although the current methodology considers a number of practical considerations, there remain more questions to be answered towards the preparation of a complete batch process modeling and control package. One important issue to be addressed is the fact that different batches have different durations. Thus, the collected training dataset may not be aligned. Several approaches for batch data alignments are proposed in the literature [22-24]. A possible addition to the current version of the LV-MPC methodology is a batch synchronization module in the preprocessing stage.

Furthermore, as mentioned in chapters 1 and 2, the control of batch processes constitutes a hierarchy containing a higher level control on the top of the lower level control. The higher level control analyzes the desired product quality and comes up with the corresponding process variable trajectories. The higher level control runs a few times during the batch and at each time the trajectories may be updated with slight changes because of the possible disturbances according to the Mid-Course Correction approach [25]. The updated process variable trajectories should be sent as the new set-point trajectories to the lower level control (trajectory tracking control) which can be the LV-MPC methodology proposed in this study. Thus, a super structure that allows for a systematic and adaptive interaction between the lower and higher level control in batch processes will bring the current algorithms for each level to the next stage and will constitute a complete package for batch process identification and control.

References

[1] J.F. MacGregor, M.J. Bruwer, and M. Golshan, "System and Methodology for the Model Predictive Control of Batch Processes using Latent Variable Dynamic Models," U.S. Patent 61/052,992, 2008.

[2] P. Nomikos, "Multi-variate statistical process control of batch processes," Ph.D. Thesis, McMaster University, 1995.

[3] P. Nomikos and J.F. MacGregor, "Monitoring batch processes using multiway principal component analysis," *AIChE Journal*, vol. 40, 1994, pp. 1361-1375.

[4] J. Flores-Cerrillo and J.F. MacGregor, "Latent variable MPC for trajectory tracking in batch processes," *Journal of Process Control*, vol. 15, Sep. 2005, pp. 651-663.

[5] M. Golshan, J.F. MacGregor, M.J. Bruwer, and P. Mhaskar, "Latent Variable MPC for trajectory tracking in batch processes: role of model structure," *Proceedings of American Control Conference*, St. Louis, USA: 2009, pp. 4779-4784.

[6] D.R. Brillinger and P.R. Krishnaiah, *Handbook of Statistics 3: Time Series in the Frequency Domain*, Elsevier, 1984.

[7] R. Shi, "Subspace identification methods for dynamic process modeling," Ph.D. Thesis, McMaster University, 2001.

[8] M. Verhaegen, "Application of a subspace model identification technique to identify LTI systems operating in closed-loop," *Automatica*, vol. 29, Jul. 1993, pp. 1027-40.

[9] P. Van Overschee and B. De Moor, "N4SID: subspace algorithms for the identification of combined deterministic-stochastic systems," *Automatica*, vol. 30, Jan. 1994, pp. 75-93.

[10] W. Larimore, "Canonical variate analysis in identification, filtering, and adaptive control," *Proceedings of the 29th IEEE Conference on Decision and Control (Cat. No.90CH2917-3), 5-7 Dec. 1990*, New York, NY, USA: IEEE, 1990, pp. 596-604.

[11] L. Ljung and T. McKelvey, "Subspace identification from closed loop data," *Signal Processing*, vol. 52, Jul. 1996, pp. 209-15.

[12] S. Qin, W. Lin, and L. Ljung, "A novel subspace identification approach with enforced causal models," *Automatica*, vol. 41, Dec. 2005, pp. 2043-53.

[13] W. Favoreel, B.D. Moor, M. Gevers, and P.V. Overschee, "Model-free subspace-based LQG-design," *Proceedings of the 7th Mediterranean Conference on Control and Automation*, Haifa, Israel, June 28-30, 1999, pp. 1926-1939.

[14] E.T. Hale and S.J. Qin, "Subspace model predictive control and a case study," *2002 American Control Conference, May 8, 2002 - May 10, 2002*, Anchorage, AK, United states: Institute of Electrical and Electronics Engineers Inc., 2002, pp. 4758-4763.

[15] I.T. Jolliffe, *Principal Component Analysis*, Springer-Verlag, 1986.

[16] T. Kourti, "Application of latent variable methods to process control and multivariate statistical process control in industry," *International Journal of Adaptive Control and Signal Processing*, vol. 19, 2005, pp. 213-246.

[17] E. Johansson, N. Kettaneh-wold, S. Wold, and L. Ericksson, *Multi- and Megavariate Data Analysis Principles and Applications*, Umetrics Academy, 2001.

[18] J. MacGregor, T. Kourti, and J. Kresta, "Multivariate identification: a study of several methods," *Advanced Control of Chemical Processes - ADCHEM'91,*

October 14, 1991 - October 16, 1991, Toulouse, Fr: Publ by Pergamon Press Inc, 1992, pp. 101-107.

[19] N. Ricker, "Use of biased least-squares estimators for parameters in discrete-time pulse-response models.," *Industrial and Engineering Chemistry Research*, vol. 27, 1988, pp. 343-350.

[20] B.S. Dayal and J.F. MacGregor, "Identification of finite impulse response models: Methods and robustness issues," *Industrial and Engineering Chemistry Research*, vol. 35, 1996, pp. 4078-4090.

[21] C. Duchesne and J. MacGregor, "Jackknife and bootstrap methods in the identification of dynamic models," *Journal of Process Control*, vol. 11, 2001, pp. 553-564.

[22] A. Kassidas, J.F. MacGregor, and P.A. Taylor, "Synchronization of batch trajectories using dynamic time warping," *AIChE Journal*, vol. 44, 1998, pp. 864-875.

[23] M. Fransson and S. Folestad, "Real-time alignment of batch process data using COW for on-line process monitoring," *Chemometrics and Intelligent Laboratory Systems*, vol. 84, Dec. 2006, pp. 56-61.

[24] G. Gins, J. Espinosa, I.Y. Smets, W. Van Brempt, and J.F.M. Van Impe, "Data alignment via dynamic time warping as a prerequisite for batch-end quality prediction," *6th Industrial Conference on Data Mining, ICDM 2006, July 14, 2006 - July 15, 2006*, Leipzig, Germany: Springer Verlag, 2006, pp. 506-510.

[25] J. Flores-Cerrillo and J.F. MacGregor, "Control of batch product quality by

trajectory manipulation using latent variable models," *Journal of Process Control*, vol. 14, 2004, pp. 539-553.

Appendix A. Derivation of Equation (2.18)

We first prove that although $x \neq \hat{x}$, $P^T x = P^T \hat{x}$:

If we perform Singular Value Decomposition (SVD) on the X matrix:

$$X_{I\times D} = W\Sigma Y^T = \begin{bmatrix} W_{1:A} & W_{A+1:D} \end{bmatrix} \begin{bmatrix} \Sigma_{1:A} & 0 \\ 0 & \Sigma_{A+1:D} \end{bmatrix} \begin{bmatrix} Y_1^T \\ Y_2^T \end{bmatrix} \tag{A.1}$$

From the similarity between PCA and SVD (PCA with A principal components is equivalent to SVD with the first A elements of each matrix):

PCA: $X=TP^T$ where $T=W_{1:A}\ \Sigma_{1:A}$ and $P^T=Y_{1:A}{}^T$ (A.2)

Now, the new observation, x, can be written as a linear combination of all vectors in the Y matrix (a and b are scalars):

$$x = aY_{1:A} + bY_{A+1:D} \tag{A.3}$$

Where $aY_{1:A} = \hat{x}$ is the model part of x from the PCA model. Now:

$$P^T x = P^T \left(aY_{1:A} + bY_{A+1:D}\right) = aP^T Y_{1:A} + bP^T Y_{A+1:D} \tag{A.4}$$

From (A.2) and (A.4):

$$P^T x = aY_{1:A}{}^T Y_{1:A} + bY_{1:A}{}^T Y_{A+1:D} = aY_{1:A}{}^T Y_{1:A} = Y_{1:A}{}^T aY_{1:A} = P^T \hat{x} \tag{A.5}$$

Note that we used the fact that the singular matrix Y is orthonormal.

Now we derive the equation (2.18). It is derived for a PLS model in [1]. A similar derivation for PCA is presented here:

After finding the term $\hat{\tau}_k + \Delta\hat{\tau}_k$, if the future outputs are computed using the PCA model:

$$\hat{x}_{f,k} = P_f \left(\hat{\tau}_k + \Delta\hat{\tau}_k\right) \tag{A.6}$$

However, the output computed from (A.6) does not have continuity with the past happened outputs because the past output is consistent with the not corrected score of the batch, $\hat{\tau}_k$. The objective is to bring the batch to the corrected score while keeping continuous move from not controlled batch to the controlled batch. We assume the modified score for this purpose is ϕ. Thus:

$$\hat{x}_{f,k} = P_f \varphi \tag{A.7}$$

But the objective is to bring the final score of the batch to the corrected value, $\hat{\tau}_k + \Delta\hat{\tau}_k$. From (2.17) and (A.7) we have:

$$P_f{}^T P_f \varphi = \hat{\tau}_k + \Delta\hat{\tau}_k - P_p{}^T \hat{x}_{p,k} \tag{A.8}$$

Combining (A.8) and (A.5):

$$\varphi = \left(P_f^T P_f\right)^{-1}\left(\hat{\tau}_k + \Delta\hat{\tau}_k - P_p^T x_{p,k}\right) \tag{A.9}$$

Combining (A.9) and (A.7):

$$\hat{x}_{f,k} = P_f \left(P_f^T P_f\right)^{-1}\left(\hat{\tau}_k + \Delta\hat{\tau}_k - P_p^T x_{p,k}\right) \tag{A.10}$$

Reference

[1] J. Flores-Cerrillo and J.F. MacGregor, "Control of batch product quality by trajectory manipulation using latent variable models," *Journal of Process Control*, vol. 14, 2004, pp. 539-553.

Appendix B. Controllability and Observability in the PCA Model

Controllability

Assume the system can be modeled by a LTV state space model as:

$$\eta(k+1) = A(k)\eta(k) + B(k)u(k) \quad , \quad k \geq k_0 \quad , \quad \eta \in \mathbb{R}^n \tag{B.1}$$

Where η is the vector of n-dimensional process states and $A(k)$ and $B(k)$ are coefficient matrices in the state space model. By recursive computation of the states, the controllability matrix can be found to be [1]:

$$\Psi = \left[\Gamma(k_1,k_1)B(k_1-1)\vdots\Gamma(k_1,k_1-1)B(k_1-2)\vdots\cdots\vdots\Gamma(k_1,k_0+1)B(k_0)\right] \tag{B.2}$$

Where Γ is the State Transition Matrix and is defined as:

$$\Gamma(k_1,k_0) = A(k_1-1)A(k_1-2)\cdots A(k_0+1)A(k_0) \tag{B.3}$$

It is shown that the above system is controllable in the interval $[k_0,k_1]$ if the matrix Ψ spans the n dimensional space [1]:

$$rank\left[\Psi\right] = n \tag{B.4}$$

It is shown in chapter 2 that in the PCA model, the prediction of the process states (Principal Components) are obtained as follows [2,3]:

$$\hat{t}_k = \beta_{1:k}x_{P,k} \quad , \quad t \in \mathbb{R}^A \tag{B.5}$$

$$\hat{t}_{k+1} = \beta_{1:k+1}x_{P,k+1} = \beta_{1:k}x_{P,k} + \beta_{k:k+1}x_{P,k+1} = \hat{t}_k + \beta_{k:k+1}x_{P,k+1} \tag{B.6}$$

where β is the coefficient in the missing data imputation. Assume $x_{P,k}$ is a vector consisting of all the past measurements (input (u), Output (y), and possible extra measurements (m)) from the beginning of the batch (batch phase) up to the sample time k:

$$x_{P,k} = [u, y, m]_{1:k} \tag{B.7}$$

Thus,

$$\hat{t}_{k+1} = \hat{t}_k + \beta_{u,k:k+1} u_{k+1} + \beta_{y,k:k+1} y_{k+1} + \beta_{m,k:k+1} m_{k+1} \tag{B.8}$$

where $\beta_{u,k:k+1}$, $\beta_{y,k:k+1}$, and $\beta_{m,k:k+1}$ are corresponding coefficients of the $\beta_{k:k+1}$ for u, y, and m respectively. The only variable that is manipulated is the input variable, u, and other variables change as a result of manipulating the input. Thus, third and fourth terms in equation (B.8) can be accumulated in an auxiliary variable Λ.

$$\hat{t}_{k+1} = \hat{t}_k + \beta_{u,k:k+1} u_{k+1} + \Lambda(k+1) \quad , \quad \Lambda(k+1) = \beta_{y,k:k+1} y_{k+1} + \beta_{m,k:k+1} m_{k+1} \tag{B.9}$$

It is seen that the equation (B.9) is a similar equation to the state space model, but includes non-causal term Λ and also it is non-causal with respect to u which is why PCA model is called non-causal model. However, from the mathematical point of view, it is similar to equation (B.1). Starting from time k_0 the controllability matrix defined in equation (B.2) can be found in the following way:

$$\hat{t}_{k_0+1} = \hat{t}_{k_0} + \beta_{u,k_0:k_0+1} u_{k_0+1} + \Lambda(k_0+1)$$

$$\begin{aligned}\hat{t}_{k_0+2} &= \hat{t}_{k_0+1} + \beta_{u,k_0+1:k_0+2} u_{k_0+2} + \Lambda(k_0+2) \\ &= \hat{t}_{k_0} + \beta_{u,k_0:k_0+1} u_{k_0+1} + \beta_{u,k_0+1:k_0+2} u_{k_0+2} + \Lambda(k_0+1) + \Lambda(k_0+2) \\ &= \hat{t}_{k_0} + \beta_{u,k_0:k_0+1} u_{k_0+1} + \beta_{u,k_0+1:k_0+2} u_{k_0+2} + \Omega(k_0+2)\end{aligned}$$

where $\Omega(k_0+f) \equiv \Lambda(k_0+1) + \Lambda(k_0+2) + \cdots + \Lambda(k_0+f)$.

If the above recursive formula is repeated:

$$\hat{t}_{k_0+k_1} = \hat{t}_{k_0} + \beta_{u,k_0:k_0+1} u_{k_0+1} + \beta_{u,k_0+1:k_0+2} u_{k_0+2} + \cdots + \beta_{u,k_0+k_1-1:k_0+k_1} u_{k_0+k_1} + \Omega(k_0+k_1)$$

$$\hat{t}_{k_0+k_1} = \hat{t}_{k_0} + \begin{bmatrix} \beta_{u,k_0:k_0+1} & \beta_{u,k_0+1:k_0+2} & \cdots & \beta_{u,k_0+k_1-1:k_0+k_1} \end{bmatrix} \begin{bmatrix} u_{k_0+1} \\ u_{k_0+2} \\ \vdots \\ u_{k_0+k_1} \end{bmatrix} + \Omega(k_0+k_1) \qquad \text{(B.9)}$$

Thus, the controllability matrix for the PCA model, $\overline{\Psi}$, is

$$\overline{\Psi} = \begin{bmatrix} \beta_{u,k_0:k_0+1} & \beta_{u,k_0+1:k_0+2} & \cdots & \beta_{u,k_0+k_1-1:k_0+k_1} \end{bmatrix} \qquad \text{(B.10)}$$

Note that equation (B.10) can be obtained from equation (B.2) as well by considering $A(k)=I$ for all k and thus $\Gamma(i,j) = I$ for the PCA model for all i,j according to equation (B.3). Thus, the PCA model is controllable in the interval $[k_0,k_1]$ by manipulating the input variable, u, if the matrix $\overline{\Psi}$ spans the A-dimensional space:

$$rank\left[\overline{\Psi}\right] = A \qquad \text{(B.11)}$$

Observability

All empirical models including LVMs and specifically PCA only model the observable subspace, which is reflected in the measured variables collected in the training dataset.

However, in order to obtain observability condition in the prediction step which is the missing data imputation step one should consider enough lags in the missing data imputation so that the right hand side of the equation (B.5) spans the A-dimensional space. Thus, the PCA model is observable in the interval $[k_0,k_1]$ if:

$$rank\left[\beta_{k_0:k_0+k_1}\right] = A \tag{B.12}$$

Note that in the equation (B.12) the β coefficient for all measurements (not just u) is involved. Thus, observability is achieved faster than controllability.

References

[1] J. Klamka, *Controllability of Dynamical Systems*, Springer, 1991.

[2] P.R.C. Nelson, P.A. Taylor, and J.F. MacGregor, "Missing data methods in PCA and PLS: Score calculations with incomplete observations," *Chemometrics and Intelligent Laboratory Systems*, vol. 35, Nov. 1996, pp. 45-65.

[3] F. Arteaga and A. Ferrer, "Dealing with missing data in MSPC: several methods, different interpretations, some examples," *Journal of Chemometrics*, vol. 16, 2002, pp. 408-418.

Appendix C. Process Model for the Temperature Control Case Study

The case study is a batch reactor in which the following exothermic reactions take place

$$A + B \rightarrow C \qquad A + C \rightarrow D \tag{C.1}$$

The model equations for the batch reactor are as follows:

$$\frac{dM_A}{dt} = -R_1 - R_2 \tag{C.2}$$

$$\frac{dM_B}{dt} = -R_1 \tag{C.3}$$

$$\frac{dM_C}{dt} = +R_1 - R_2 \tag{C.4}$$

$$\frac{dM_D}{dt} = +R_2 \tag{C.5}$$

$$\frac{dT_r}{dt} = \frac{(Q_r + Q_j)}{M_r C_{pr}} \tag{C.6}$$

$$\frac{dT_j}{dt} = \frac{(T_j^{sp} - T_j)}{\tau_j} - \frac{Q_j}{V_j \rho_j C_{pj}} \tag{C.7}$$

$$R_1 = K_1 M_A M_B \tag{C.8}$$

$$R_2 = K_2 M_A M_C \tag{C.9}$$

$$k_1 = \exp\left(\frac{k_1^1 - k_1^2}{T_r + 273.15}\right) \tag{C.10}$$

$$k_2 = \exp\left(\frac{k_2^1 - k_2^2}{T_r + 273.15}\right) \tag{C.11}$$

$$Q_r = -\Delta H_1 R_1 - \Delta H_2 R_2 \tag{C.12}$$

$$M_r = M_A + M_B + M_C + M_D \tag{C.13}$$

$$Cpr = \frac{\left(C_{pA} M_A + C_{pB} M_B + C_{pC} M_C + C_{pD} M_D\right)}{M_r} \tag{C.14}$$

$$Q_j = UA\left(T_j - T_r\right) \tag{C.15}$$

The parameters of the above model are given in Table C.1.

Table C.1- the constant parameter in the reactor model

Parameter	*Value*
C_{pA}	*18.0 kcal/kmol°C*
C_{pB}	*40.0 kcal/kmol°C*
C_{pC}	*52.0 kcal/kmol°C*
C_{pD}	*80.0 kcal/kmol°C*
ΔH_1	*-10000.0 kcal/kmol*
ΔH_2	*-6000.0 kcal/kmol*
C_p	*0.45 kcal/kg°C*
C_{pj}	*0.45 kcal/kg°C*
U	*9.76 kcal/ min m^2 °C*
ρ_j	*1000.0 kg/m^3*
K_1^1	*20.9057*
K_1^2	*10000*
K_2^1	*38.9057*
K_2^2	*17000*
V_j	*0.6921 m^3*
A	*6.24 m^2*
Δt	*0.2 min*
τ_j	*2.0 min*
W_r	*1560.0 kg*