RUN-TO-RUN MODELLING AND CONTROL OF BATCH PROCESSES

A thesis submitted to The University of Manchester for the degree of

Doctor of Philosophy

in the Faculty of Engineering and Physical Sciences

2015

By

Carlos Alberto Duran Villalobos

School of Electrical and Electronic Engineering

Table of Contents

List of Tables

List of Figures

Abbreviation

ANN	Artificial Neural Networks
CV	Cross Validation
FIR	Finite Impulse Response
FPLS	Functional Partial Least Squares
HMM	Hidden Markov Models
ICA	Independent Component Analysis
ILC	Iterative Learning Control
KDR	Known Data Regression
LV	Latent Variable
MLR	Multiple Linear Regression
MPC	Model Predictive Control
MPCA	Multiway Principal Component Analysis
MPLS	Multiway Partial Least Squares
MS	Multivariate Statistical
MSPC	Multivariate Statistical Process Control
RMSECV	Root Mean Square Error of Cross Validation
RMSEP	Root Mean Square of Prediction
MVT	Manipulated Variable Trajectory(ies)
NIR	Near Infrared Spectroscopy
NIPALS	Nonlinear Iterative Partial Least Squares
NOC	Normal Operation Conditions
ODE	Ordinary Differential Equations

PAT	Process Analytical Technology
PC	Principal Components
PCA	Principal Component Analysis
PCR	Principal Components Regression
PID	Proportional Integrative Derivative
PLS	Partial Least Squares
PLS+F	Partial Least Squares and Filtering
PMP	Projection to the Modal Plane
PPLS	Penalized Partial Least Squares.
PRBS	Pseudo-Random Binary Sequence
PRESS	Predictive Residual Sum of Squares
QP	Quadratic Programming
S. cerevisiae	Saccharomyces cerevisiae
SIMPLS	Simple Implementation for the PLS method
SPE	Square Prediction Error
SVM	Support Vector Machines
TSR	Trimmed Scores Regression
VC	Validity Constraints

Notation

$A \in \mathbb{R}$	Number of Latent variables
$\boldsymbol{A} \in \mathbb{R}^{n \times m}$	Matrix of constraints for the QP
$\boldsymbol{b} \in \mathbb{R}^{n}$	Vector of constraints for the QP
$\boldsymbol{E} \in \mathbb{R}^{I \times JK}$	Matrix of residuals from the predictors
$\boldsymbol{e} \in \mathbb{R}^{JK}$	Vector of residuals from the predictors
$\boldsymbol{F} \in \mathbb{R}^{I \times nqV}$	Matrix of residuals from the responses
$H \in \mathbb{R}$	Number of experiments for the results
$I \in \mathbb{R}$	Number of batches
$J \in \mathbb{R}$	Number of input variables
$J_e \in \mathbb{R}$	Residual validity indicator
$J_t \in \mathbb{R}$	Score validity indicator
$K \in \mathbb{R}$	Time intervals
$L_e \in \mathbb{R}$	Residual validity limit
$L_t \in \mathbb{R}$	Score validity limit
$\boldsymbol{lb} \in \mathbb{R}^{K}$	Vector of lower bound constraints
$\boldsymbol{M} \in \mathbb{R}^{n \times n}$	Symmetric matrix for the QP
$N \in \mathbb{R}$	Number of batches effectively used
$nqV \in \mathbb{R}$	Number of quality variables
$\boldsymbol{P} \in \mathbb{R}^{A \times JK}$	Matrix of loadings for the predictors
$\boldsymbol{Q} \in \mathbb{R}^{A \times nqV}$	Matrix of loadings for the responses
$s \in \mathbb{R}$	Sampling interval
$\boldsymbol{T} \in \mathbb{R}^{I \times A}$	Matrix of scores

$\mathbf{t} \in \mathbb{R}^{A}$	Vector of scores
$\boldsymbol{U} \in \mathbb{R}^{I\times A}$	Matrix of qualities scores
$\mathbf{u} \in \mathbb{R}^{n}$	Vector of manipulated variables
$\mathbf{u} \in \mathbb{R}^{n}$	Vector of manipulated variables
$\mathbf{u}_{\boldsymbol{n}} \in \mathbb{R}^{n}$	Vector of nominal manipulated variables
$\mathbf{ub} \in \mathbb{R}^{K}$	Vector of upper bound constraints
$V_{ini} \in \mathbb{R}$	Initial volume
$V_{max} \in \mathbb{R}$	Maximum volume
$\boldsymbol{V} \in \mathbb{R}^{JK\times A}$	Matrix of weights for the bilinear model
$\boldsymbol{V}_{\boldsymbol{uf}} \in \mathbb{R}^{JK\times A}$	Matrix of weights for the MVT after PMP
$v \in \mathbb{R}$	Variables of the process
$\boldsymbol{W} \in \mathbb{R}^{JK\times A}$	Orthogonal matrix of weights
$\mathbf{x} \in \mathbb{R}^{JK}$	Vector of predictors
$\boldsymbol{X} \in \mathbb{R}^{\mathrm{I}\times \mathrm{J}K}$	Matrix of predictors
$\boldsymbol{y} \in \mathbb{R}^{I}$	Vector of responses
$\boldsymbol{Y} \in \mathbb{R}^{\mathrm{nqV}\times J}$	Matrix of responses
$y_{sp} \in \mathbb{R}$	End-point quality set-point (desired end-point quality)
$y \in \mathbb{R}$	End-point quality
$\hat{y} \in \mathbb{R}$	End-point quality prediction
$z \in \mathbb{R}$	Number of batches sin the initial data-set
$\Delta\mathbf{u} \in \mathbb{R}^{K}$	Vector of adjustments in the MVT
$\varepsilon \in \mathbb{R}$	Sensor noise in the end-point quality
$\lambda \in \mathbb{R}$	Forgetting factor
$\boldsymbol{\eta} \in \mathbb{R}^{JK}$	Vector of predictor variables after PMP
$\boldsymbol{\pi} \in \mathbb{R}^{R}$	Vector of decision variables in the optimisation problem

Sub index

f	Estimated variables after the control action
k	Current batch
n	Nominal parameters
p	Measured variables before the control action
u	Manipulated variables

Abstract

The University of Manchester

Carlos Alberto Duran Villalobos

Doctor of Philosophy in the Faculty of Engineering and Physical Sciences

December 2015

This thesis presents an innovative batch-to-batch optimisation technique that was able to improve the productivity of two benchmark fed-batch fermentation simulators: Saccharomyces cerevisiae and Penicillin production. In developing the proposed technique, several important challenges needed to be addressed:

For example, the technique relied on the use of a linear Multiway Partial Least Squares (MPLS) model to adapt from one operating region to another as productivity increased to estimate the end-point quality of each batch accurately. The proposed optimisation technique utilises a Quadratic Programming (QP) formulation to calculate the Manipulated Variable Trajectory (MVT) from one batch to the next. The main advantage of the proposed optimisation technique compared with other approaches that have been published was the increase of yield and the reduction of convergence speed to obtain an optimal MVT.

Validity Constraints were also included into the batch-to-batch optimisation to restrict the QP calculations to the space only described by useful predictions of the MPLS model. The results from experiments over the two simulators showed that the validity constraints slowed the rate of convergence of the optimisation technique and in some cases resulted in a slight reduction in final yield. However, the introduction of the validity constraints did improve the consistency of the batch optimisation.

Another important contribution of this thesis were a series of experiments that were implemented utilising a variety of smoothing techniques used in MPLS modelling combined with the proposed batch-to-batch optimisation technique. From the results of these experiments, it was clear that the MPLS model prediction accuracy did not significantly improve using these smoothing techniques. However, the batch-to-batch optimisation technique did show improvements when filtering was implemented.

Declaration

No portion of the work referred to in this Thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright statement

Dedicatory

Dedicated to my parents, Rafael and Bertha, my brother Rafael, my sister Deni, and all my friends, who have supported, inspired and encouraged me throughout my life.

"Experientia docet" - Tacitus

Acknowledgements

First, my sincere gratitude to my supervisor, Professor Barry Lennox, for his guidance, support and insightful advice given throughout the development of this research project. Also, I would like to thank him for the time he dedicated to provide invaluable suggestions during the writing stage of this Thesis.

I am also grateful to the National Council of Science and Technology of Mexico (CONACyT) for providing the scholarship number 214965, the National Polytechnic Institute of Mexico (IPN), and the Public Ministry of Education of Mexico (SEP), for their financial support to pursue my PhD degree with the scholarship number 43901.

Special thanks to PhD David Lauri Pla and PhD Qiaolin Yuan for their experience and useful discussions for the development of the methodologies implemented in this thesis.

I am very grateful with my family for the support they provided me from overseas and with my friends Alejandro Villaruel, Jana Kefurtova, Luiza Lense and Kinga Smolenska for their invaluable support and the great moments we lived together in the UK.

Publications

[102]: C. Duran-Villalobos, B. Lennox, Iterative Learning Modelling and Control of Batch Fermentation Processes, Proceedings of the 10th IFAC International Symposium on Dynamics and Control of Process Systems, 2013: pp. 511–516.

[Under review]: C. Duran-Villalobos, B. Lennox, D. Lauri Multivariate Batch to Batch Optimisation of Fermentation Processes Incorporating Validity Constraints, Journal of Process Control.

Chapter 1: Introduction

The purpose of this chapter is to present a brief summary of the content of the thesis including: the motivation for the research in Section 1.1, the objectives for the research in Section 1.2, a list of the main contributions in Section 1.3 and the outline of the following chapters in Section 1.4.

1.1 MOTIVATION

Typically in industry, batch processes are considered superior alternatives to continuous operations when a high standard of product quality is required, when there is variability in the production rate and when the cost of building a new processing plant is too high to make a significant profit from the production. Fed-batch (or semi-batch) processing is a type of operation that is similar to batch processing, but involves the addition of feed or other materials over time based on a profile or a *recipe* [1]. Generally, in the work presented in this thesis, the term batch will be used in a wide sense for both fed-batch and batch processing.

Batch processing consists of iterative operations that perform a sequence of one or more steps to obtain a high value quantity of a specialty product (such as pharmaceutical products, high purity chemicals, cosmetics, etc.). Generally speaking, one batch is ‘run’ by introducing raw materials into a vessel and processing them through several stages that each follow predefined instructions until the desired end-product is obtained. Within batch processing, productivity, product quality and safety requirements are operational issues of great concern due to the economic industry objectives. Therefore, it is an engineering requirement to develop an optimal control strategy to address these issues if the benefit-cost relationship is to be maximized.

According to [2], batch processes have 3 important defining characteristics which must be considered when designing a control system: absence of a steady-state, limited corrective action and repetitive nature.

The first characteristic does not allow the use of a single operation point since the batch is usually divided into several stages. The second characteristic implies that if the batch presents an abnormal behaviour and if no corrective action is taken in time, the batch will be lost. The third characteristic is related to an iterative run of batches, which makes it possible to use information from previous batches to improve the behaviour in future batches. As a result of these characteristics, and others, such as highly non-linear dynamics and unmeasured disturbances [3], batch operations are especially difficult to control and often impossible to improve using traditional Proportional-Integrative-Derivative (PID) type control systems. This is because PID assumes that the process is linear and is insufficient to characterise the plant under a wide range of operating conditions [4].

Batch processes are regularly controlled using recipes based on specific process knowledge [2]. However, in practice the knowledge of the process is incomplete and the presence of unknown disturbances, which were not considered explicitly in the design, may significantly affect the output quality. To effectively implement the control of the product quality these recipes would need to be adjusted to the changing conditions from one batch to the next which would be impractical using a single *recipe* approach.

A potential solution to the operational issues of batch processes is to utilise a dynamic modelling approaches based on multivariate statistics, such as Partial Least Squares (PLS). Multivariate statistical models are relatively simple mathematical representations that describe the dynamics of a batch using empirical data; these models allow us to use the data from previous batches to make future predictions during future batch runs. Until recently, Multivariate Statistical Process Control (MSPC) techniques have been used to monitor and optimise batch processes in an attempt to improve the quality of the final product [5]. However, the statistical models can be used to solve a cost function whose objective is to increase the profit/cost relationship for example.

If batch quality optimisation is to be performed *within* a batch, then Model Predictive Control (MPC) is usually employed. In MPC, the actuator sequence is calculated every time that the process reaches an operational point. On the other hand, if the optimisation is performed from one batch to the next, Iterative Learning Control (ILC) is usually applied; ILC iteratively computes the trajectory of the actuator for the full batch offline.

Using ILC or MPC techniques raises several issues that need to be addressed in the control strategy. For example, one common problem encountered, when building a model, is to make sure that the accuracy of the model is adequate to guarantee an acceptable prediction. Similarly, the degree of freedom in the optimisation constraints must be high enough to allow the search for new ways to solve a problem but not so high that the model of previous batches is no longer valid. If such techniques are to be applied in a control strategy, issues relating to the accuracy of prediction should be addressed properly.

The main challenges in designing batch to batch, and particularly iterative learning optimisation techniques lies in choosing and integrating suitable control systems in to the process. Such control should take into consideration the natural characteristics of a batch and the issues commonly encountered within the industry. Moreover, the objective of batch optimisation, which is focused on increasing the productivity within the limits of the plant, must also be included in the control strategy. Thus, there is a powerful incentive in batch optimisation to search for innovative and better ways to integrate iterative and model-based control techniques.

1.2 Aims and objectives

The primary aim of this thesis is to develop an innovative batch-to-batch optimisation scheme, using multivariate statistical models, that has the ability to adapt the inputs of a certain batch process to reject undesired disturbances, and to optimize the final quality from one batch run to the next.

Although recently published articles have shown impressive advances in batch to batch control, a secondary objective of this thesis is to improve the results obtained in similar approaches by comparing this thesis approach with different benchmark software simulations used in relevant literature.

The work presented in this thesis fulfils other secondary objectives which must be taken in consideration to demonstrate the completion of the main objectives:

- To understand the mathematical and practical limitations of diverse techniques used in batch process control, such as those found in the relevant literature regarding MPC, ILC and adaptive control.

- To develop an optimisation routine that solves a performance criteria problem for various batch conditions. Such a mathematical problem must include the physical constraints of the system.

- To design a recursive identification algorithm that considers the effect of incremental changes in the process variables to optimise the output from one batch to the next.

- To include model validity constraints in the optimisation routine to ensure that the predictions of the process model remain valid under different operating conditions.

- To compare and analyse the results of different smoothing techniques in combination with PLS models for the proposed batch to batch optimisation.

- To simulate and objectively compare the proposed batch-to-batch optimisation technique with analogous approaches.

1.3 CONTRIBUTIONS AND SIGNIFICANCE

The major contribution of this research is the design of an innovative batch to batch optimisation technique, which provides a way to respond to changes in initial conditions and react to disturbances that affect the process from one batch to the next. The main advantage of this design was the increase of yield in a number of simulations that were selected to test the system when applied to realistic and generic batch processes. In industry, this improvement would likely lead to the possibility of significant competitive advantages in batch production.

Other contributions of this work are:

- The formulation and analysis of a Quadratic Programming (QP) problem to find an optimal Manipulated Variable Trajectory (MVT). The proposed formulation presents a QP problem that is solved in the MVT space. This is in contrast to alternative formulations found in the literature which solve the QP problem in the Latent Variable (LV) space [6]–[9].The drawback of the LV approach is that once the optimized points are found, it is necessary to compute the real MVT by inverting the PLS model, which can cause actuation changes that are detrimental to the yield if the PLS model is not sufficiently constrained. In most of the work presented in this thesis, the QP problem was solved in the MVT space.

- The inclusion and analysis of validity restrictions in the QP problem in Chapter 6. The proposed design use validity restrictions inside the MVT optimization to limit the solution of the QP problem to the region within which there is confidence in the predictions made by the PLS model. Similar validity restrictions used have been used before for MPC in the literature [10], [11]. However, in this thesis, validity restrictions are fitted to the proposed methodology for batch to batch optimization.

- The results and analysis of the smoothing techniques applied to the proposed batch to batch optimisation in Chapter 7. During this research, it was found that the correlation in time that low filtered signals have, provides improvements in the predictions made by PLS models. This thesis provides a performance evaluation of smoothing techniques when used within the proposed batch to batch optimisation.

1.4 OUTLINE OF THE THESIS

This chapter has introduced the key aspects of this thesis, its motivations, objectives and contributions. The chapter that follows, Chapter 2, moves on to presenting an overview of the literature review of the fields of study related to this research.

Chapter 3 describes the materials and methods used to obtain the results presented in this thesis; this description includes the case studies used to test the different proposed algorithms.

Chapter 4 provides the main theoretical framework used for the batch-to-batch optimisation proposed in this thesis, and used in Chapters 5, 6 and 7.

The following three chapters are the core of this thesis, where a theoretical framework is introduced for each chapter, followed by the case study results and discussion:

In Chapter 5, an innovative batch to batch optimisation scheme is proposed. This chapter starts with mathematical definitions of previous studies and proposes a batch to batch optimisation strategy. This strategy is then analysed and the results are compared with those in previous studies.

Chapter 6 provides an extension for the strategy formulated in Chapter 5, which includes validity constraints in the optimisation problem. This change in the formulation ensures that the optimisation is performed in the region of validity of the model. Moreover, the results are compared and discussed using different constraint specifications.

Chapter 7 presents a study of smoothing techniques applied to the strategy formulated in Chapter 5. This study compares the results of three smoothing techniques, one of which is a post-processing filtering algorithm; another is based in functional analysis, while the third one smooths the factors within the PLS model itself.

Finally, Chapter 8 presents conclusions of this thesis and guidelines for future work.

Chapter 2: Recent Advances in Batch Process Control

This chapter summarizes the literature review relevant to the work in this thesis. Firstly, Section 2.1 introduces the main control techniques employed in batch processes and provides an overview of the different published modelling approaches used for the control of batch processes. Secondly, Section 2.2 provides examples and improvements in multivariate statistical control of batch processes. Thirdly, section 2.3 presents some of the advances that have been made with regard of the extensions formulated in Chapters 6 and 7. Finally, Section 2.4 provides a brief summary of this chapter.

2.1 BATCH PROCESS MODELLING

The characteristics of batch processes that were discussed in Chapter 1 present problems that require a control system that differs from continuous processes in two ways [2]: The first difference is that control signals in batch processes are typically set in accordance to time varying profiles, as steady-state operating points do not exist. The second difference is an additional type of output (usually named product quality, output quality or end-point quality), which needs to be controlled and, that often is only available at the end of the batch.

These differences impose a type of control that needs to allocate a sequence of discrete time varying variables in pre-set trajectories [5], often called recipe. Usually, process variable trajectories, for those variables which are measured, are driven to their pre-defined set-points using Proportional-Integral-Derivative (PID) control to keep them under Normal Operation Conditions (NOC) [12]. However, this type of control can only exploit completely the characteristics of batch processes if it is supported by mathematical modelling and optimization techniques [5].

Published studies have already shown the benefit of applying modelling and optimization techniques to industrial batch processes. For example, in case studies undertaken by Li et al. [13] and Abel et al. [14] significant savings in operational times were reported following the application of modelling and optimisation techniques to an industrial batch process.

With respect to modelling, which is typically the first stage in the development of any monitoring, control or optimisation system, three model-based approaches have been proposed in the literature: knowledge-based (or white-box or fundamental), data-based (or black-box or empirical) and hybrid models. Such approaches are discussed respectively in Sections 2.1.1, 2.1.2 and 2.1.3.

2.1.1 Knowledge-based approach

Modelling techniques establish mathematical relationships between parameters and the dynamical behaviour of a physical process. In the knowledge-based approach, such relationships are based on simplified mathematical theories or laws such as the conservation of energy or mass. This enables engineers to evaluate potential changes in control operations over a specific process.

In the batch control area, several studies have applied this type of approach with beneficial results. For example, the authors of the already mentioned studies presented in [13] and [14], used dynamic models of a distillation process and an exothermic polymerization respectively, combined with optimization algorithms to reduce operational times.

Other authors have used knowledge-based models for optimization monitoring and control of batch processes, such as in [3], [15]–[17]. For instance in [15], the authors combined a fundamental model of batch nylon 6,6 autoclave with inferential control strategies to reduce the effect of typical disturbances, Their results showed the usefulness of this control strategy, however, the authors also acknowledged that developing accurate first principle models could be prohibitive under certain circumstances.

One important reason to use knowledge-based modelling is that it has the potential to allow the extrapolation of the applicability of such models beyond the regions of empirical models. Therefore, knowledge-based models are usually used as benchmark simulations to assess robust control techniques. For example, the two case studies used in the work presented in this thesis are based on this type of modelling technique.

The first case study used in this thesis was the Saccharomyces cerevisiae (S. cerevisiae) production simulation, which utilises a fundamental model proposed in [4]. This model describes the aerobic growth of S. cerevisiae and focuses on the overflow metabolism at pyruvate and acetaldehyde branch points. Similarly, Pham and Larsson [18] proposed a different model based on the overflow metabolism of S. cerevisiae including an inhibitory effect of ethanol. Jobé et al [19] developed this modelling work further and proposed a S. cerevisiae production model which focused on the oxidative and oxidoreductive growth.

The second case study used in this thesis was the penicillin production simulation presented in [20]. The author of this article developed a detailed simulator that extends a model reported earlier [21] with additional input variables such as pH and Temperature. A related simulation, which was industrially validated was presented in [22], where the authors extended the model reported by [23] to include the main environmental effects in penicillin production. Further details of the two simulations used in the work described in this thesis can be found in Chapter 3.

As can be observed from the examples given above, the development of knowledge based models requires a new problem formulation for each case study; hence the control technique based solely in this type of modelling may be time consuming and challenging. A further drawback with knowledge-based models is that they can deliver inaccurate predictions if the complex dynamics involved in the process are not considered because of the prohibitive time it may require to develop them.

2.1.2 Data-based approach

Data-based modelling approaches rely on the use of empirical data in order to identify process models. Data-based models are obtained by designing experiments, during which a process is perturbed to generate data. This data is then used to calculate the model parameters; this procedure is usually called empirical identification or, simply, identification. As a result, the identified-process model can provide a dynamic relationship between previously selected input and output variables [24].

One of the most commonly used methods for identifying data-based models of continuous processes uses process reaction curves. The process reaction curve method involves the analysis of response curves to determine control parameters, such as those used within Proportional-Integral-Derivative (PID) controllers. The problem with this approach in batch control is that the process needs to reach a steady-state in order to collect the response data and the dynamics of the process may change throughout the batch. Nevertheless, some process reaction curve techniques have been published to overcome this problem, such as the one proposed in [25]. In this article, the author assumed that the information obtained from *referential evolution* was correspondent to the information contained within process reaction curves for stationary process. However, he discussed only the main concept from a practical point of view, not taking into account the effects of possible disturbances.

Multivariate Statistical (MS) modelling has been shown to provide an accurate and flexible approach to batch process control. This type of method uses multivariate analysis to represent observed data via a model. To create these models, many authors have proposed and applied different methods, such as Principal Component Analysis (PCA) [26], Partial Least Squares (PLS) [27], [28], Independent Component Analysis (ICA) [29], Artificial Neural Networks (ANN) [30], Support Vector Machines(SVM) [31], Hidden Markov Models (HMM) [32], among others.

One of the most commonly used MS methods in batch process monitoring is Principal Component Analysis (PCA). This technique was first described in [33], as a method for reducing a high-dimension dataset into lower-dimensional dataset by “finding the closest fit to systems of points in space”. This idea was further developed in [34], naming this lower dimensional dataset as Principal Components (PC) and providing an extensive derivation of this method. Principal Components

Analysis (PCA) has been used in many applications such as psychology, numerical analysis, electrical engineering, image analysis, chemistry, among others [35].

For application to batch processing, Nomikos and MacGregor [26] were the first to publish the use of PCA as a monitoring tool; they used a technique referred to as Multiway PCA (MPCA), developed by Wold and Geladi [36], to extract the information from the trajectories of process variables to create simple monitoring PC-based charts. With these charts, they were able to track the progress of successive batch runs and find possible "upsets" in a simulation study of a batch reactor for the production of styrene-butadiene latex. Since then, several studies have used MPCA models for monitoring and fault detention in batch processes. MacGregor and Cinar [37] provide a thorough review of this field of applied research.

MPCA modelling is a very useful technique in monitoring and fault-tolerant control of batch process. In optimisation, however, the MPCA model itself does not provide the relationship between the explanatory (measured and manipulated) variables and the dependent (quality) variable that are necessary to solve an optimization problem. One way to deal with this problem still using PCA, is to perform a regression of the dependant variable onto the PCs of the quality variables. This technique is known as PCR (Principal components regression) [38]. An important use of PCR, is when techniques such as MLR (Multiple Linear Regression) suffers from multicollinearity (as PCs are orthogonal to one another they are unaffected by collinearity) or when the computing power is not sufficient to deal with the problem and it is necessary to reduce the model complexity.

A similar MS technique employed widely in applications of batch monitoring and more recently control is known as Partial Least Squares (PLS) regression [39]. PLS regression seeks to find lower-dimensional factors, known as Latent Variables (LV), which capture both the greatest amount of variation in the explanatory variables (similar to PCR),but which also are relevant for predicting the dependant variables [40]. PLS has been demonstrated in several studies to be a good alternative to MLR. Research has shown that when the calibration data set is changed, the model parameters in a PLS model do not vary as much as when MLR is used [39], [41], suggesting that PLS is a more robust algorithm.

Traditionally the most popular method for calculating a PLS model is known as Nonlinear Iterative Partial Least Squares (NIPALS) [42]. This iterative method, first calculates one LV from the explanatory variables, then a residual matrix is obtained by subtracting the outer product of the vectors that compose the first LV from the explanatory variables, this residual matrix is then used to calculate the subsequent LV. More details of this algorithm and others related to PCA and PLS are provided in Chapter 4.

Analogous to NIPALS, the Simple Implementation of Modified Partial Least Squares (SIMPLS) [43] algorithm has been widely employed as a way to calculate a PLS model. SIMPLS obtains the vectors that compose the LV by deflating the covariance matrix of the explanatory and dependant variables instead of deflating the variables themselves (as NIPALS does). The resulting algorithm is not as intuitive as NIPALS but is considerably faster to implement [44].

Many authors have published other improvements and modifications to the PLS modelling technique, such as Multiway PLS(MPLS) [28], [36], nonlinear PLS modelling [45], [46], recursive algorithms to include new objects [47]–[50], clustering [51], [52], model validation [10], [53], [54], among others [27], [44], [55]–[63] .

PLS models have been widely used for batch process monitoring and control [64]. For example Nomikos and MacGregor [28] proposed a generic procedure for batch monitoring using a MPLS model using a similar methodology they employed for MPCA [26]. They extracted the information from the measured variable trajectories that were relevant to the final quality variables using MPLS over a historical database of successful past batches, in a simulation study of a styrene-butadiene batch reactor. Then, they used this MPLS model to create monitoring charts that could provide on-line predictions of the final product quality. The techniques were able to quickly detect faults in the score space that occurred during the progress of a batch in the simulation study.

Data-based models and MS models in particular have been shown to be a very effective approach when dealing with different types of batch processes. However, a major limitation with the approach is that the development of an accurate empirical model requires a high number of observations, which for many batch processes is simply not possible. In comparison, fundamental models do not require such a high number of observations. Consequently, data-based models do not provide enough information to satisfy all process design and analysis requirement and cannot replace fundamental models for all applications.

2.1.3 Hybrid approach

As discussed above, both knowledge and data based approaches have advantages and disadvantages on their own. To overcome these disadvantages, hybrid models that combine these two modelling approaches have been proposed. In hybrid models, knowledge based models are used to provide the information and analysis of the process variables interactions, whereas data based model are used to capture the unknown process characteristics, plant-model mismatches that result from the knowledge based model.

Several authors have successfully applied this methodology to batch processes, such as [3], [65]–[68]. For example, in [3], the authors proposed a practical hybrid approach that combined a knowledge based model to predict the final particle size with a linear regression model to predict the final product quality of a polymer, which they applied to an industrial emulsion polymerization process. The authors claimed that the knowledge based model reduced significantly the variability in the final product quality, while the regression model compensated for the unknown disturbances that appeared from one batch to the next.

The findings of this and other studies suggest that the combination of both methodologies in the hybrid approach offers advantages when solving engineering problems. However, it also requires the most engineering effort to build as the infrastructure, computing power and human knowledge of both approaches is needed.

To further illustrate the use of modelling in industrial batch processing, different control approaches utilised by several pharmaceutical companies are discussed in [69]. In this article, the authors presented an industrial survey that covered several aspects of process control. The results of that survey show that fundamental models are applied sparingly and only when their development is technically feasible, yet the pharmaceutical industry has widely adopted the use of

Process Analytical Technology (PAT) measurement systems, such as Near Infrared Spectroscopy (NIR), biosensors and Raman spectroscopy, combined with MS models. The survey results also show that the pharmaceutical industry has a very keen interest in the application of advanced process control approaches to reduce the variability observed during batch process operations.

The next section provides further examples of hybrid, data-based and knowledge-based modelling approaches that are used in batch control. On this occasion they are organized not by their modelling characteristics but by their problem formulation.

2.2 MONITORING AND CONTROL OF BATCH PROCESSES

Batch process management includes four layers of plant operation: planning, scheduling, monitoring, and control [5]. As discussed in the previous sections if a model of a batch process can be developed then it can be used to carry out planning and scheduling of plant operations. Furthermore, an accurate model of a batch process can also be used for monitoring and control applications.

One of the most commonly available measures of quality in industrial batch-processes is the end-point quality (also called final or end-product quality). End point quality is the value of a desired batch product, such as biomass or penicillin concentration, measured at the end of the batch.

As explained in the introduction, it is desirable to obtain reproducibility in the end-point quality from one batch to the next. However, batches are sensitive to variations in raw material properties and other disturbances. The purpose of monitoring is to detect abnormalities during process operation; therefore, having a reliable model from a batch process enables batch process operations to be monitored. Once such a model has been identified and its accuracy verified, then it is reasonable to proceed to the further implementation of this model within an optimal control system. Although, it must be recognised that in doing this considerably more effort is required in ensuring that the model remains accurate during control action.

Batch control has the problem of dealing with two types of variation: intra-batch (such as the deviation in concentration of a feed supply) and inter-batch disturbances (such as the variation in the initial conditions, resulting from changes in raw material for example). To solve this problem, a number of control strategies have been proposed. These control strategies fall in to two categories depending on whether the control is performed during the batch (referred to as within-batch control) or only at the start of the batch (referred to as batch-to-batch control).

2.2.1 Within-batch control (online)

This type of batch control obtains parameters measurements, performs calculations using these measurements and then applies control action during the batch. When using within-batch control in the presence of disturbances, the main task of the control system is to adjust the manipulated variables at single or multiple points during the batch, to minimise any undesired variation in the final output quality (end-point quality).

Many authors have proposed the use of knowledge-based models for within-batch control systems, such as [70]. This strategy uses knowledge from a process model to estimate certain batch parameters, such as the end-point output quality. Changes to the operation of the batch, such as an increase or decrease of a manipulated variable can then be applied to ensure that the end point matches the target.

An example of within-batch control using knowledge models was carried out by Kozub and MacGregor [71]. In their work they proposed an optimal feedback controller, which successfully used knowledge-based models to infer several copolymer properties from measurements taken from an industrial emulsion copolymerization reactor. However, many of these knowledge–based strategies are difficult to implement on industrial processes because they need a precise theoretical model for each application [70]. Data-based approaches overcome this limitation because they use information gathered iteratively and because data-based models are comparably easier to build and modify than knowledge-based models.

A widely used data-based approach that has been used extensively to regulate continuous processes and more recently to control the end-point quality of a batch is Model Predictive Control (MPC) [72]. The MPC algorithm collects the measurements available from the batch at each ‘decision’ or ‘control’ point and then uses this data and a model of the process to optimise the process over a finite time horizon. This series of actions is repeated until the end of the process to calculate all the optimum control moves through the entire duration of the process. MPC doesn’t specify a single, definitive control strategy but includes a range of methods that use model predictions to solve an objective function over a receding horizon [72]. Some examples of MPC techniques have been published in [7], [9], [73]–[76].

In within-batch control it is frequent to use MPC in conjunction with MS modelling techniques, such as MPCA and MPLS. MPCA and MPLS models can be derived by reducing the dimensionality of the variables involved in a process during past batch operation runs and then these models are used to solve on-line optimisation problems in future batches [77]. Flores-Cerrillo and MacGregor illustrated this method in two publications [6], [7]. In both articles the authors used dimensionally-reduced models in an innovative way to calculate the best Manipulated Variable Trajectory (MVT) of batch processes in a within-batch optimisation scheme. MPCA models were used in the within-batch control, proposed in Flores and Macgregor (2005) for tracking and correcting the trajectory of different explanatory variables in an attempt to maintain end-point quality. However, in [6], MPLS models were used to solve an optimisation problem which explicitly included the quality variables within the cost function.

Many studies have used the work of Flores-Cerrillo and MacGregor as a guideline for dealing with problems in process monitoring and control [64]. This is exemplified in the work published in [9], which aimed to control the end-product quality using MPLS in a similar way to Flores-Cerrillo and MacGregor. However, in their work they included soft and hard constraints and also extended the technique to reject disturbances. Another example where MPLS models were used for within-batch control is the study carried out in [78]. This study formulated an on-line monitoring and feedback control, which regulated the end-product quality of a crystallization process (crystal mean size). In this article the authors used image texture analysis (Fractal-wavelet) and PLS modelling to control the final product quality and claimed to have attained satisfactory predictions and control of the crystal mean size.

In all the approaches mentioned in this section, the control system was designed to operate within the batch. This type of control design does not exploit the iterative nature of batch processes and it also suffers from significant problems if measurements are not available for all variables throughout the batch. The next section discusses the development and application of batch-to-batch control systems, which analyse the behaviour of one batch and make changes to the subsequent batch in an attempt to improve product quality.

2.2.2 Batch-to-batch control (offline)

Batch-to-batch control (also called Run-to-run control) differs from within batch control in that it uses an offline control strategy between batches. This strategy consists of using the data collected from previous batches to adjust all or some of the MVTs for the next batch. This adjustment has the objective of bringing the end-point quality closer to the desired operation in the presence of disturbances and/or to optimise some economic cost function. The main task in batch–to-batch control is to complete the control objectives in the face of disturbances and potentially to changes in the dynamics of the process: If any disturbances are highly correlated from batch to batch, then the information from previous batches can be useful to determine how the next batch should operate to mitigate any similar disturbance. However, if the disturbances behaviour changes from batch to batch, within-batch control is a more sensible option [70].

Iterative Learning Control (ILC) is a widely used techniques in batch-to-batch control [79]–[84]. ILC deals with the control of repetitive actions by using the information in previous iterations to obtain the best inputs for the next iteration. These inputs are determined such that they minimise a function of the error [85]. ILC was first introduced by Arimoto et al. [86] as a specific technique in robotics which was based on the ability of humans to learn from repetitive tasks. The term ILC in batch-to-batch control, however, is usually used in different control methods that work in a repetitive mode and that have a control law based in the tracking error of previous measurements.

A classic example of ILC in batch-to-batch control is the study presented in [81]. In this study, the authors proposed an ILC strategy for a fed-batch fermentation process. The strategy used iteratively-updated linearized models to calculate the MVTs necessary to eliminate undesired variations and disturbances from one batch to the next. The main objective with this strategy was to enhance the end-point quality, thus, they proposed and solved an optimisation to minimize the tracking error.

Strategies that solve a cost function to optimise the operation from batch-to-batch, such as the example presented above [81], are usually referred as batch-to-batch optimisation. This term applies to those approaches that search for an unknown optimal operating MVT that will minimise a predefined cost function. In other words, batch-to-batch optimization is defined in [5] as '*the procedure to calculate an optimal trajectory in a feed forward sense after a model update which exploits the repetitive nature of batch processes'*.

Another batch-to-batch optimisation strategy was proposed in [68]. The authors of this article used a hybrid control scheme for a cobalt oxalate synthesis, which combined a first principle model and a MPLS model to predict the end-point particle size distribution. This control scheme also proposed an optimization of a cost function which considered the tracking error of the end-point quality. The resulting controller was claimed to have improved the performance of the production from one batch to the next in the presence of disturbances.

Turning now to the modelling techniques employed in batch-to-batch control, it has been observed that batch processes have a tendency in nature to exhibit strongly nonlinear behaviour, which means that the model identification using linear modelling can produce inaccurate or unreliable predictions [87]. Some authors have proposed the use of nonlinear modelling techniques to deal with this problem, such as in [84], [87], [88]. The work published in [84] describes the application of a MPLS Non-Linear Regression technique using a moving window strategy for modelling. The result presented in this publication showed that the ILC control strategy using PLS regression and a moving window exceeded those using a linear regression technique.

Some authors have claimed that nonlinear modelling techniques are often so time consuming and expensive that they have failed to impress the batch processes industries [70], and that it is better to apply a combination of alternative control techniques for such processes [89].For example, the batch-to-batch optimization presented in [70] included a quadratic term in the manipulated variables in the MPLS model to regulate nonlinear effects in the process. This study also included a combination of within-batch and batch-to-batch control strategy, which used a MPLS model and an objective function based on the quadratic error of the final quality to control the particle-size distribution in an emulsion polymerization.

Another batch-to-batch optimisation technique proposed in [90] was reported to have low-sensitivity to nonlinearities and uncontrolled variability. The authors of this publication developed a control methodology based on the gradient estimation for a batch-to-batch evolutionary optimization using MPLS models. Their results showed a significant increase in the end-point quality of a fermentation process compared with knowledge based approaches.

The work in this thesis formulates an alternative batch-to batch optimisation approach that uses a straightforward Quadratic Programming (QP) optimization. The objective of the QP formulation was to find an optimal MVT that minimised the predicted error in the end-point quality from batch to batch. This formulation also used a MPLS modelling technique similar to that previously formulated in [6], [9], [80]. Although these articles work in a within-batch control scheme, they also used PLS models that contained batch-to-batch information. This thesis formulates an adaptive batch-to-batch optimization that combines the MPLS modelling technique with an adaptive scheme.

2.3 BATCH-TO-BATCH OPTIMISATION EXTENSIONS

Section 2.2 described the relevant literature advances related to batch-to-batch control, the main topic of this thesis. However, this thesis also proposes the inclusion of validity constraints within the QP formulation and the addition of a smoothing technique within the PLS modelling. The next section describes some of the recent published advances in these areas.

2.3.1 Validity constraints

An important aspect of the research presented in this thesis is the accuracy of the end-point quality prediction. To ensure good accuracy when estimating end-point quality there is a need to validate the model and restrict it into a region described by previous batches. In other words, the model should not be allowed to extrapolate in to regions that it was not calibrated for. This key element of the optimization scheme has been investigated in several publications using different approaches. For example, Nomikos and MacGregor [28] proposed an expression for confidence intervals when using MPLS models, which are now commonly used in the monitoring of dynamical processes [91]. In their work, they claimed that it was reasonable enough to use a simple approximation based in the confidence intervals calculation to indicate how valid the model was.

Similar work was published in [92]. The authors of this article used Squared Prediction Error (SPE) charts and the confidence intervals proposed by [28], for online batch monitoring, quality prediction and fault diagnosis. Their scheme used the confidence intervals as a tool to detect faults and disturbances in the plant, with the SPE charts being used for online monitoring of batch evolution.

Another interesting modification to the confidence interval used by Nomikos and MacGregor [28] was proposed in [93]. This modification considered the period at which the prediction was made and the amount of information available to estimate the confidence intervals. Their proposed approach was shown to provide a more accurate prediction, with smaller confidence limits towards the end of the batch as more information was available than at the start of the batch and hence the accuracy of the model was improved. Their approach was validated using data from a real chemical reactor.

There have been other approaches in addition to the technique proposed by Nomikos and Macgregor for calculating confidence intervals in MPLS models, such as the work published in [94]. The authors of this publication modelled the operation of a pharmaceutical process using data obtained from NIR spectroscopy. They used a MPLS residual bootstrap calculation to estimate the confidence intervals in the prediction of the tablet potency in this pharmaceutical product. Although considerably more complicated than the Nomikos and MacGregor approach, the authors of this article claimed that their technique ensured that the value obtained by a classical chemical analysis was close to the model calculation.

A method for restricting the control movements to ensure that the process remained within a region covered by the MPLS models was proposed by Flores-Cerrillo and MacGregor [6]. In this work, the authors controlled the batch end-product quality of two different case studies by using the Hotelling's statistic within the cost function. This allowed the control system to restrict the optimization space to the region of the score space that was consistent with past batches. This restriction however, had a weighting factor that needed to be tuned from batch-to-batch, increasing the productions costs by requiring more sub-optimal batches.

A notable modification of this approach was published in [10]. This modification included the use of normalized restrictions in the cost function to avoid having to tune the weighting factor used by Flores-Cerrillo and Macgregor [6]. In addition, they added a quadratic error term in the residuals, usually employed in monitoring and faults diagnosis [95], to the cost function as part of the strategy to ensure the validity of their MPLS model in an innovative Latent Variable MPC strategy for continuous processes. The same research was later improved in [8] by using the validity restrictions, not as soft constraints but as hard constraints in the cost function, ensuring that the validity restrictions were not disregarded in their within-batch control technique.

This thesis proposes the inclusion of two model validation techniques within a batch-to-batch control system. These were included as validity constraints similarly to the work in [8], [63] and are integrated in to the cost function in the QP problem formulation to ensure the validity of the MPLS model is respected. The addition of these terms in the control strategy aims to improve its performance by increasing the robustness of the control methodology under different scenarios.

2.3.2 Smoothing techniques

In control system design, smoothing techniques are usually used prior to the identification stage to handle noisy data. Noise in the raw calibration data can create errors in the estimated calibration parameters and hence systematic errors in the prediction of the output [96]. Therefore, by smoothing process measurements, the signal-noise ratio of input data can in theory reduce the noise in the calibration data.

While many industrial applications use pre-processing filtering techniques such as moving average filters and Fourier analysis-based filters prior to identification [96], most of these techniques are disconnected from the modelling algorithm formulation. One approach that is used in this work as a method for introducing this concept in to PLS regression is known as Functional PLS (FPLS) [97]. This algorithm is based on functional analysis [98] and it applies PLS regression over a linear combination of smoothed functions that represent the output variables. Other publications [99], [100] have evaluated the performance of FPLS and found that the FPLS models provided better estimations than classical PLS models

Although FPLS considers the smoothing of the explanatory variables inside the modelling technique, the smoothing operation can be classified as a pre-processing technique as the PLS regression itself does not change. Another approach that has tackled the smoothing problem within the PLS regression was formulated in [101]. This smoothed-model regression technique is known as Penalized PLS (PPLS) and it includes a penalty term in to the NIPALS regression to smooth the

PLS loading coefficients, which according to the authors should, in theory lead to improved prediction performance.

For batch-to-batch optimisation, several publications have reported that the performance of the control system was improved when filtering techniques were applied prior and/or after the optimisation was performed [90], [102]. Camacho et al. [103] suggested two hypothesis for this phenomena. One hypothesis was that smoothing over a specific dataset could improve the PLS modelling performance, thus improving the accuracy of the model estimations. The second hypothesis was that smooth inputs in the minimization problem could reduce the search space in the optimisation sequence, hence improving the problem solution. According to their results using a number of simulations, they observed that the second case was more likely to be the correct hypothesis. They also evaluated PPLS, FPLS and the filtering approach employed in [90] over two batch process simulations and concluded that the filtering approach outperformed the other smoothing techniques .

Chapter 7 of this thesis presents a similar analysis to that published by Camacho et al. [103]. The objective of this analysis was to corroborate the findings of Camacho et al. regarding the possible hypothesis for the filtering problem. The results of this analysis are obtained comparing the filtering approach used in [102], the FPLS, and the PPLS algorithms over 2 batch-process simulations.

2.4 SUMMARY

This chapter has provided the relevant literature review for the application of batch-to-batch optimisation using MS models. The most relevant points presented in this chapter are listed next:

- Section 2.1 provides the main modelling techniques used in batch processes divided in three categories: knowledge-based, data-based or hybrid models. Examples of these strategies are also provided in this section.

- Section 2.2 presents the main techniques employed in monitoring and control of batch process. Some examples are provided divided in two categories: within-batch and batch-to-batch.

- Section 2.3 present some relevant advances published with regard to the subject matter of Chapters 6 and 7. These advances include the addition of smoothing techniques and validity constraints in batch-to-batch optimisation.

Chapter 3: Preliminary Methodology

This chapter outlines the design and overall structure of the experiments presented in this thesis. These experiments are designed to test the control systems that are developed and presented in this thesis. Section 3.1 begins by listing the hardware and software used to obtain the results presented in the following chapters. Then, Section 3.2 describes the case studies used to simulate batch production. Section 3.3 provides the data structure that was used for the experiments described in this thesis. Finally, Section 3.4 presents a brief summary of this chapter.

3.1 HARDWARE AND SOFTWARE

The hardware used for the computations consisted of:

DELL Optiplex 790

- Processor: Intel(R) Core(TM) i5-2400 CPU @ 3.10GHz
- RAM: 8.00 GB.
- System type: 64-bit.

The software used for the computations consisted of:

- Microsoft Windows 7 Enterprise 2009 Service Pack 1.
- MATLAB (The MathWorks, Inc.) 64 bit R2012a and R2013a.
- MATLAB Statistics and Machine Learning Toolbox
- MATLAB Optimization Toolbox

The PLS regression and the Principal Component Analysis (PCA) coefficients were obtained using, respectively, **plsregress** and **pca** functions from MATLAB. Similarly, the QP problems solutions were found using **quadprog** and **fmincon** functions from MATLAB. Finally, filtering to smooth the MVT was done using the function **filtfilt** and the coefficients were obtained using **fdatool**.

3.2 Case studies

One of the most common industrial sectors that uses batch processing is the pharmaceutical industry, which uses this approach in the production of high value added pharmaceutical and biochemical products. The particular type of batch process that is used in this industry is fermentation processing [69]. This type of process was therefore used as a simulation to evaluate the performance of the different control algorithms developed in this thesis. One of the simulations used in this work was a yeast production process, which consisted of simplistic models based on kinetic expressions that was published in [4]; the other simulation was a more complex penicillin production published in [20], which included built-in PID controllers.

The first simulation, named *Saccha* through this thesis, was programmed using a MATLAB ODE (Ordinary Differential Equations) solver based in the work described in [4]. The second simulation known as Pensim was provided by the Illinois Institute of Technology and modified to obtain the final penicillin concentration as the quality variable instead of the biomass concentration.

3.2.1 Saccharomyces cerevisiae production

The first case study that was used in this work was the simulation of *Saccharomyces cerevisiae (Saccha)*. This is one of the most useful species of yeast as it has been used for centuries in the production of food and alcoholic drinks. The Saccha simulation used in this thesis was presented in [4] and has been used as a benchmark simulation in several studies [11], [18], [84], [90], [103]. Its knowledge model includes 11 reactions and 9 mass balance dynamic equations. The simulation is able to operate in both batch and fed-batch modes, however, in this work the focus was on fed-batch.

The variables used in Saccha simulation are shown in Figure 3.1. The terms that follow the variables are the notation that will be used throughout this chapter for describing the measured variables ***v***, manipulated variables ***u*** and output variables ***y***. Similarly, the sub index of the input variables will be useful to describe the allocation of variables for identification, which will be described in the model building section in Chapter 4.

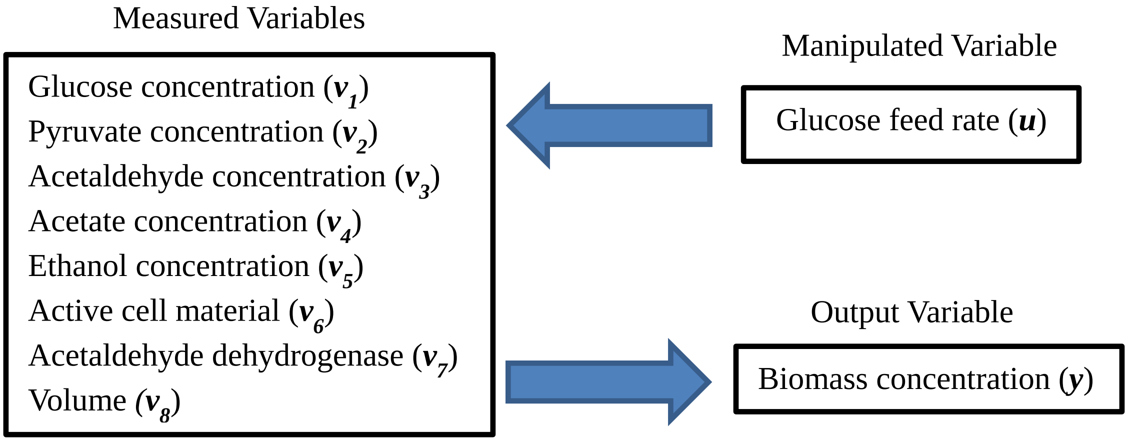

Measured Variables
Glucose concentration (v_1)
Pyruvate concentration (v_2)
Acetaldehyde concentration (v_3)
Acetate concentration (v_4)
Ethanol concentration (v_5)
Active cell material (v_6)
Acetaldehyde dehydrogenase (v_7)
Volume (v_8)
Manipulated Variable
Glucose feed rate (u)
Output Variable
Biomass concentration (y)

Figure 3.1: Variables in the Saccha simulation.

Table 3.1 lists the initial conditions of the input and output variables shown in Figure 3.1. These initial conditions are the same as those used in [18], [4] and [90] .

The objective when operating the Saccha simulation is to maximize the output quality (biomass concentration) by manipulating a highly concentrated glucose feed (100g/l) throughout a 10-hour batch process. The sample time for the simulation was 0.1 hours. In addition, 1% Gaussian noise was added to the end-point quality and 5% to the initial conditions to investigate the behaviour of the control systems to sensor noise and unmeasured disturbances respectively.

Table 3.1: Saccha initial conditions.

Variable	Initial condition
Glucose concentration	0 (g/l)
Pyruvate concentration	0 (g/l)
Acetaldehyde concentration	0 (g/l)
Acetate concentration	0 (g/l)
Ethanol concentration	0 (g/l)
Biomass concentration	1 (g/l)
Active cell material	0.3
Acetaldehyde dehydrogenase	0.0075
Volume	7 (l)

Figure 3.2 shows the trajectories of the process variables in the Saccha simulation when a feeding law similar to the *optimal* MVT that was used in [90] was applied.

More detailed information regarding the Saccha parameters is available in [4] and [90] .

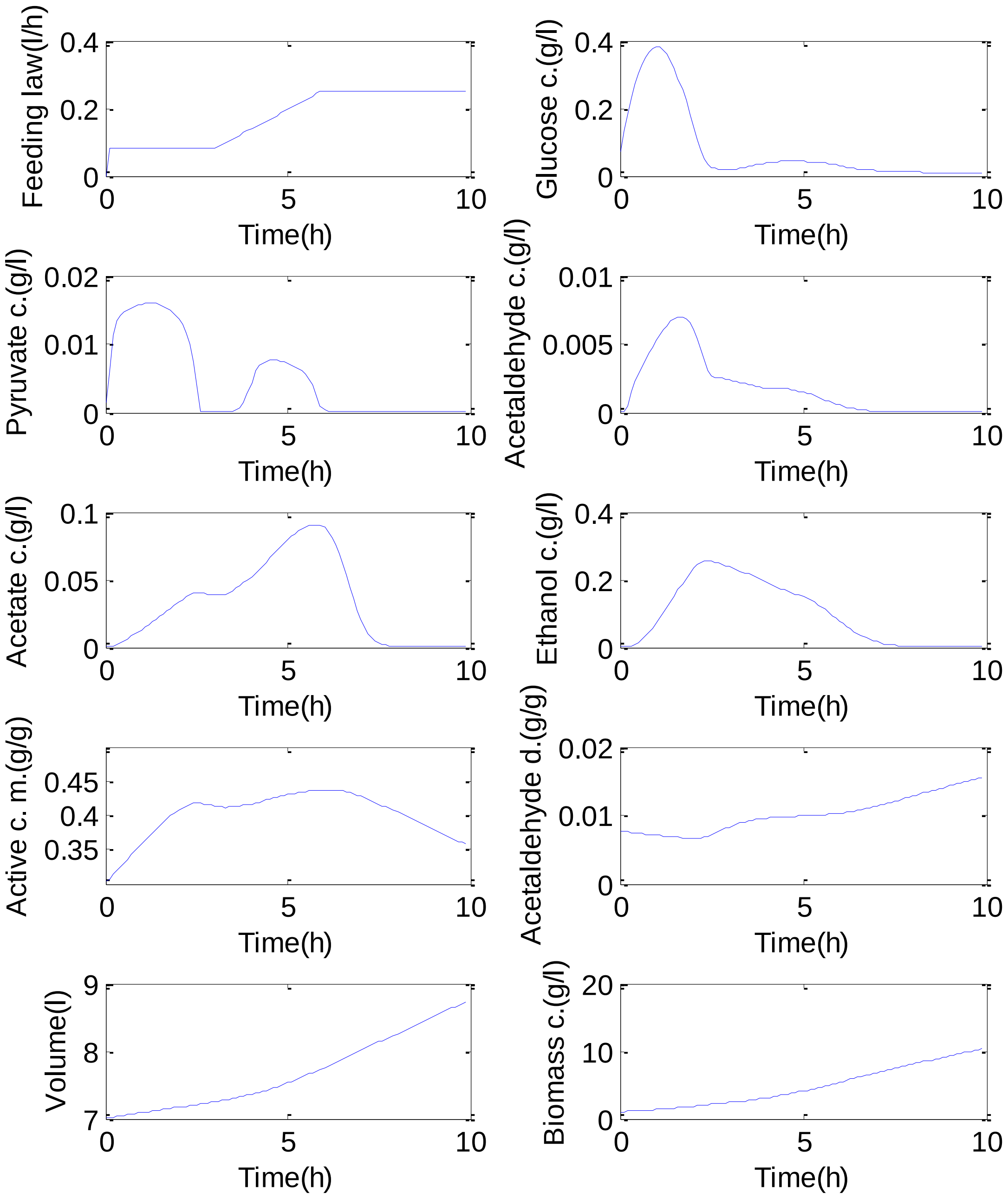


Figure 3.2: Variables trajectories in a Saccha simulation.

3.2.2 Penicillin fed-batch fermentation production

The second case study was a benchmark simulation developed for penicillin production (*Pensim)* that was presented in [20]. A major reason for selecting this simulation was that penicillin is one of the most useful groups of antibiotics used in modern medicine. Furthermore, the *Pensim* simulation has been used as a benchmarking tool in a range of publications describing new batch monitoring and control designs [9], [104]–[108].

The knowledge model of the *Pensim* simulation is an extension of the work of Bajpai and Reuß [21] and considers a process with 5 input variables, 8 measured variables, 1 manipulated variable, and 2 output variables (from which one was selected to be controlled). These variables are shown in Figure 3.3 with their respective notation.

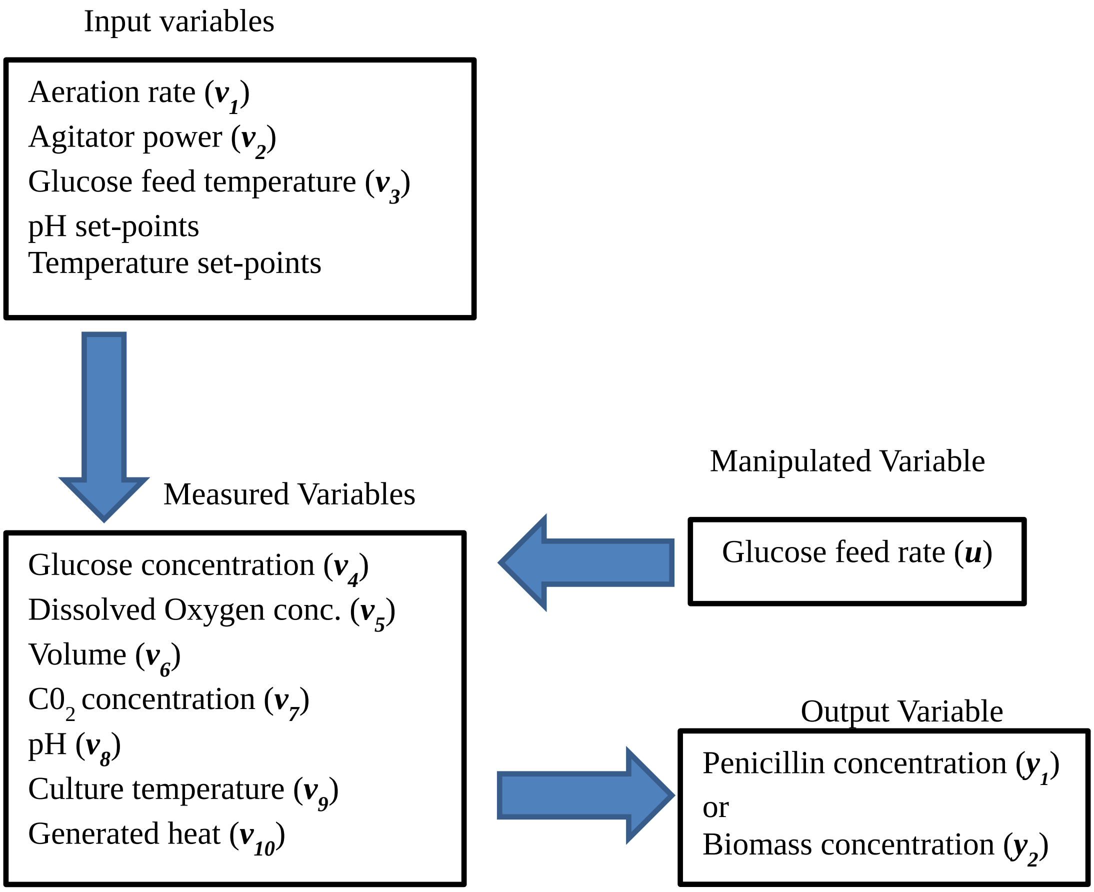

Input variables
Aeration rate ($\boldsymbol{v}_1$)
Agitator power ($\boldsymbol{v}_2$)
Glucose feed temperature ($\boldsymbol{v}_3$)
pH set-points
Temperature set-points
Manipulated Variable
Measured Variables
Glucose feed rate ($\boldsymbol{u}$)
Glucose concentration ($\boldsymbol{v}_4$)
Dissolved Oxygen conc. ($\boldsymbol{v}_5$)
Volume ($\boldsymbol{v}_6$)
$C0_2$ concentration ($\boldsymbol{v}_7$)
pH ($\boldsymbol{v}_8$)
Culture temperature ($\boldsymbol{v}_9$)
Generated heat ($\boldsymbol{v}_{10}$)
Output Variable
Penicillin concentration ($\boldsymbol{y}_1$)
or
Biomass concentration ($\boldsymbol{y}_2$)

Figure 3.3: Variables in the Pensim simulation.

This simulation includes a PID controller that manipulated the acid/base feed-rate to regulate the pH, and a second PID controller that manipulated the heating/cooling water flow rate to regulate the temperature of the culture. The set-points of these controllers, along with nominal values of the input and manipulated variables were excited with filtered Pseudo Random Binary Sequence (PRBS) for model identification.

The primary objective when operating this case study was to maximize either the final penicillin concentration or the final biomass concentration by manipulating the MVT of the glucose feed rate in the least number of batches. Each batch in the simulation consisted of 200 hours divided in to 200 discrete intervals of 1 hour. For the first 45 hours, the process operated in batch mode and after this time, operation was switched to fed-batch. In the control studies described in this thesis, the process was only manipulated after this 45 hour point. The simulation therefore consisted of two stages: a biomass growth-phase with pre-set values and a penicillin production phase which required glucose feed [20], which could be manipulated to maximise penicillin or biomass production.

Table 3.2 shows the initial conditions of the Pensim variables with the standard configuration described in [9].

The quality measure that was used for this case study throughout the work described in this thesis was chosen to be final penicillin concentration. Although several studies have tried to maximise biomass concentration, the desire in any manufacturing process will be to maximise the production of penicillin. Penicillin is much more complicated to control than biomass because it’s relationship to glucose addition is highly non-linear, whereas biomass has a linear relationship with glucose feed [107]. Applying control and optimisation techniques to penicillin production therefore provides a much more realistic evaluation of the capabilities of any developed technique. In addition, Gaussian noise with 1% amplitude was added to the end-point quality to investigate the behaviour of the control systems to sensor noise as in the Saccha simulation.

Table 3.2 Pensim initial conditions.

Variable	Initial condition
Aeration rate	8 (h^{-1})
Agitator power	30 (W)
Glucose feed rate	0.045 (l/h)
Glucose feed temperature	296 °K
pH set-points	0 (ml/h)
Temperature set-points	298 °K
Volume	100 l
Culture temperature	297 °K
Generated heat	0 cal
pH	5
Glucose concentration	15 (g/l)
Biomass concentration	0.1 (g/l)
Penicillin concentration	0 (g/l)
Dissolved oxygen concentration	1.16 (g/l)
CO_2 concentration	0.5 (mmol/l)

Figure 4 shows the trajectories of the major variables within the pensim simulation when a feeding law, similar to the *optimal* MVT obtained in [104] was applied.

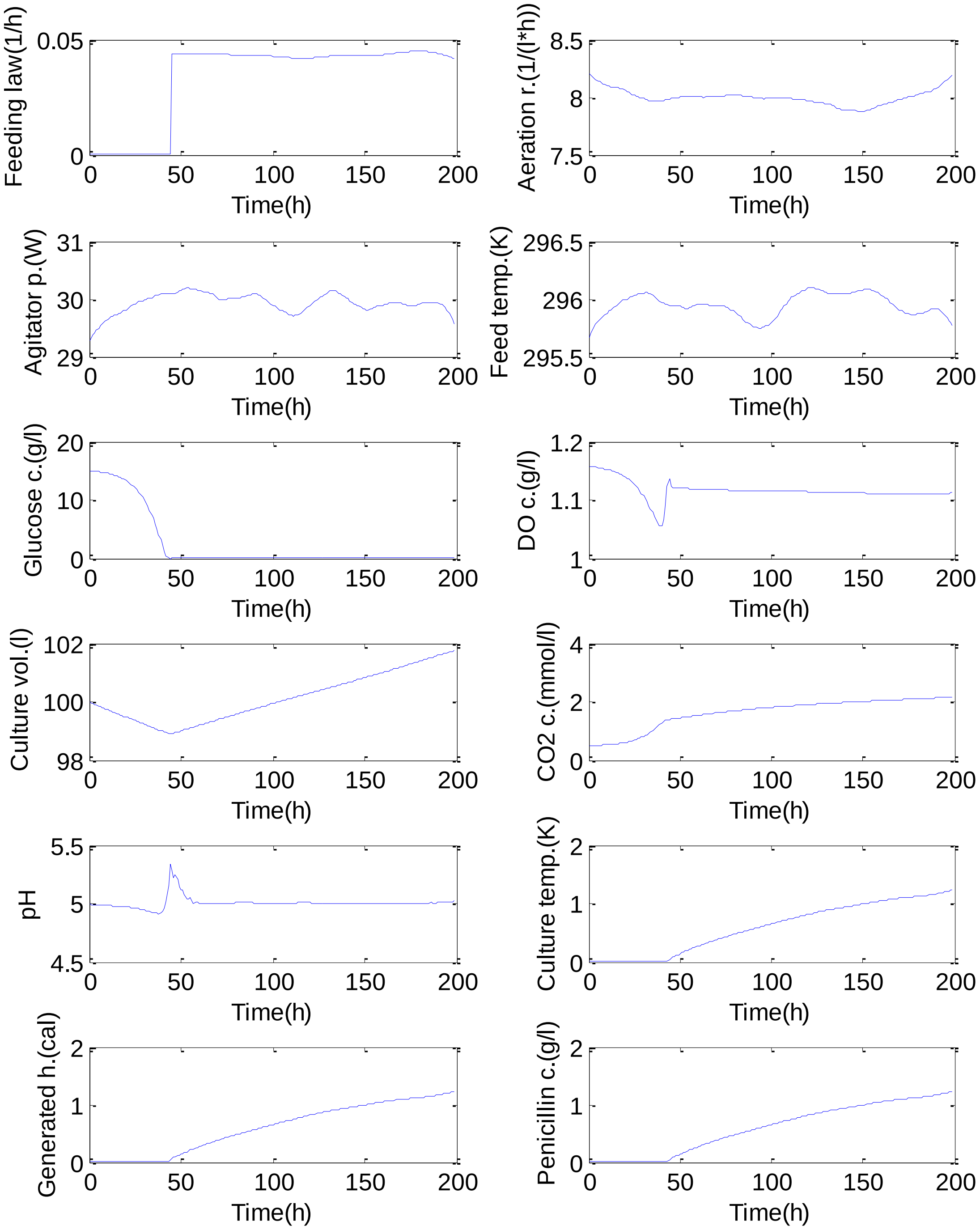


Figure 3.4: Variables trajectories in a Pensim simulation.

3.3 DATA STRUCTURE

In batch-to-batch operation, the measurements from consecutive batches are obtained over a two dimensional space: time within batch and batch number. Within-batch-time measurements are obtained similar to that of periodic operation, by dividing the operation time into K discrete time data-points. The number of batches is also by definition discrete so the variables measurements are regularly distributed into batch number data-points I. This means that the data of each variable for each experiment in this thesis, from batch-to-batch operation had two time dimensions $(I \times K)$.

Figure 3.5 illustrates the data structure for one process variable over the 2-dimensional space $(I \times K)$. This figure shows sample-data (100 samples for 10 hours) from a 100 batches of the Saccha simulator.

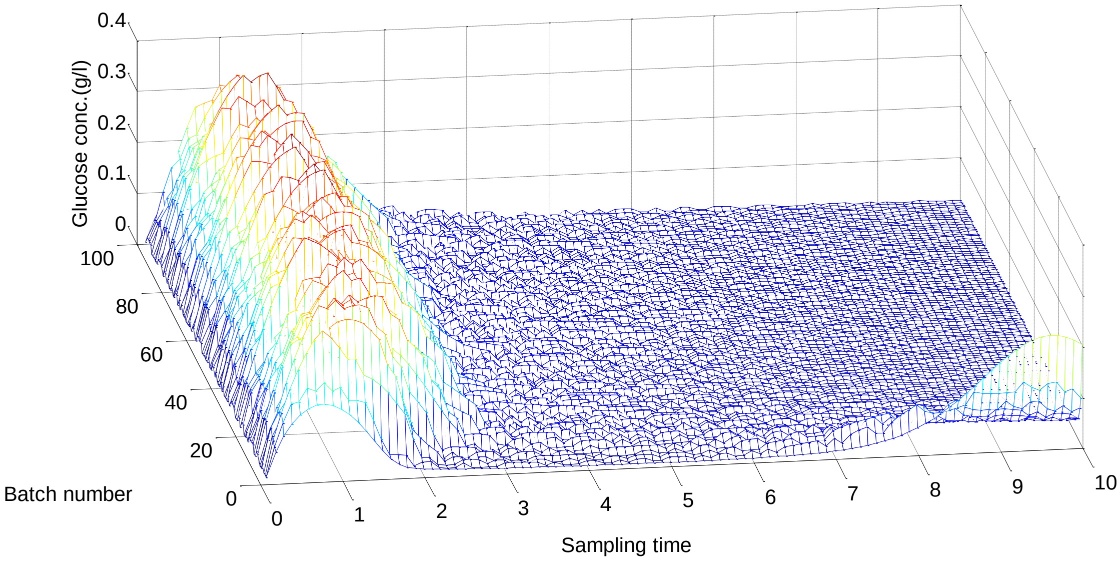


Figure 3.5: Glucose concentration measurements (100 over 10 hours) from a batch-to-batch optimisation of Saccha over 100 batches.

Simulation data such as that observed in Figure 5 was stored in a predictor matrix $\boldsymbol{X}$ along with the other variables. This matrix was then used as an input for the PLS modelling algorithm. This resulted in $\boldsymbol{X}$ having an extra dimension J for the different process variables. As a result of this, the predictor matrix $\boldsymbol{X}$ was defined as a 3-dimensional space $(I \times J \times K)$. This matrix was then transformed using an unfolding technique within the MPLS algorithm to have two dimensions $(I \times JK)$. A more detailed explanation of this process is provided in Chapter 4.

The data structure for the response variables was obtained from the end-point quality of each batch, resulting in a response vector $\boldsymbol{y}$ (I). However, if multi-output optimisation were to be used in future work (i.e. penicillin and biomass concentration), then the response matrix $\boldsymbol{Y}$ would become 2-dimensional *(number of quality variables ×J).*

3.4 Summary

This chapter described the experiment design and case studies for the experiments presented in this thesis. The chapter was divided as follows:

- Section 3.1 describes the hardware and software used to obtain the calculations.

- Section 3.2 described two fermentation-based case studies for the experiments: a yeast production named Saccha and penicillin production known as Pensim.

- Section 3.3 provided the data structure use as input and outputs in the numerical experiments.

Chapter 4: Optimisation and Modelling of Batch Processes

This chapter presents the main theoretical framework used for the batch-to-batch optimisation proposed in this thesis. Firstly, Section 4.1 states the type of methodologies used to solve the control problem in batch-to-batch optimisation. Secondly, Section 4.2 describes the optimisation theory used to calculate the optimal MVT. Finally, Section 4.3 provides a brief explanation of the LV statistical modelling techniques, necessary for the MVT optimisation proposed in Chapter 5.

4.1 INTRODUCTION

As was mentioned in the literature review in Chapter 2, it is clear that the main objective of **batch-to-batch optimisation** (also referred to as run-to-run optimisation or simply batch optimisation) is to fully exploit the repetitive nature of batch processes to find the optimal operating conditions of a batch under the effect of disturbances in the fewest number of batch runs. This is achieved by using the measurements of past batch operations to improve subsequent runs.

To achieve this objective, Srinivasan et al. [109] suggest three types of methodologies:

1. "Model-free evolutionary optimisation": Instead of using a model, the process performance, based on the trajectory of the input or manipulated variables, is evaluated from experiment to experiment.

The process inputs are re-calculated for each batch by calculating the gradient and running an optimisation algorithm.

2. "Evolutionary optimisation with model-based gradient": This type of optimisation uses a dynamic model instead of the process itself to calculate the gradient and find optimal inputs.

3. "Optimisation via model refinement": A process model is 'refined' using the information from previous experiments. Then, this model is used to solve an optimisation problem.

Each of these three methodologies has advantages and disadvantages over the others. The advantage of the first methodology is that, unlike the second methodology, it does not require an accurate model to successfully optimise. The disadvantage is that it requires numerous batch runs before finding the gradient and, thus, returning an optimal response. In light of this, the third methodology seems a natural choice for the proposed design as it improves the accuracy of the model without incurring the costs resulting from numerous *sub-optimal* batch runs.

Srinivasan et al. [109] also describe a common procedure for optimisation via model refinement, where θ represents the parameters of the model and π represents the decision variables:

1. "Choose initial guesses for parameters θ."
2. "Use the model and an optimisation algorithm to obtain the optimal π."
3. "Run the batch with the optimal π."

4. "Use an identification algorithm and all the variable measurements to obtain a new estimate for θ."
5. "Repeat steps 2-4 until convergence."

These steps serve as the basis for developing the batch-to-batch optimisation procedure proposed in this thesis. Following the steps clearly necessitates two basic components:

- An **optimisation algorithm** to obtain the optimal MVT defined for each case study, from one batch to the next. The type of algorithm used for this purpose is defined in Section 4.2.

- A **process model** to estimate the dynamic behaviour of the process. The type of modelling technique is provided in Section 4.3

4.2 OPTIMISATION

In the mathematical sense, optimisation deals with problems of minimising or maximising a function that is usually subject to constraints. Applying this description to engineering problems gives rise to the objective of finding the values of the variables that yield the best value of a performance criterion in a process. This criterion, which uses a process model is referred to as the *optimisation problem* [110].

The general notation of an optimisation problem can be described as in Equation 4.1.

$$\min_{\boldsymbol{\pi}} f(\boldsymbol{\pi})$$

$$\text{Subject to: } \boldsymbol{h}(\boldsymbol{\pi}) = 0 \tag{4.1}$$

$$\boldsymbol{g}(\boldsymbol{\pi}) \geq 0$$

where $f(\boldsymbol{\pi})$ is the objective function to be optimised, also known as *cost function*, $\boldsymbol{h}(\boldsymbol{\pi}) = 0$ is an equality constraint and $\boldsymbol{g}(\boldsymbol{\pi}) \geq 0$ an inequality constraint.

In optimisation problems over a real plant operation, the minimisation depicted in Equation 4.1 has many solutions that satisfy the quality and inequality constraints. Hence, the optimisation algorithm selects the 'best' solution among the entire set of solutions. The 'best' solution is called the 'optimal solution' whereas all possible solutions in the set are known as 'feasible solutions'.

Many formulations have been proposed to solve optimisation problems depending mainly on the characteristics of the process model and its constraints. Some examples of such are as linear programming, nonlinear programming, robust programming, and integer programing, etc. This thesis proposes the use of Quadratic Programming (QP) as the optimisation algorithm to obtain the optimal MVT. This is because such a formulation would be relatively simple to formulate and solve. Additionally, if the QP is convex, then any local solution could be considered the optimal solution for the problem.

4.2.1 Quadratic Programming (QP)

In this optimisation problem, the objective function to be minimised is quadratic and is subject to linear equality or inequality constraints. The general QP problem formulation is shown in Equation 4.2.

$$\min_{\boldsymbol{\pi}} \boldsymbol{c}^{\boldsymbol{T}}\boldsymbol{\pi} + \frac{1}{2}\boldsymbol{\pi}^{\boldsymbol{T}}\mathbf{M}\boldsymbol{\pi} \qquad (4.2)$$

$$\text{Subject to: } \boldsymbol{A}\boldsymbol{\pi} \leq \boldsymbol{b}$$

where $\boldsymbol{c}$ is a (n) vector of constant coefficients, $\boldsymbol{A}$ is an $(n\times m)$ matrix, $\boldsymbol{b}$ is a (m) vector and $\mathbf{M}$ is a $(n\times n)$ symmetric matrix. This symmetric matrix is usually defined as $\boldsymbol{Q}$, however, this thesis reserves the letter $\boldsymbol{Q}$ for the loadings of the PLS model.

An important characteristic of Equation 4.2 is that if the $\boldsymbol{M}$ matrix is positive-semidefinite, then the QP objective function is convex. This means that the local solution of the minimisation problem would also be an optimal solution, hence a 'global solution' [111].

This section has defined the optimisation algorithm used to calculate the optimal MVT in the main batch-to-batch optimisation methodology. This section also clarified that this methodology is classified as an optimisation via model refinement, which therefore requires a suitable modelling technique. Such a technique will be the focus of the next section.

4.3 Latent-variables models

The design of the control scheme proposed in this chapter utilises some well-known theoretical concepts in MS modelling and the optimisation of batch processes. As previously mentioned in Chapter 2, MS modelling methodologies such as the PCA or PLS algorithms are extremely useful in batch control. This section describes these and other algorithms that naturally derive from the PLS regression.

For example, this section considers MPLS, which explains how to deal with 3-dimensional data; and adaptive PLS, which is used to update PLS models from one batch to the next. This section also describes another concept that is related to the mathematical optimisation used in this methodology: QP algorithms for the solution of constrained problems.

4.3.1 System identification

Chapter 2 described the importance of MS statistical methods in modelling and optimization of batch processes. In brief, MS compression techniques (or lower dimensional modelling techniques) are especially useful in collinearity situations. This is due to the redundancy necessary between predictor variables to ensure that MS models do not leave out important information. Therefore, it is inconvenient to use identification methods such as MLR that assumes that each variable in $\boldsymbol{X}$ has unique information about the responses variables in $\boldsymbol{Y}$. In other words, to find the relationship between the variables in $\boldsymbol{X}$ and $\boldsymbol{Y}$ (calibration), some rank reduction is necessary if collinearity exists among the predictors variables in $\boldsymbol{X}$.

In the case of batch processes, it is common for collinearity to exist between the variables in $\boldsymbol{X}$ as many of the mathematical relationships between variables are dependant among themselves. For example, Figure 4.1 shows the Pearson's linear correlation coefficient that one sample demonstrates relative to the other samples in $\boldsymbol{X}$ in a Saccha simulation process. If this coefficient is close to 1 or -1, the variables have a positive or negative correlation, respectively. On the other hand, if the coefficient is close to zero, then the variables are not correlated.

The main idea of rank-reduction methods is that the relevant information in the variables in $\boldsymbol{X}$ is condensed into A number of 'latent' factors. These factors are then used as a regressor to find a model. To illustrate this, Equations 4.3 and 4.4 formulate the compressed-data linear regression model; this is common practice to approximate the relationships of the data [96].

$$\boldsymbol{T} = \boldsymbol{X}\boldsymbol{V} \tag{4.3}$$

$$\boldsymbol{Y} = \boldsymbol{T}\boldsymbol{Q}^{\boldsymbol{T}} + \boldsymbol{F} \tag{4.4}$$

where $\boldsymbol{T}$ comprises the scores of the ‘latent’ factors, $\boldsymbol{V}$ is a matrix of weights necessary to find $\boldsymbol{T}$, $\boldsymbol{Q}$ is a matrix of loadings necessary to regress $\boldsymbol{T}$ into $\boldsymbol{Y}$; and $\boldsymbol{F}$ represents those contributions to $\boldsymbol{Y}$ which cannot be explained by the ’latent’ factors.

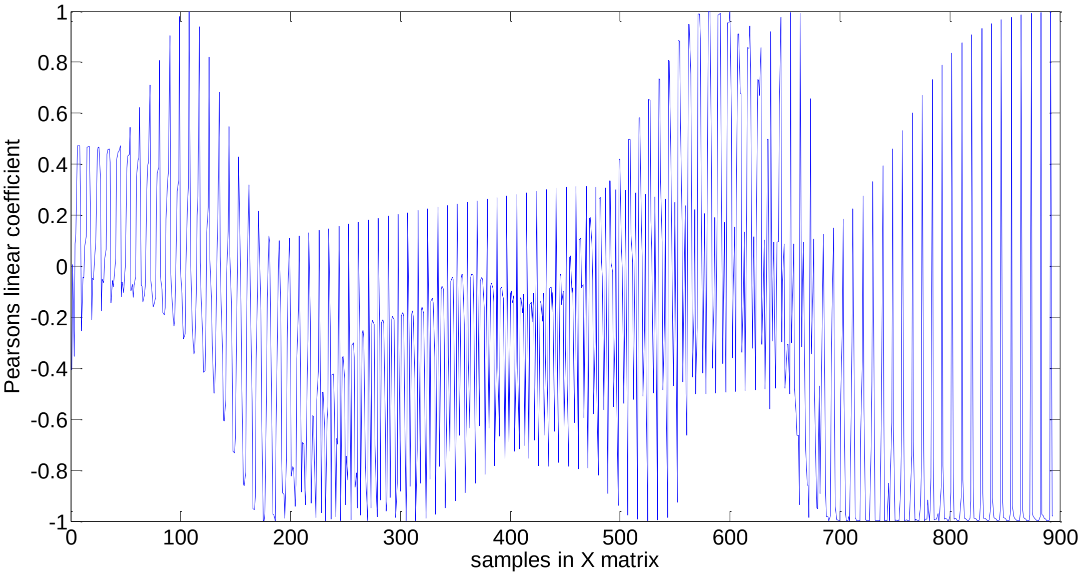


Figure 4.1: Pearson’s linear correlation coefficient for the predictor matrix in a Saccha simulation process.

It should be noted that in this model, for each variable to have the same importance than the others, the information in $\boldsymbol{X}$ and $\boldsymbol{Y}$ should be ’normalized’ by dividing each column by subtracting the mean and dividing by its standard deviation. This pre-processing adjustment is known as auto-scaling and normalising and it is illustrated in Equations 4.5 and 4.6.

$$x_{ijk} = \left(\frac{x_{ijk} \dots - \bar{x}_{jk}}{std_{jk} \dots}\right) \tag{4.5}$$

$$y_{ijk} = \left(\frac{y_{ijk} \dots - \bar{y}_{jk}}{std_{jk} \dots}\right) \tag{4.6}$$

The effect of mean-centring can be observed in Figure 4.2. This figure shows the data contained in one vector (or batch) of the ***X(I×JK)*** matrix before and after auto-scaling, for a Saccha data-set of 3 batches.

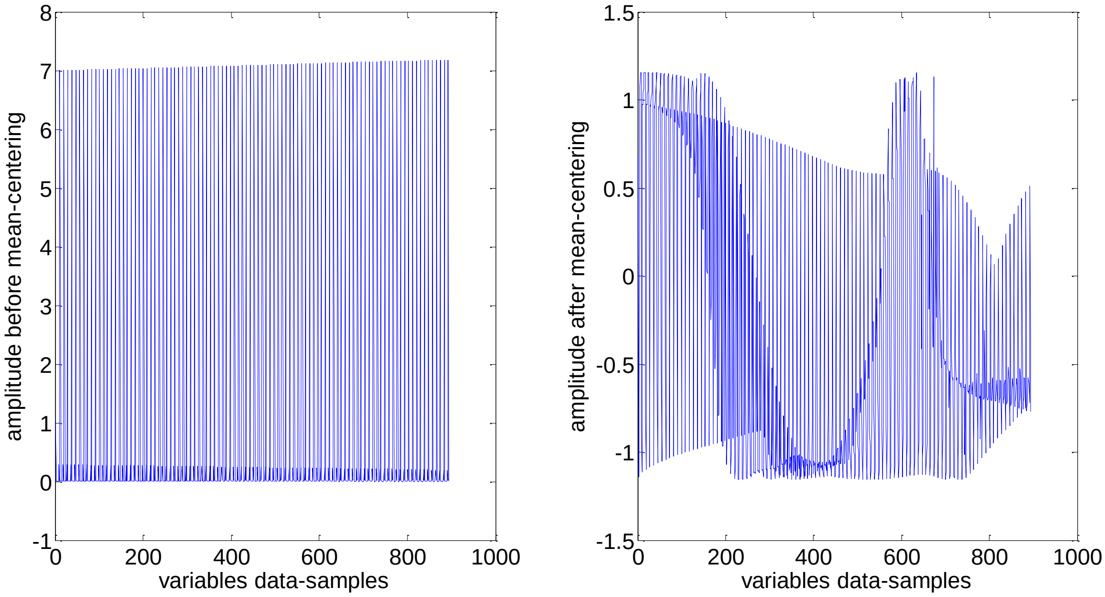


Figure 4.2: Effect of auto-scaling of 1 batch over a data-set of 3 batches of a Saccha simulation

Returning to the subject of data-compressed identification techniques, Martens and Naes [96] divided several compression methods into four areas:

- "Stepwise multiple linear regression": each vector of $\boldsymbol{V}$ consists of all zeroes except for one selected element with the value 1 for $\boldsymbol{X}$

variables. Various strategies are used to select which ***X***-variables to include'

- "Hruschka regression": Each vector of ***V*** represents the unique information in the spectrum of a certain object (a row in ***X***)'

- "Fourier regression": Each vector of ***V*** consists of smooth, mathematically pre-defined spectra consisting of low and medium frequency sine and/or cosine functions'

- "Bilinear modelling": each vector of ***V*** consists of coefficients estimated from the calibration data themselves according to some optimization criterion'.

However, this thesis only addresses bilinear modelling, which is a method that does not require substantial prior knowledge about the causal relationships among the variables in ***X*** and ***Y*** (which is generally true when building a black-box model) as it estimates the matrix of weights ***V***from the identification data-set.

4.3.2 Bilinear Modelling

In bilinear modelling, the coefficients of the weights matrix ***V*** are estimated according to an optimisation function, such as an optimisation based in the least squares criterion. The term 'bilinear' is given to this type of method because the resulting model approximates ***X*** as the product of two sets of linear parameters to be estimated: scores ***T*** and loadings ***P*** are illustrated by Equations 4.3 and 4.7. These equations, in addition to Equation 4.8, describe a full bilinear model.

$$\boldsymbol{X} = \boldsymbol{TP^T} + \boldsymbol{E} \tag{4.7}$$

$$\boldsymbol{Y} = \boldsymbol{TQ^T} + \boldsymbol{F} \tag{4.8}$$

where $\boldsymbol{P}$ is a matrix of loadings necessary to regress $\boldsymbol{T}$ into $\boldsymbol{X}$; and $\boldsymbol{E}$ represents those contributions to $\boldsymbol{X}$ which cannot be explained by the 'latent' factors. Matrices $\boldsymbol{E}$ and $\boldsymbol{F}$ are usually called residuals and they can be explained by the presence of noise or nonlinearities in the process.

The general methodology employed to model a process using bilinear regression techniques is divided into four steps [96] :

1. The estimated matrix of weights $\boldsymbol{V}$ is obtained from the identification data-set using an optimisation function.
2. The matrix of scores $\boldsymbol{T}$ are calculated using $\boldsymbol{V}$ as in Equation 4.3.
3. The matrices of loadings $\boldsymbol{P}$ and $\boldsymbol{Q}$ are calculated by the linear regression of $\boldsymbol{X}$ and $\boldsymbol{Y}$ on $\boldsymbol{T}$, as can be observed in Equations 4.9 and 4.10.

$$\boldsymbol{P'} = (\boldsymbol{T'T})^{-1}\boldsymbol{TX} \tag{4.9}$$

$$\boldsymbol{Q'} = (\boldsymbol{T'T})^{-1}\boldsymbol{TY} \tag{4.10}$$

4. The matrices of residuals $\boldsymbol{E}$ and $\boldsymbol{F}$ are obtained using Equations 4.11 and 4.12.

$$\boldsymbol{E} = \boldsymbol{X} - \boldsymbol{TP^T} \tag{4.11}$$

$$\boldsymbol{F} = \boldsymbol{Y} - \boldsymbol{TQ^T} \tag{4.12}$$

Once the model has been calculated, it is possible to predict a quality variable $\boldsymbol{\hat{y}_i}$ (such as the end-point quality of a batch) from new measurements $\boldsymbol{x_i}$ (such as measurements from a new batch) by using an estimation of the new scores $\boldsymbol{\hat{t}_i}$, as shown in Equations 4.13 and 4.14.

$$\boldsymbol{\hat{t}_i} = \boldsymbol{x_i V} \tag{4.13}$$

$$\boldsymbol{\hat{y}_i} = \boldsymbol{\hat{t}_i Q^T} \tag{4.14}$$

Bilinear models, such as PCR and PLS models, can be described using the notation defined by Equations 4.7 to 4.12. The difference between the models is each model’s conceptual objectives for its optimisation criterion. For instance, PCR attempts to find the factors in ***T*** that best describe the variation in ***X***. It then obtains the linear predictor model by regressing ***Y*** on ***T***. In contrast, PLS uses the variables in ***Y*** actively in the optimisation criterion to find ***T***. This difference causes the PLS model to reduce the impact of irrelevant variations in ***X***; however, it is more complex than PCR as it requires either two sets of loading vectors for the ***X*** matrix, or that the vector scores in ***T*** are inter-correlated [112].

That PLS models can reduce the impact of irrelevant factors in the prediction of ***Y*** is of great use where performance in the prediction of end-point quality is the focus. For this reason, PLS regression is employed in this thesis as the identification algorithm for the batch-to-batch optimisation methodology.

4.3.3 Partial Least Squares Regression

PLS regression is a Latent Variable (LV) regression technique which finds both the maximum variance in the involved variables, and the correlation between the quality variables $\boldsymbol{Y}$and the predictor variables $\boldsymbol{X}$. The name originates from work originally developed as a sequence of simple, partial models fitted by least squares [39].

In its original form, the PLS regression method was applied for one single $\boldsymbol{y}$ variable and was developed simultaneously (orthogonalised PLS and non-orthogonalised PLS) by Wold [39] and Martens [96]. Wold's approach attempted to find orthogonal scores $\boldsymbol{T}$ by using an additional set of orthogonal loadings weights $\boldsymbol{W}$. On the other hand, Martens' algorithm had fewer parameters, but the scores $\boldsymbol{T}$ were not orthogonal. Over time, the algorithm proposed by Wold, known as PLS-NIPALS [36], has become the standard method to express the PLS regression. However, Jong [43] introduced a faster version of the algorithm, known as SIMPLS, which is now commonly used when computing speed is a relevant factor.

The mathematical model of the PLS regression can be described by Equations 4.7, 4.8, 4.11 and 4.12 of the Bilinear Model and the prediction of the end-point quality as in Equations 4.13 and 4.14. The matrix of orthogonal weights $\boldsymbol{W}$ from the NIPALS algorithm can be defined in terms of the weights matrix $\boldsymbol{V}$from the bilinear model as shown in Equation 4.15.

$$\boldsymbol{V} = \boldsymbol{W}(\boldsymbol{P}^{\boldsymbol{T}}\boldsymbol{W})^{-\boldsymbol{1}} \tag{4.15}$$

For multivariate $\boldsymbol{Y}$ Equation 8 can also be described as in Equations 4.16 and 4.17.

$$\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{Q}^{\boldsymbol{T}} + \boldsymbol{G} \tag{4.16}$$

$$\boldsymbol{U} = \boldsymbol{T}\boldsymbol{B} \tag{4.17}$$

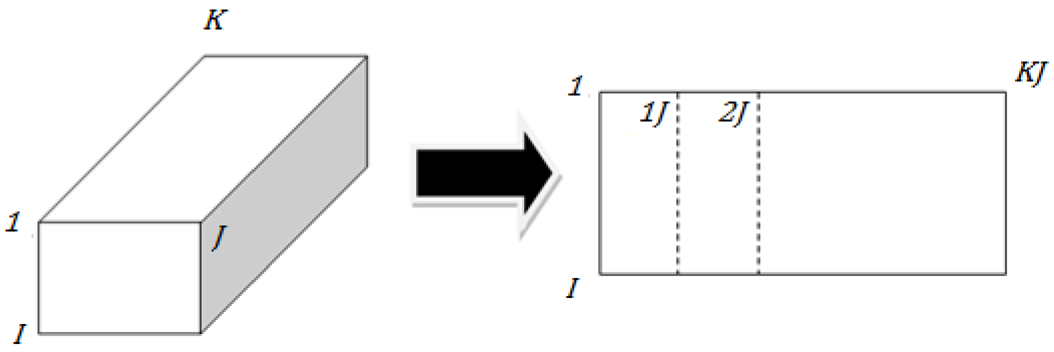
where $\boldsymbol{U}$ are the scores for the qualities in $\boldsymbol{Y}$ and $\boldsymbol{B}$ is the inner relationship between the scores $\boldsymbol{T}$ and $\boldsymbol{U}$. Although this thesis only tackles the problem of univariate quality more information about notation can be found in [112].

4.3.4 Multiway PLS (MPLS)

A complication introduced with batch processes is that the measurements are typically stored in a 3-dimensional data array, with dimensions corresponding to measured variables (*J*), time intervals (*K*) and batch number (*I*). Before PLS can be applied to such data it is necessary to transform the data into a 2-dimensional matrix. There are two main approaches used in MPCA and MPLS to achieve this *unfolding* according to [113]: The first approach is known as batch-wise. In this approach the variables and time are unfolded for each-batch. This allows the variability among the batches to be observed by analysing the measured variables and their time variation. In contrast the second approach, known as variable-wise, unfolds the batches and time for each variable and can be used to obtain information on the variability among the variables.

From the previous paragraph, it is clear that batch-to-batch optimisation is interested in batch-wise unfolding. In [28], [36] a batch-wise unfolding technique referred to as MPLS, was proposed and is now used routinely for modelling batch processes. The result of the MPLS transformation is shown in Equation 4.18 and Figure 4.3.

$$\boldsymbol{X} \in \mathbb{R}^{I \times J \times K} \rightarrow \boldsymbol{X} \in \mathbb{R}^{I \times JK} \tag{4.18}$$



Where I = Batch number
J = Number of variables
K = Time intervals

Figure 4.3: MPLS unfolding of 3D data into 2D data sets

This method, presents the variability of each process variable J about the mean trajectory at each time interval K along the batches I. This means that the data matrix X was arranged to have the structure shown in Equation 4.19, where t stands for the data sample time during a batch run and i stands for the iteration from one batch to the next.

$$\boldsymbol{X} = \begin{bmatrix} v_1(t_1, i_1) & v_2(t_1, i_1) & \dots & v_J(t_1, i_1) & \dots & v_J(t_2, i_1) & \dots & v_J(t_K, i_1) \\ v_1(t_1, i_2) & v_2(t_1, i_2) & \dots & v_J(t_1, i_2) & \cdots & v_J(t_2, i_2) & \dots & v_J(t_K, i_2) \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ v_1(t_1, i_I) & v_2(t_1, i_I) & \dots & v_J(t_1, i_I) & \cdots & v_J(t_2, i_I) & \dots & v_J(t_K, i_I) \end{bmatrix} \quad (4.19)$$

4.3.5 Adaptive PLS

Since MPLS is a linear modelling technique it is be unable to accurately represent the dynamics of the process over different operating conditions. It is therefore necessary for the model to adapt to the localised batch conditions. The purpose of MPLS models in this thesis is to track a number of batch process with certain operating conditions and then adapt the model from one batch to the next to keep a record of the changes in the dynamics of the batch from one region to another. The adaptive mechanism proposed in [49] was thus used to update the MPLS model at the end of each batch. This technique utilises a ‘forgetting factor’, λ, to ensure that the model forgets the behaviour of historical batches but remembers the most recent batches. The inclusion of the forgetting factor is essential as, without it, linear MPLS would not be able to track the changing characteristics from one batch to the next.

To adapt the MPLS model using this technique, the $\boldsymbol{X}$ and $\boldsymbol{Y}$ matrices are appended with data from the previous batch, as shown in Equation 4.20, where $\boldsymbol{x}_k$ and $\boldsymbol{y}_k$ are vectors containing the trajectories of the predictor and response variables for batch k. A new model is then identified by applying MPLS to these matrices. It should be noted that to forget past batches, λ needs to be positive and smaller than 1. In this thesis, λ is defined as a scalar, but it could be defined as a weighting vector where smaller values are associated with the oldest batches, and increasing values are assigned for the more recent batches.

$$\boldsymbol{X}_{\boldsymbol{k}+\mathbf{1}} = \begin{bmatrix} \lambda \boldsymbol{X}_{\boldsymbol{k}} \\ \boldsymbol{x}_k \end{bmatrix} \quad and \quad \boldsymbol{Y}_{\boldsymbol{k}+\mathbf{1}} = \begin{bmatrix} \lambda \boldsymbol{Y}_{\boldsymbol{k}} \\ \boldsymbol{y}_k \end{bmatrix} \tag{4.20}$$

This method’s drawback is that the size of the $\boldsymbol{X}$ and $\boldsymbol{Y}$ matrices increases with each batch, consequently increasing computational and storage requirements. To overcome this, [50] showed that the same results can be obtained if PLS is applied to the matrices defined in Equation 4.21. Applying PLS to these matrices has the advantage that the size of $\boldsymbol{X}$ and $\boldsymbol{Y}$ remains unchanged and hence the computational and storage requirements are not as demanding.

$$\boldsymbol{X}_{\boldsymbol{k+1}} = \begin{bmatrix} \lambda \boldsymbol{P}^T \\ \boldsymbol{x}_k \end{bmatrix} \; and \; \boldsymbol{Y}_{\boldsymbol{k+1}} = \begin{bmatrix} \lambda \boldsymbol{Q}^T \\ \boldsymbol{y}_k \end{bmatrix} \tag{4.21}$$

When selecting a suitable value for λ, it is necessary to consider the number of batches of data, *N*, that is to be *remembered* by the model. The approximate relationship between λ and N is given by Equation 4.22 [49]. The forgetting factor λ must be chosen such that the number of batches N is relevant to the conditions around which the process is currently operating.

$$N = \frac{1}{1-\lambda} \tag{4.22}$$

Where $0 < \lambda \leq 1$

If N is chosen to be too large, then the model will not adapt quickly enough to follow substantial changes in process dynamics. In contrast, if N is chosen to be too small, then the model will not consider enough past batches to obtain a useful model. In the work presented in this thesis, values of N between 5 to 10 were found to yield acceptable results, although this is likely to be problem dependent.

4.3.6 Missing data Algorithms

The objective of the optimisation design proposed in this thesis is to increase the yield from one batch to the next despite variation in the initial conditions. For this reason, some of the values in the predictor vector $\boldsymbol{x}$ in the optimization routine need to be estimated. A way to estimate these values is using missing data algorithms [6].

A variety of missing data algorithms have been proposed and compared in [114]. The authors of this article found the KDR (Known Data regression) method outperformed other algorithms over 3 case studies. According to this article, the advantages of Trimmed Score Regression (TSR) and Projection to the Modal Plane (PMP) is that they require less calculations but have the disadvantage that some matrices may be ill-conditioned which could lead to large score errors.

In the research developed for this thesis, in results obtained over the Pensim simulation, the TSR and the PMP algorithms outperformed other algorithms; the relevant results of these experiments are shown in Figure 4.4. In this figure, each graph presents the Mean Square Error (MSE) of the estimation of future values using each algorithm. The PMP method was thus chosen because of its satisfactory results in the case studies used in this thesis and its simplicity. The problem of the ill-conditioned matrices was solved by proposing limits in the minimum value of the variance in the variables among different batches.

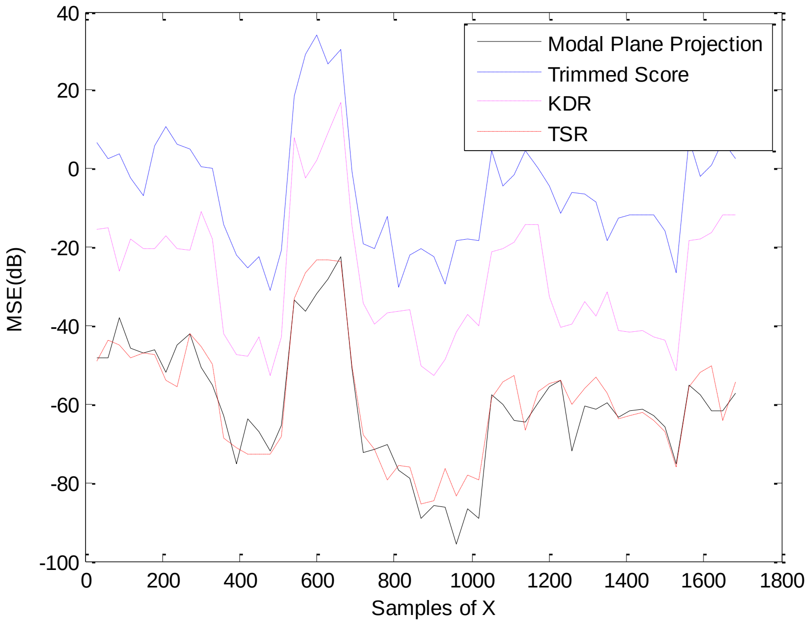


Figure 4.4: Estimation MSE for unknown variables using missing data algorithms for PLS prediction over Pensim

The PMP method for PCA estimates the score vector by regressing the known variables on the plane defined by the loading matrix $\boldsymbol{P}$. In this case, the score can then be estimated by regressing the known vector of input variables $\boldsymbol{x_p}$ to the plane defined by the loading matrix $\boldsymbol{P}$, as shown in Equation 4.23.

$$\widehat{\boldsymbol{t_k^T}} = \left(\boldsymbol{P_p^T P_p}\right)^{-1} \boldsymbol{P_p^T}\left(\boldsymbol{x_p}\right)^{\boldsymbol{T}} \tag{4.23}$$

where the matrix of known loadings $\boldsymbol{P_p^T}$ contains the rows of $\boldsymbol{P^T}$ that account for the known vectors of predictors $\boldsymbol{x_p}$.

Similarly, the score vector can be estimated for PLS by regressing the known variables on the plane defined by the weight vector $\boldsymbol{V}$ as equation 4.15. However, the bi-diagonal model proposed by Martens [112] used in this thesis has only one set of loadings ($\boldsymbol{V} = \boldsymbol{W} = \boldsymbol{P}$), this means that score can be estimated as in equation 4.24.

$$\widehat{\boldsymbol{t_k^T}} = \left(\boldsymbol{P_p^T W_p}\right)^{-1} \boldsymbol{W_p^T}\left(\boldsymbol{x_p}\right)^{\boldsymbol{T}} = \left(\boldsymbol{P_p^T P_p}\right)^{-1} \boldsymbol{P_p^T}\left(\boldsymbol{x_p}\right)^{\boldsymbol{T}} = \boldsymbol{V_p}\left(\boldsymbol{x_p}\right)^{\boldsymbol{T}} \tag{4.24}$$

The estimated score $\widehat{\boldsymbol{t_k}}$, can then be used to obtain an estimate of the future measurements as shown in Equation 4.25, where the matrix of future loadings $\boldsymbol{P_f^T}$ contains the rows which are related to the unknown variables $\boldsymbol{x_f}$.

$$\boldsymbol{x_f} = \widehat{\boldsymbol{t_k}} \boldsymbol{P_f^T} \tag{4.25}$$

4.4 SUMMARY

This chapter described the main theory of optimisation and modelling used for the proposed batch-to-batch optimisation described in Chapter 5. In conclusion, the main methodology that is employed can be described as follows:

- From Section 4.1, an **optimisation via model refinement** which is composed of several sequential steps and two main components: an **optimisation algorithm** and a **process model**.

- From Section 4.2, the **optimisation algorithm** has the structure described in Equation 4.2, which is used to calculate the optimal MVT from one batch to the next.

- From Section 4.3, the **process model** consists of a bilinear modelling technique expressed in Equations 4.7 to 4.14. This model calculates its parameters by utilising an adaptive MPLS regression, described in Equations 4.15 and 4.20 to 4.24.

Chapter 5: Batch-to-batch Optimisation

This chapter proposes an innovative design of a batch-to-batch optimisation and discusses its results. Section 5.1 describes the utilisation of an adaptive MPLS model to solve a QP cost function that provides an optimised MVT from one batch to the next. Next, Section 5.2 presents some preliminary experimentation with the case studies described in Chapter 3. The main results of the proposed optimisation follow in Section 5.3, along with its relevant discussion. Finally, Section 5.4 provides a summary of relevant findings in this chapter.

5.1 Batch-to-batch optimisation design

The main purpose of this design is to gradually increase the yield from one batch to the next while there is variation in the initial conditions. Moreover, this design can be classified as an 'optimisation via model refinement' approach and, as such, it requires the MPLS model to adapt to the most recent batch conditions. This section describes the model's adaptation and the process optimisation.

To approach these objectives, this thesis utilises **QP optimisation to identify the MVT** from one batch to the next. This optimisation looks for the changes $\Delta\boldsymbol{u}$ of a nominal MVT $\boldsymbol{u_n}$ which minimised the error between a reference y_{sp} and the predicted end-point quality $\hat{y}$. Equally important in the proposed design was the **adaptive PLS model** used within the QP optimisation. This model was updated using the theory described in Section 4.3.5 to track the change in the dynamics from batch to batch.

These two elements are the basic elements required for an optimisation via model refinement as described in Section 4.1. This design also contains two more elements: a **decision maker algorithm** to select the nominal MVT for the next batch optimisation and a **historical database** of the past end-point qualities and MVT.

The main flow diagram considering these elements in the optimisation design is shown in Figure 5.1.

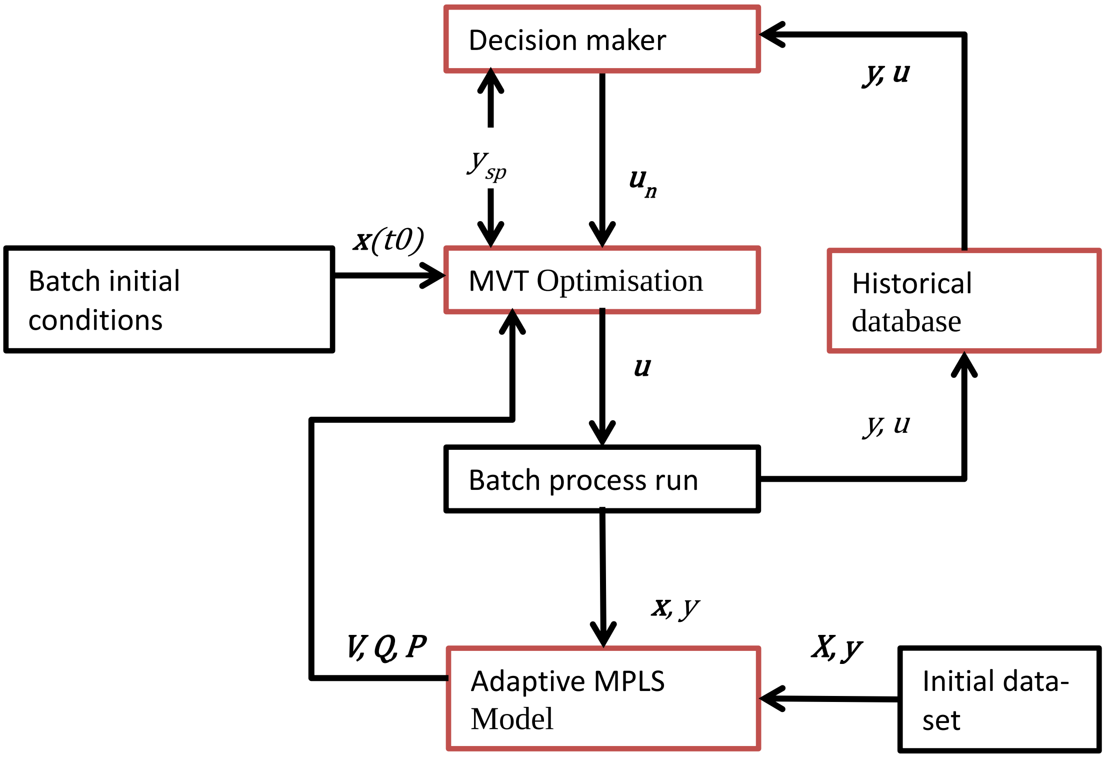


Figure 5.1: Flow diagram of the proposed batch-to-batch optimisation

These two extra elements were used together to improve even further the yield according to Equation 5.1.

$$\boldsymbol{u}_n(k+1) = \begin{cases} \mathbf{u}(\max(\boldsymbol{y})) & \mathrm{y}(\mathrm{k}) \le \max(\boldsymbol{y}) - E(\epsilon) \\ \mathbf{u}(\mathrm{k}) & \text{otherwise} \end{cases} \tag{5.1}$$

where $\max(\boldsymbol{y})$ is a function that searches the maximum end-point quality value in the historical database, and $E(\epsilon)$ is the expected value of the sensor noise in the end-point quality. The historical database in this equation was updated as well with the values the *N* batches considered in the adaptive MPLS model.

The design proposed in Figure 5.1 deals with the objective of Iterative Learning Control (ILC) in that it improves the input of the system by using the repetitive nature of most manufacturing processes to minimise the error between the reference and the output quality [85]. However, the design can also work towards the objective of Modal Predictive Control (MPC), or predicting the response and adjusting the input to disturbances [80]. This can be achieved by passing on-line measurements of the plant into the MVT optimisation algorithm instead of using the batch initial conditions. The disturbance rejection study is not included in this thesis and will be considered for future work.

To complete these objectives, the proposed design can also be divided into sequential stages as shown in Figure 5.2. Sections 5.1.1 to 5.1.6 describe each of the stages in detail.

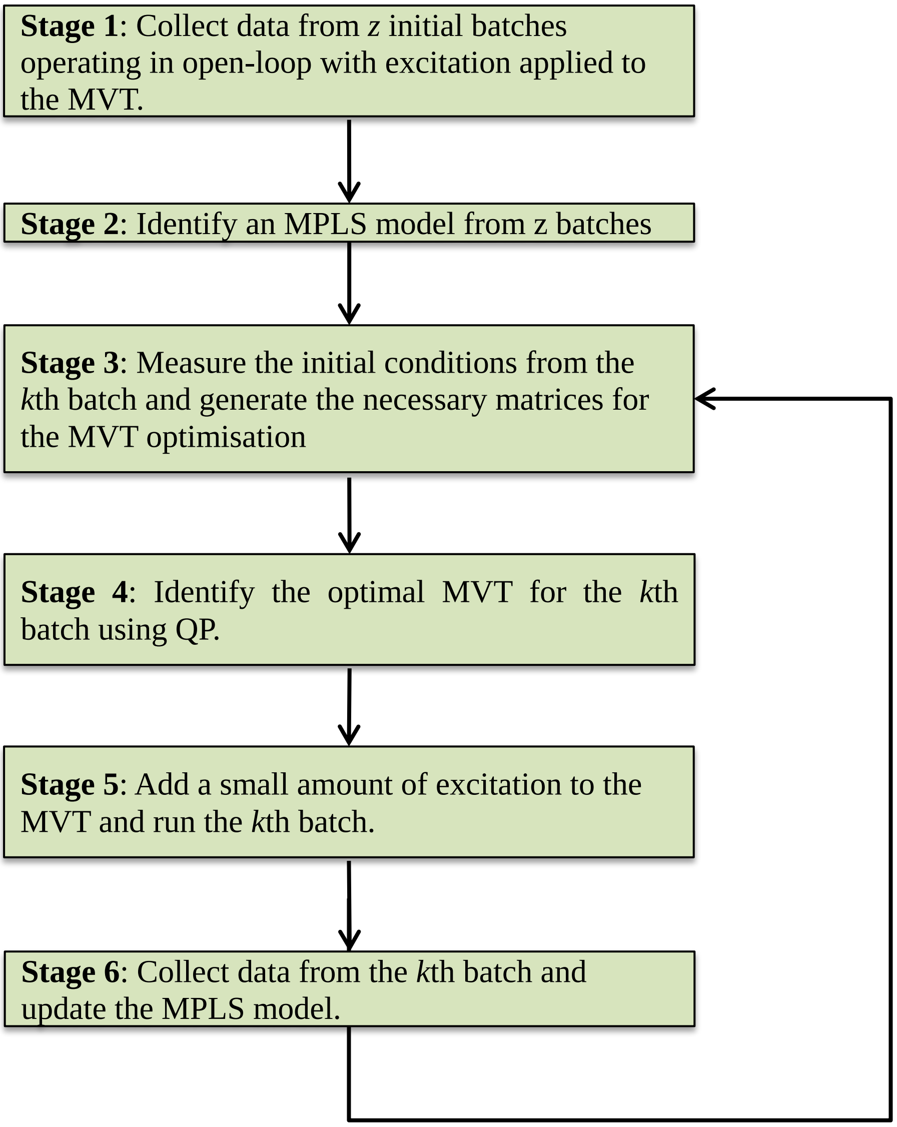

Stage 1: Collect data from z initial batches operating in open-loop with excitation applied to the MVT.
Stage 2: Identify an MPLS model from z batches
Stage 3: Measure the initial conditions from the kth batch and generate the necessary matrices for the MVT optimisation
Stage 4: Identify the optimal MVT for the kth batch using QP.
Stage 5: Add a small amount of excitation to the MVT and run the kth batch.
Stage 6: Collect data from the kth batch and update the MPLS model.

Figure 5.2: Diagram of the stages in the proposed batch-to-batch optimisation.

5.1.1 Stage 1: Run Pre-set batches

In this stage, the data from z batches were collected with the batch process operating in open-loop and a filtered Pseudo-Random Binary Sequence (PRBS) added to the nominal MVT. The reason to filter the PRBS is that in [90], [102], [103] it was suggested that the MVT be passed through a low-pass filter before being applied to the process. Filtering the MVT proved to be beneficial in the context of this paper, hence MVT was passed through a zero-phase, low pass Finite Impulse Response (FIR) filter with a cut- off frequency of 10% of the maximum frequency (Nyquist frequency) in each case study. Experimental results demonstrated little difference when the maximum frequency was varied between 3% and 10%.

The filtered PRBS was required to excite the process and allowed identification of a MPLS model. The PRBS can be added to a nominal MVT that was based on a previously identified trajectory, the trajectory of a 'golden batch', or if necessary a vector of zeros. The specified amplitude of the PRBS was low enough not to significantly alter the behaviour of the process, but high enough for the characteristics of the process remain distinct from background noise.

5.1.2 Stage 2: PLS Model Identification of the initial data-set

Once the data was collected in stage 1, an MPLS model was identified as described in Section 4.3.4. When modelling industrial applications, the number of batches of data required to identify the model should be minimised to reduce costs.

In experiments performed with 5, 20, 50 and 100 batches, it was observed that the end-point qualities reached their optimal values faster when data from more batches where used. However, running these *sub-optimal* batches is undesired in industrial production. Therefore, this thesis started the batch-to-batch optimisation with the minimal amount of batches necessary to build a PLS model. In this work it was found that good results were obtained when data from as few as three batches were used to identify the initial MPLS model.

5.1.3 Stage 3: Initialization for the MVT optimisation

The initial conditions for the current *k*th-batch were collected and the matrices required to define the optimisation function were arranged. This meant that the matrices $\boldsymbol{P}$and $\boldsymbol{V}$were arranged according to the predictors vector $\boldsymbol{x_k}$ for the *k*th-batch as shown in Equations 5.2, 5.3 and 5.4.

$$\boldsymbol{x_k} = [\boldsymbol{x_p} \quad \boldsymbol{u_n} + \Delta\boldsymbol{u} \quad \boldsymbol{x_f}] \tag{5.2}$$

$$\boldsymbol{P^T} = [\boldsymbol{P_p} \quad \boldsymbol{P_u} \quad \boldsymbol{P_f}] \tag{5.3}$$

$$\boldsymbol{V^T} = [\boldsymbol{V_p} \quad \boldsymbol{V_u} \quad \boldsymbol{V_f}] \tag{5.4}$$

Where the sub index p represents the part of the vector or matrix that accounts for the known measurements (initial conditions) in the batch, the sub index u represents the part of the vector or matrix that accounts for the MVT, and the sub index *f* represents the part of the vector or matrix that accounts for the variables to be estimated by the PMP technique shown in Section 4.3.6.

5.1.4 Stage 4: MVT Optimisation

This MVT was determined using a similar approach to that proposed in [6]. However, in this thesis, the cost function was formulated in the real MVT space, rather than in the latent variable (LV) space. The reason for this is that it is more meaningful to constraint the change in the real MVT. This avoids large variations caused by the change in the score that might lead to erratic control behaviour [115].

The cost function used to optimise the MVT is shown in Equation 5.5.

$$\min_{\Delta \boldsymbol{u}} \left(\widehat{\boldsymbol{y}} - \boldsymbol{y}_{sp}\right)^T \left(\widehat{\boldsymbol{y}} - \boldsymbol{y}_{sp}\right) + \Delta \boldsymbol{u}^T \boldsymbol{M} \Delta \boldsymbol{u}$$
$$s.t. \begin{cases} \widehat{\boldsymbol{y}} = \widehat{\boldsymbol{t}_k} \boldsymbol{Q}^T \\ \boldsymbol{lb} \leq \boldsymbol{u_n} + \Delta \boldsymbol{u} \leq \boldsymbol{ub} \end{cases} \tag{5.5}$$

where $\Delta \boldsymbol{u}$ is the vector of adjustments that is to be made to the MVT from the previous batch to minimise the error between the desired and predicted end-point quality, $\widehat{\boldsymbol{y}}$ and $\boldsymbol{y}_{sp}$ respectively; and where the diagonal matrix of weights $\boldsymbol{M}$ is used to moderate the change in the manipulated variable throughout the batch. The desired end-point quality $\boldsymbol{y}_{sp}$ can be provided by a production standard in real manufacturing applications. Another option for $\boldsymbol{y}_{sp}$ is to remove it from the equation and to change the sign of the first part of the equation to maximize the end-point quality.

Also in Equation 5.5, the optimised MVT is equal to the sum of the nominal value, $\boldsymbol{u_n}$**,** and the change in the MVT, $\Delta \boldsymbol{u}$**.** The minimisation problem is formulated to be constrained by physical limitations (such as the maximum flow allowed by a valve). These constraints are represented by the lower and upper bound vectors, $\boldsymbol{lb}$ and $\boldsymbol{ub}$ respectively.

Finally, the estimated score vector $\widehat{\boldsymbol{t}_k}$ for the new batch, k, in Equation 5.5 was obtained, accordingly to Equation 4.13, using the vector of measurements $\boldsymbol{x_k}$ into the weight matrix $\boldsymbol{V}$ as shown in Equation 5.6.

$$\widehat{\boldsymbol{t}_k} = \boldsymbol{x_k} \boldsymbol{V} \tag{5.6}$$

The effect that the change $\Delta\boldsymbol{u}$ has on the future estimates within $\boldsymbol{x_f}$ was also considered in the score that accounts for the unknown measurements, described in Equation 5.7 as $\widehat{\boldsymbol{t_f}}$**.**

$$\widehat{\boldsymbol{t_f}} = [\boldsymbol{x_p} \quad \boldsymbol{u_n} + \Delta\boldsymbol{u}]\left(\boldsymbol{P_p^T P_p}\right)^{-\mathbf{1}}\boldsymbol{P_p^T} \tag{5.7}$$

Introducing this term to Equation 5.6 and substituting in Equation 5.5, the full effect of the PMP estimation was included in the QP formulation as in Equation 5.8.

$$\widehat{\boldsymbol{y}} = [\boldsymbol{x_p} \quad \boldsymbol{u_n} + \Delta\boldsymbol{u} \quad \widehat{\boldsymbol{t_f}}\boldsymbol{P_f^T}]\boldsymbol{VQ}^T \tag{5.8}$$

The variables that are dependent on $\Delta\boldsymbol{u}$ from Equation 5.8 were then separated as shown in Equation 5.9.

$$\begin{aligned}\widehat{\boldsymbol{y}} = & \underbrace{\left[\boldsymbol{x_p} \quad \boldsymbol{u_n} \quad [\boldsymbol{x_p} \quad \boldsymbol{u_n}]\left(\boldsymbol{P_p^T P_p}\right)^{-\mathbf{1}}\boldsymbol{P_p^T P_f^T}\right]}_{\boldsymbol{\eta}}\boldsymbol{VQ}^T + \\ & \Delta\boldsymbol{u}\underbrace{\left(\left(\boldsymbol{P_p^T P_p}\right)^{-\mathbf{1}}\boldsymbol{P_u^T P_f^T V_f} + \boldsymbol{V_u}\right)}_{\boldsymbol{V_{uf}}}\boldsymbol{Q}^T\end{aligned} \tag{5.9}$$

Equation 5.9 was then used to reformulate the cost function of Equation 5.5 as shown in Equation 5.10.

$$\min_{\Delta\boldsymbol{u}}\left(\widehat{\boldsymbol{y}} - \boldsymbol{y}_{sp}\right)^T\left(\widehat{\boldsymbol{y}} - \boldsymbol{y}_{sp}\right) + \Delta\boldsymbol{u}^T\boldsymbol{M}\Delta\boldsymbol{u} \tag{5.10}$$

$$s.t.\begin{cases} \hat{\boldsymbol{y}} = \left(\boldsymbol{\eta V} + \Delta\boldsymbol{u}\boldsymbol{V}_{\boldsymbol{uf}}\right)\boldsymbol{Q}^T \\ \boldsymbol{lb} \leq \boldsymbol{u}_{\boldsymbol{n}} + \Delta\boldsymbol{u} \leq \boldsymbol{ub} \end{cases}$$

Finally, the previous equation can be expressed as a QP problem shown in Equation 5.11, which is the final mathematical expression used to optimise the MVT. For a detailed explanation, appendix 1 contains the complete mathematical formulation of the MVT optimisation.

$$\min_{\Delta\boldsymbol{u}} \ \frac{1}{2}\Delta\boldsymbol{u}^T\boldsymbol{H}\Delta\boldsymbol{u} + \boldsymbol{f}^T\Delta\boldsymbol{u}$$
$$s.t.\begin{cases} \boldsymbol{H} = \boldsymbol{V}_{\boldsymbol{uf}}\boldsymbol{Q}^T\boldsymbol{Q}\boldsymbol{V}_{\boldsymbol{uf}}^T + \boldsymbol{M} \\ \boldsymbol{f}^T = \left(\boldsymbol{\eta V}\boldsymbol{Q}^T - \boldsymbol{y}_{\boldsymbol{sp}}\right)\boldsymbol{Q}\boldsymbol{V}_{\boldsymbol{uf}}^T \\ \boldsymbol{lb} \leq \boldsymbol{u}_{\boldsymbol{n}} + \Delta\boldsymbol{u} \leq \boldsymbol{ub} \end{cases} \qquad (5.11)$$

5.1.5 Stage 5: Batch run

This stage consisted of running the *k*th batch with the optimised MVT obtained in the previous stage. As the data collected from the *k*th batch was used to adapt the model in the subsequent stage, it was necessary to introduce further excitation into the process. This was achieved by adding a low-amplitude PRBS to the optimised MVT. However, the amplitude of the PRBS should not be large enough to significantly change the end-point. The amplitude of the PRBS was specified to be equal to 3% of the mean of the amplitude of the MVT measured from one batch to the next. It was found, however, that there was not a significant difference in the results if the amplitude of the PRBS was varied between 1% and 5% of this amplitude. As an example, Figure 5.3 shows the MVT that resulted from solving Equation 5.11 compared to the MVT once it had been filtered and the PRBS had been added.

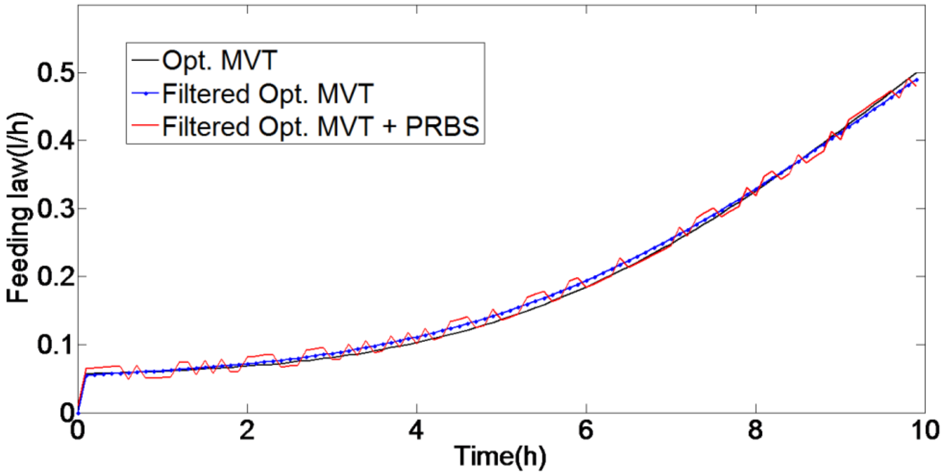


Figure 5.3: Optimised MVT filtered and PRBS-added for a Saccha simulation

5.1.6 Stage 6: Data collection and MPLS model updating

In this stage, data was collected at the end of the batch, and the MPLS model was updated, as described in Section 4.3.5. In general, the maximum number of LVs that could be utilised in the initial model was insufficient to enable the MPLS model to fully describe the process (2 latent variables for 3 batches). This was because the number of batches used to identify the initial model was limited [63]. Consequently, the model was updated using Equation 4.20 to increase the size of the $\boldsymbol{X}$ and $\boldsymbol{y}$ matrices by adding more batches. If the number of LVs was not limited, Equation 4.21 was then used, and the number of LVs remained constant. The number of LVs that produced the smallest mean squared error of cross validation (MSECV) also known as Mean Square Estimation Error of Cross Validation (MSEECV) is the point at which the remaining LVs were still relatively small. This was used as an estimator in leave-one-out cross validation to determine the number of LVs for the model. This process is explained in Section 5.2.

5.2 PRELIMINARY MODELLING

The proposed design predicts $\boldsymbol{y}$ from $\boldsymbol{X}$ within the MVT optimisation using an MPLS model. It is, therefore, necessary to consider several types of statistical information within the model. It is especially important to consider the average precision to be expected from the model, as there are many sources of variability that can affect the predictive capability of the model. The causes of such variability includes inherent model errors, random noise in the data sensors and lack of representation of variability in the process that describes future batches in the data used for identification [96].

The first and second causes of variability cannot be tackled directly in the identification stage. However, they must be considered to reach a solution within the control scheme, such as measurements smoothing to reduce noise error. The third cause of variability can be mitigated by considering the optimal number of LVs in the MPLS model. If there are not enough LVs to represent the general dynamic relationships of the process, then the model is *underfitted*. In contrast, the model is *overfitted* if there are too many LVs, such that the model describes noise or another external disturbance instead of the batch process.

There two types of techniques to decide the number of LVs to be used: The first type relies on sets of batches external from the calibration data to obtain indexes that measure the error between the predicted and real quality such as Root Mean Square of Prediction (RMSEP). The second type relies in Cross Validation (CV) [116], [117] of the calibration data set to obtain indexes such as the Prediction Error Sum of Squares (PRESS) and Root Mean Square Error of cross validation (RMSECV).

To obtain the number of LVs to be used for the MPLS model in the batch-to-batch optimisation a numerical calculation of the RMSEP with confidence limits can be used to obtain the number of LVs needed from batch to batch. To observe the effect of the number of LVs in the prediction of the MPLS model, 50 batches for calibration and 50 batches of test sets were used for each case study described in Chapter 3. Filtered PRBS was added to the MVT of each case study, as described in Section 5.1.1, and pseudorandom values of 5% and 3% amplitude were respectively added to the initial conditions to simulate uncertainties for the Saccha and Pensim simulations. The results of this experimentation are described in Sections 5.2.1 and 5.2.2.

5.2.1 Modelling of the case study: Saccha

The first method of assessing the quality of predictions of the model in this thesis was to obtain the RMSECV of the MPLS model for predicting the end-point quality using only the 50 batches in the calibration data set. Assuming that the test objects were representative of unknown future batches, this should illustrate the importance of finding the correct model complexity. The square root of the MSE may be preferable to the MSE because it is measured in the same units as y itself.

Figure 5.4 shows the RMSECV of the MPLS model and the Saccha simulation using CV in 10 groups of objects for 10 LVs. This figure illustrates how MSECV develops as a function of the number of LVs.

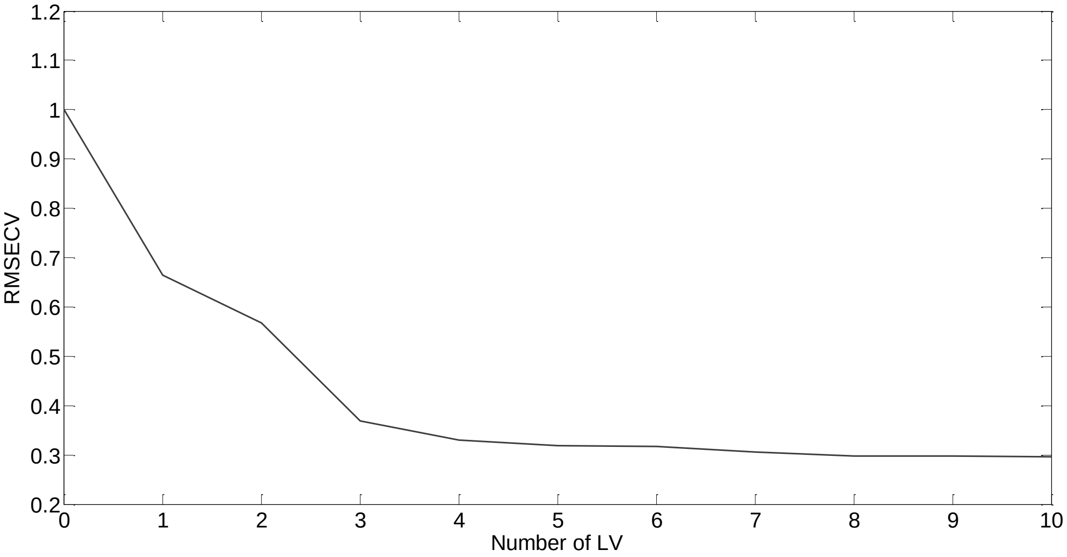


Figure 5.4: Validation using RMSECV for prediction of the Saccha simulation

It is clear from Figure 5.4 that the model experiences a marked decrease of the RMSECV for the first 4 LVs, and gradually drops with the addition of more LVs. This could indicate that all LVs have at least some degree of relevant information regarding the internal dynamics of the batch process. A good indicator of the correct number of latent variables to use is the point at which RMSECV has the smallest values or does not drop further [96]. From this figure we can conclude that 3 to 5 should be a reasonable number LVs to choose.

It is important to notice that in real manufacturing situations this analysis would require costly sub-optimal batches. If previous data from past process runs is available, this could be used to determine the number of LVs to use at the beginning of the batch-to-batch optimisation. On the other hand, if no previous data is available, it is possible to evaluate the numerical values of the RMSECV over a threshold to obtain the optimal number of LV to be used for each batch optimisation.

As an example, the RMSECV obtained in the cross-validation of 4, 5, 10 and 20 batches in the calibration data-set can be observed in Figure 5.5.

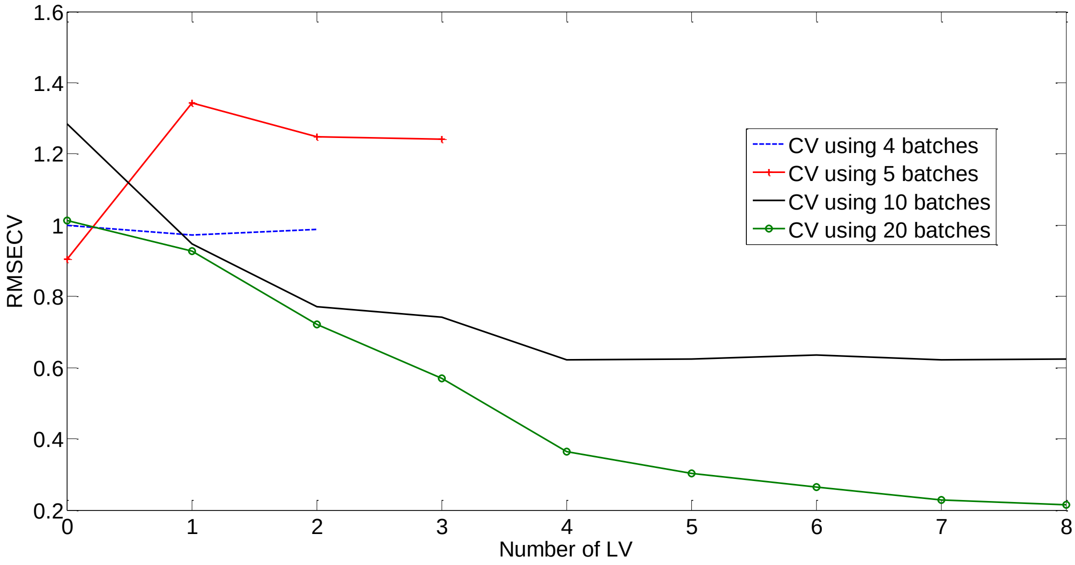


Figure 5.5: Validation using RMSECV for prediction of the Saccha simulation using 4, 5, 10 and 20 batches in the calibration data-set.

As can be observed in Figure 5.5, it is difficult to analyse the results of the RMSECV using just 2 or 3 LV, however, as the number of batches increases, it is possible to evaluate the numerical values for successive batch optimisations. This suggests that cross validation is not sensible when very few batches are used in the calibration data-set at the beginning of the batch-to-batch optimisation. However, this topic was left for future research. Instead, in this thesis, the initial calibration data-set to build the MPLS model uses the maximum number of LVs until 5 LVs have been reached; then, the RMSECV is obtained iteratively to determine the number of LVs to be used in the MPLS model.

The test data set was used to test the model accuracy with the selected number of LVs by observing the correlation between the fitted (predicted) and the observed responses. This is illustrated in Figures 5.6, 5.7 and 5.8: The first figure shows the *overfitting* sample using a 10-LVs MPLS model. The second figure shows the *optimal fitting* case using a 4-LVs MPLS model; although the test set response is close to the *overfitting* case. Finally, the third figure shows the *underfitting* case using a 2-LVs MPLS model where it is clear that the predicting ability of the MPLS over the test and calibration sets decreases drastically.

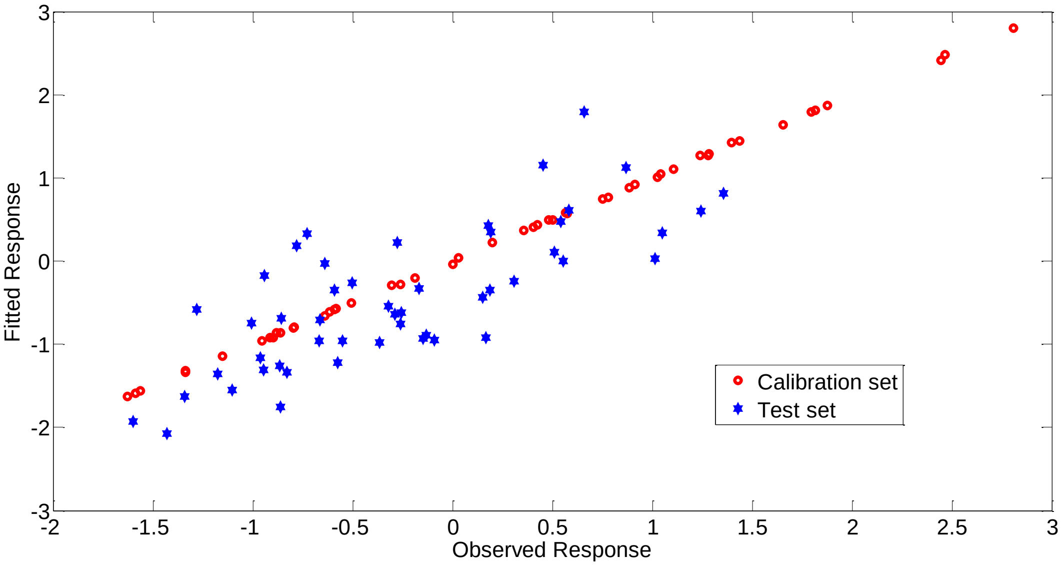


Figure 5.6: *Overfitting* illustrated by the correlation between the fitted and observed responses using 10 LVs

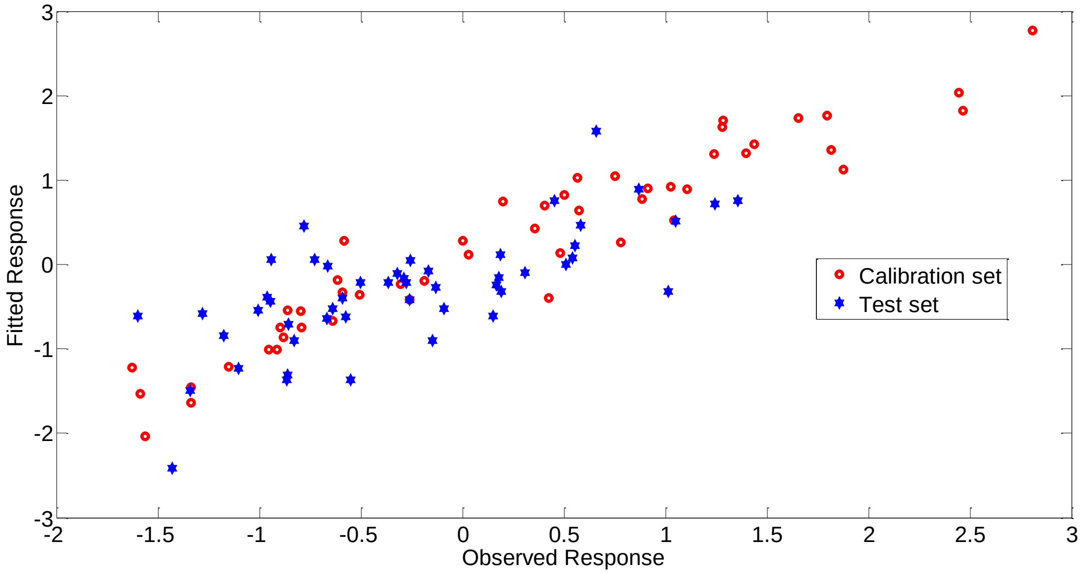


Figure 5.7: *Optimal fitting* illustrated by the correlation between the fitted and observed responses using 4 LVs

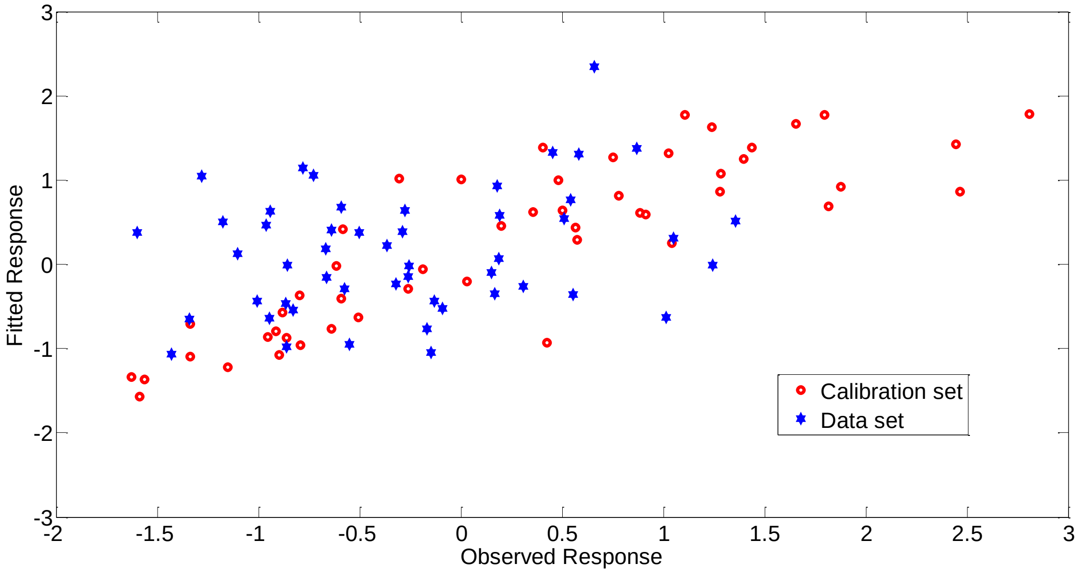


Figure 5.8: *Underfitting* illustrated by the correlation between the fitted and observed responses using 2 LVs

The findings of these experiments suggest that it is not necessary to use more than 3 or 4 LVs, as the model prediction did not show improvement in the RMSECV or the responses correlation if more LVs were used. However, the proposed optimisation in Section 5.1 also needs to predict future measurements using the MPLS model. For this reason it was necessary to observe the percent of variance explained by the MPLS model over the calibration set, as illustrated in Figure 5.9.

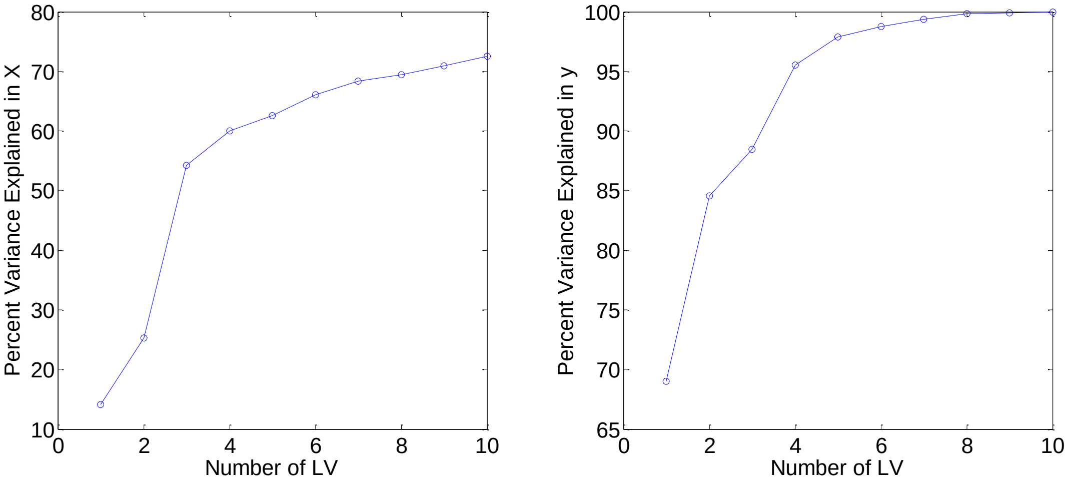


Figure 5.9: Percent of variance explained in $\boldsymbol{X}$ and $\boldsymbol{y}$ by the MPLS model

The first graph in Figure 5.9 shows the sharp increase in percent of variance explained in $\boldsymbol{X}$ from 1 to 4 LVs, this could also be a good indicator of the number of LVs to be used for the model.

5.2.2 Modelling of the case study: Pensim

The same assessment used for the MPLS model of the Saccha simulation was also used for the Pensim simulation. Figure 5.10 shows the RMSECV of the MPLS model using CV in 10 groups of objects for 10 LVs.

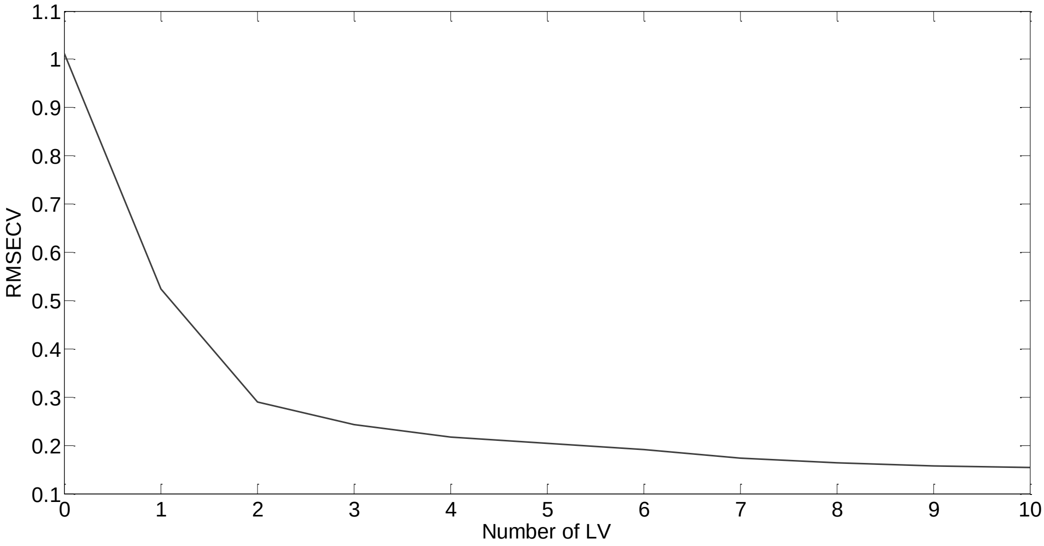


Figure 5.10: Validation using RMSECV for prediction of the Saccha simulation

This figure reveals that the model has a marked decrease of the RMSECV for the first 2-4 LV and gradually drops with the addition of more LV, similar to the Saccha case study.

Figure 5.11 shows the RMSECV obtained in the cross-validation of 4, 5, 10 and 20 batches in the calibration data-set, observing similar complications when the number of batches is relatively small in the calibration data-set.

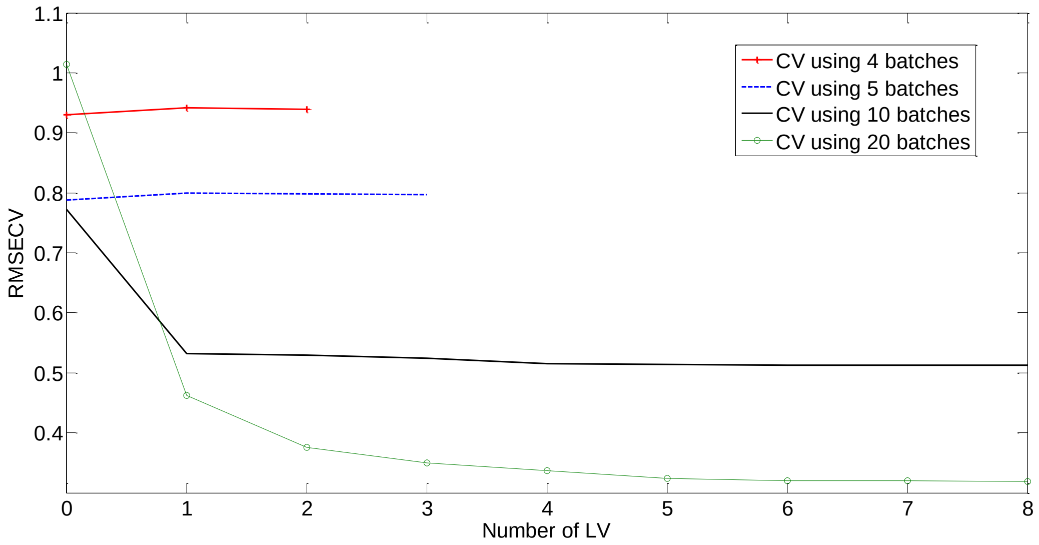


Figure 5.11: Validation using RMSECV for prediction of the Pensim simulation using 4, 5, 10 and 20 batches in the calibration data-set.

Figure 5.12 shows the correlation between the fitted and the observed responses for the *optimal fitting* case using a 4-LVs MPLS model, which shows similar predictive capability than with the Saccha simulation for the test set, although some extrapolation in the magnitude exists.

Finally, Figure 5.13 illustrates the percent of variance explained by the MPLS model over the calibration set.

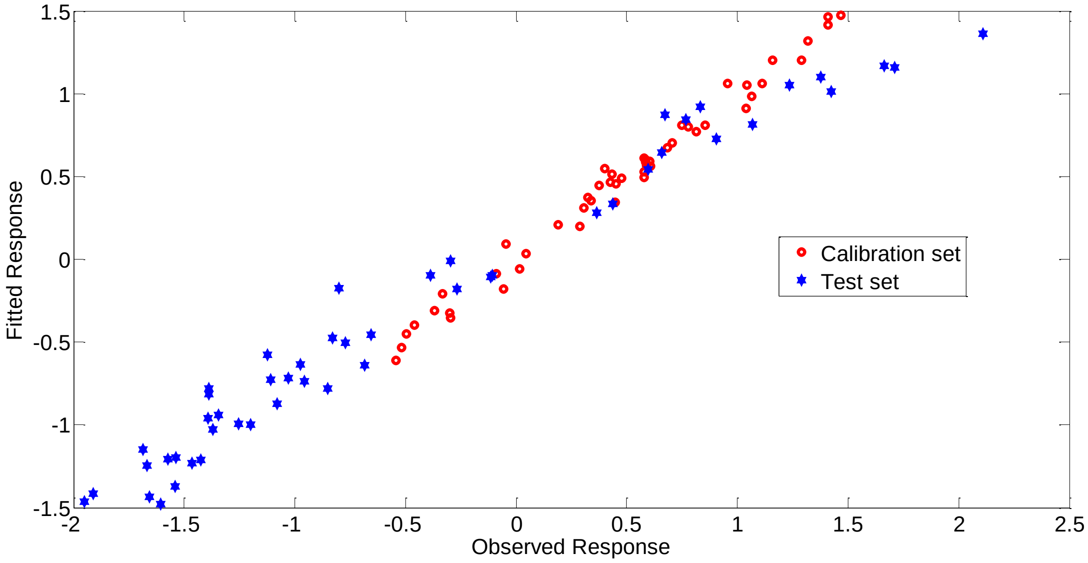


Figure 5.12: *Optimal fitting* illustrated by the correlation between the fitted and observed responses using 4 LVs

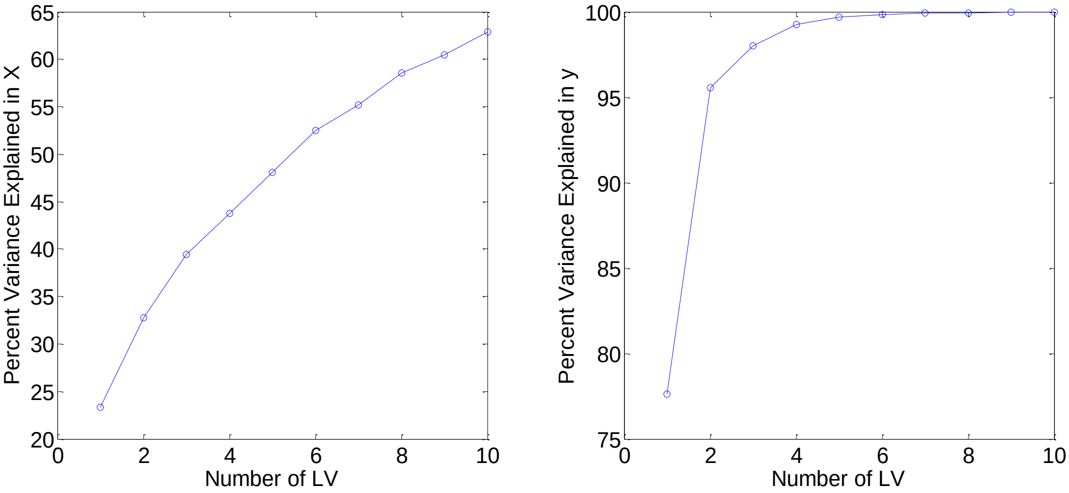


Figure 5.13: Percent of variance explained in $\boldsymbol{X}$ and $\boldsymbol{y}$ by the MPLS model

The results from the majority of the experiments in this section indicate that the MPLS model can probably describe both case studies using only 4LVs. Thus the adaptive MPLS model built up from the initial data-set was expanded using Equation 4.20 up to 4 LVs. After this, the number of LVs was kept under 6 LVs obtaining the RMSECV to calculate the LVs for the remaining batches using 4.21 when new batches measurements were added to the model.

5.3 Results and discussion

Turning now to the analysis of the results, the output data used to evaluate the batch-to-batch control performance was analysed from several experiments to confirm reproducibility. In accordance with this data structure and to compare the experimental results generated in this work with those obtained from Camacho et.al [90], each output data-set from Saccha in this thesis consisted of 100 discrete points of final biomass concentration from 100 experiments over 100 batches.

Figure 5.14 illustrates the output distribution of results from 100 Saccha batch-to-batch experiments.

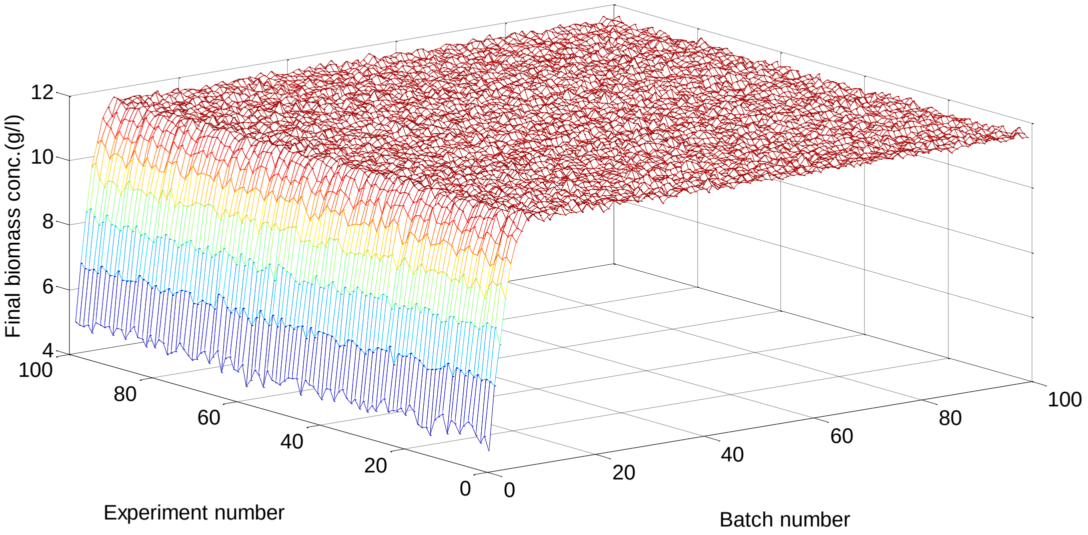


Figure 5.14: Example of Saccha end-point quality optimisation from 100 experiments over 100 batches.

Likewise, the results of this thesis were compared with those obtained in [9]. Each output data-set for Pensim consisted of 100 discrete points of final penicillin concentration from 30 experiments over 100 batches. Figure 5.15 shows the output distribution of results from 100 Pensim batch-to-batch experiments.

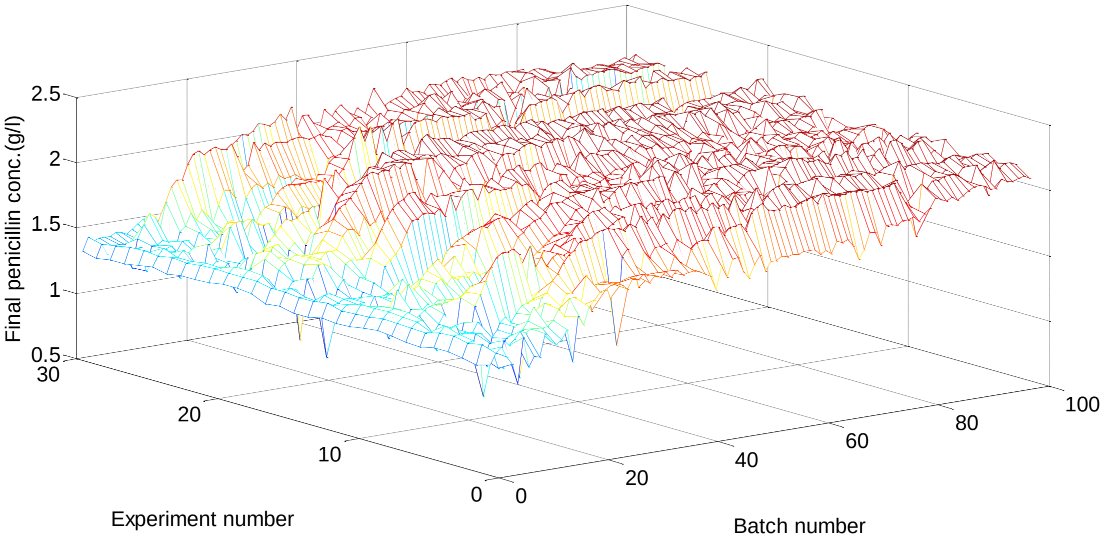


Figure 5.15: Example of Pensim end-point quality optimisation from 30 experiments over 100 batches.

Figures 5.14 and Figure 5.15 display the end-point quality variability for all the experiments in one output data-set. In spite of this, the visual analysis of results in these figures is not ideal as there is an excess of information to be comprehended. This thesis conveniently presents the end-point quality results using the average value of all the experiments along with information about statistics of the optimisation.

The vector of initial conditions $\boldsymbol{x_p}$ was obtained from the initial states of the process variables for the Saccha simulation, and during the initial batch phase of the Pensim simulation, where no glucose was fed into the reactor.

5.3.1 Results of case study: Saccha

The initial PLS model used for this case study was identified using a training dataset containing 3 batches with 5% variability in the manipulated variable and in the initial conditions of the biomass concentration, active cell material and the acetaldehyde dehydrogenase. In addition 1% Gaussian noise was added to the end-point quality to simulate sensor noise.

The optimization problem included constraints of 0.0 (l/h) and 0.6 (l/h) in the magnitude of $\Delta\boldsymbol{u}$ at each time interval, simulating actuator constraints. In addition, a volume constraint (9 litres) was added to match the parameters that were used to those used in [90].

This constraint was included in the cost function by considering the mean, $\boldsymbol{mean}_u$, and standard deviation, $\boldsymbol{std}_u$, of the manipulated variable as shown in Equation 5.12.

$$s[1\ \ 1\ \dots 1]_u \bar{\boldsymbol{u}} \leq V_{max} - V_{ini}$$

Where $$\bar{\boldsymbol{u}} = \boldsymbol{diag}(\boldsymbol{std}_u)(\boldsymbol{u_n} + \Delta\boldsymbol{u}) + \boldsymbol{mean}_u \qquad (5.12)$$

Where the terms s is the sample interval, V_{max} is the final accepted volume and V_{ini} is the initial volume. Equation 5.12 can be redefined by the inequality constraint shown in Equation 5.13.

$$\boldsymbol{A}\Delta\boldsymbol{u} \leq \boldsymbol{b}$$

$$Where\;\; \boldsymbol{A} = s[1\;\;1\;\;\ldots 1]_u \boldsymbol{diag}(\boldsymbol{std}_u) \qquad (5.13)$$

$$\boldsymbol{b} = V_{max} - V_{ini} - \;t_i[1\;\;1\;\;\ldots 1]_u(\boldsymbol{mean}_u{}^T + \boldsymbol{diag}(\boldsymbol{std}_u)\boldsymbol{u_n})$$

Figure 5.16 shows the final volume that was measured from 100 batches when the volume constraint was imposed within the cost function. This figure shows that the 9 (l) volume constraint is occasionally exceeded. The reason for this is that the constraint does not consider the small PRBS that is added at the end of the MVT optimisation. This should not be a problem in real applications, as long as there is some minor overcapacity allowed in the vessel.

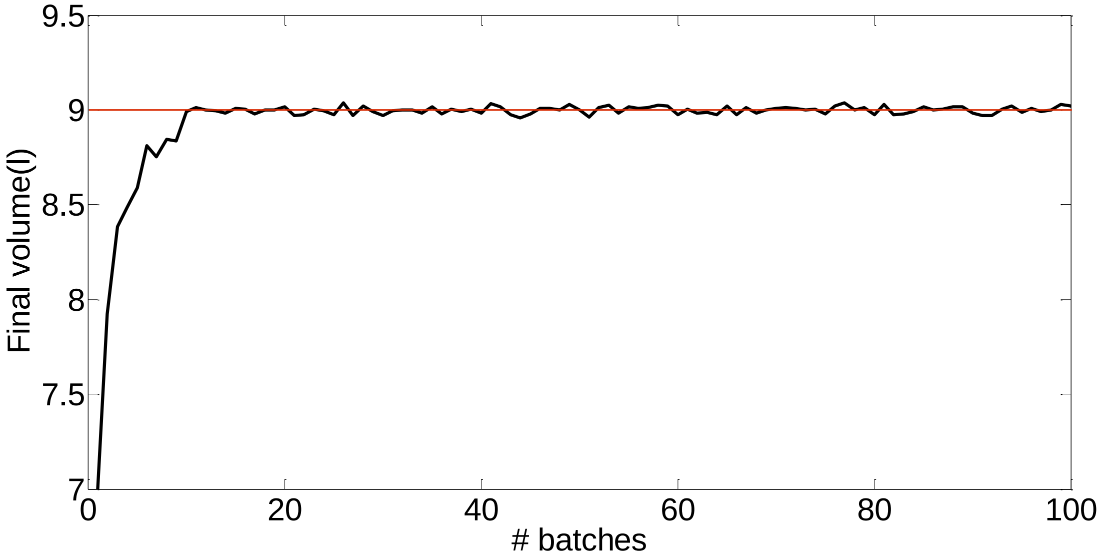


Figure 5.16: Saccha final volume constraint over 100 batches

Figure 5.17 shows the results obtained when the proposed batch-to-batch optimisation technique was applied to the Saccha simulation. This figure shows that the final biomass concentration increased steadily from about 2 (g/l) to 11.3 (g/l) and after approximately 15 batches levels off at this output concentration. This figure shows that the final biomass concentration increased consistently to approximately 11.3 (g/l) in each experiment. The standard deviation for the end-point biomass concentration, measured over the 100 experiments was 0.2 g/l. Figure 5.14 also shows the consistency of the proposed approach when it was applied to 100 experiments on the Saccha simulator.

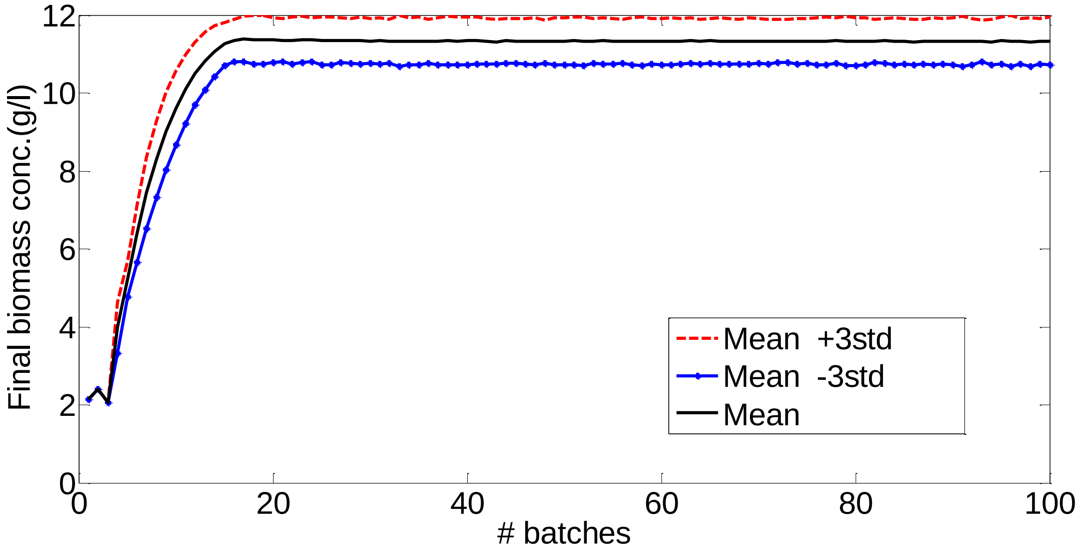


Figure 5.17: Saccha final biomass concentration mean over 100 experiments of 100 batches each.

The resulting end-point biomass concentration compares very favourably with other results applied to this simulator. In [90] for example, their batch-to-batch optimisation technique increased biomass concentration to 10.74 g/l, with a standard deviation of 0.16 (g/l), after approximately 60 batches. Although the work was not specifically attempting to maximise batch yield, [4] and [18] recorded end-point biomass concentrations of approximately 10 (g/l). A recent publication [103] reported final biomass concentrations of around 10.66 (g/l) at the end of 100 batches run, which is still lower that that reported in this thesis.

Figure 5.18 shows how the trajectory of the substrate feed changes during a typical experiment after the initial-data-set (3 batches). This figure shows that the trajectory changes considerably from the first through to the final batch. The feeding law presented in this graph does not include the additional PRBS that was applied to these trajectories.

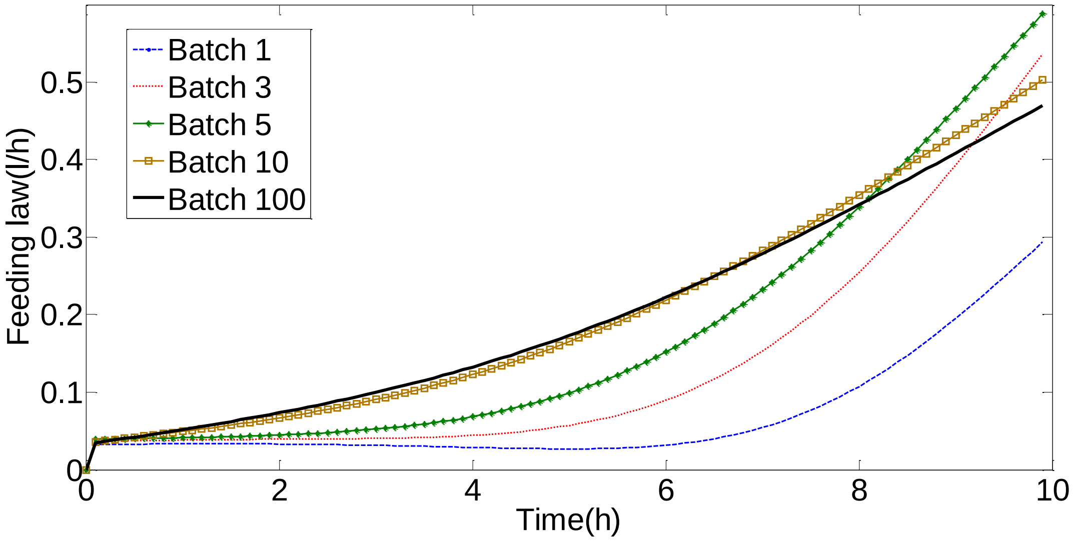


Figure 5.18: Saccha MVT evolution from batch to batch for a single experiment of 100 batches after the initial data-set

The prediction accuracy of the adaptive MPLS model throughout the batch to batch optimisation can be assessed from the Root Mean Squared Error of Prediction (RMSEP) after the MVT optimisation shown in Figure 5.19.

This figure illustrates not only the quality of predictions of the MPLS model, but also the QP solution performance. As expected, the RMSEP dropped as soon as more batches were added to the model. When the 100th-batch RMSEP of 0.26 g/l was compared to the 100th-batch end-point quality of 11.3 g/l, it was found that the Error/Output relationship of 2.3% was not far from the added 1% output noise.

These findings suggest that the predictive capability of the optimisation was constant through the whole 100 batches, as long as the number of LVs was kept high.

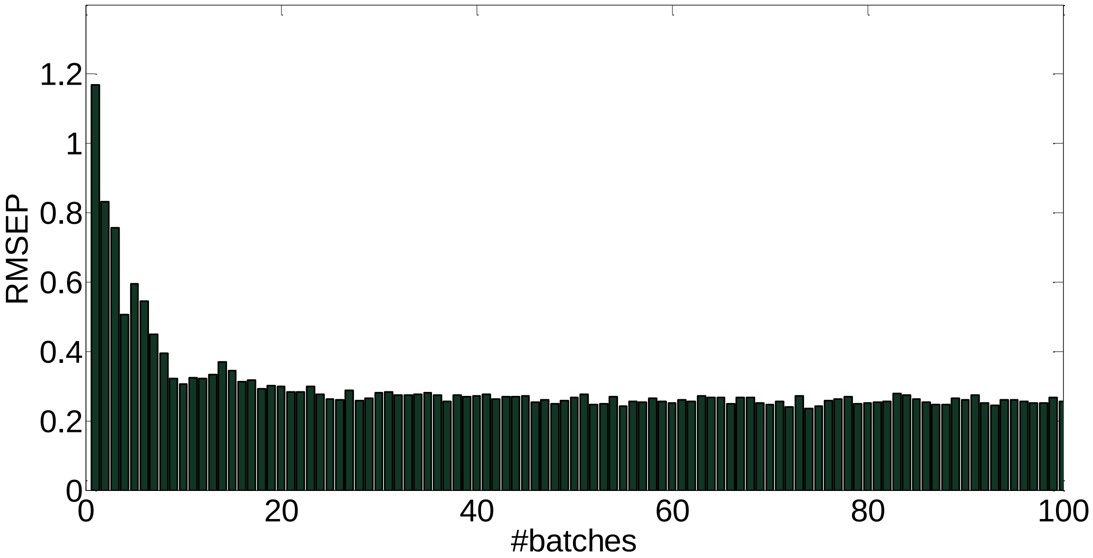


Figure 5.19: Saccha Root Mean Squared Error of Prediction after the MVT optimisation over 100 experiments of 100 batches each.

5.3.2 Results of case study: Pensim

The initial PLS model used for this case study was identified using a training dataset containing 3 batches with 3% variability in the manipulated variable and in the initial conditions of the biomass concentration, glucose concentration and temperature. The only additions for this work was that hard constraints where imposed on the glucose feed rate magnitude, which was restricted to between 0.0 (l/h) and 0.2 (l/h) and, and the addition of 1% Gaussian noise to the end-point quality to simulate sensor noise.

Figures 5.20 shows how the end-point penicillin concentration increases over 100 batches for the Pensim simulator following the introduction of the proposed batch-to-batch optimisation system. In this figure the presented results are the average taken over 30 experiments. These results show that the end-point penicillin concentration increases to approximately 2.11 g/l.

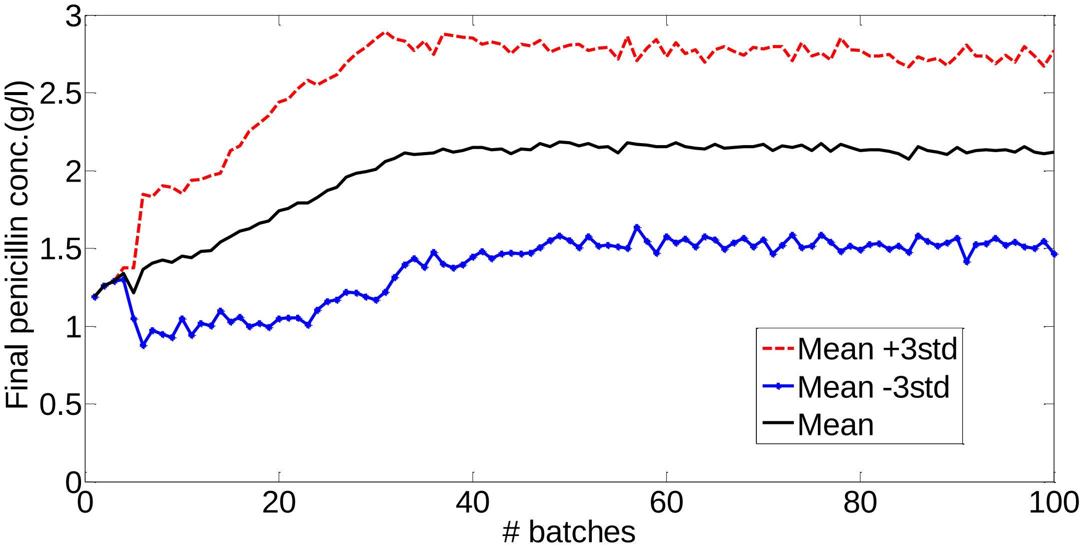


Figure 5.20: Pensim final penicillin concentration mean over 100 experiments of 100 batches each.

The average value of the culture volume obtained at the end of the batch was approximately 120 l, which means that the average quantity of penicillin produced from each batch was approximately 253 g. This result compares favourably to [20], where the final penicillin production was 106 g, and the results of the evolutionary optimisation technique presented in [104], which produced 160 g of penicillin.

Figure 5.21 shows how the MVT evolves over 100 batches for a single experiment. This figure shows that the MVT for the first batch switches to fed-batch mode after approximately 45 hours.

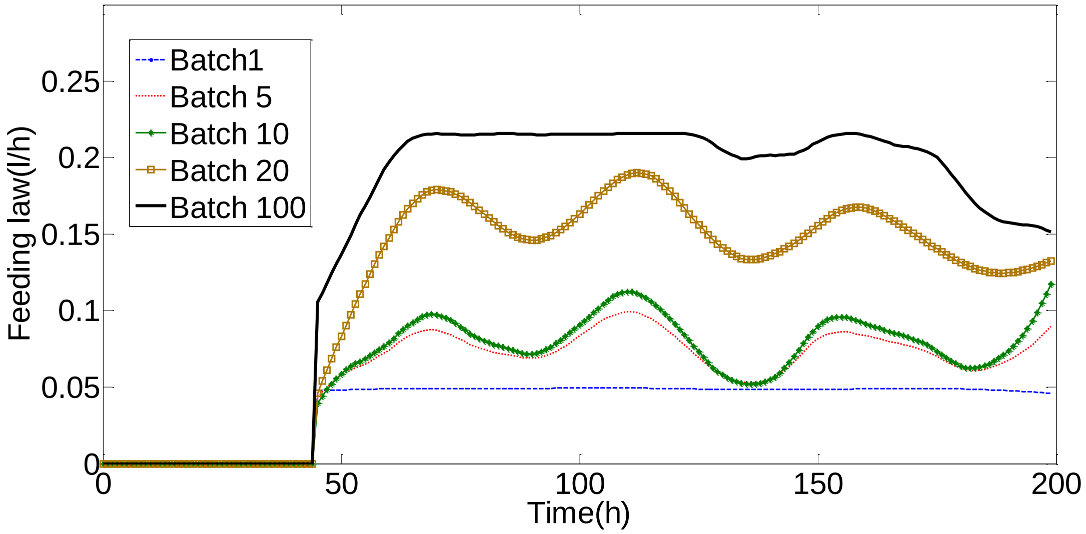


Figure 5.21: Pensim MVT evolution from batch to batch for a single experiment of 100 batches

Figure 5.22 shows the RMSEP after the MVT optimisation from 100 experiments of 100 batches each.

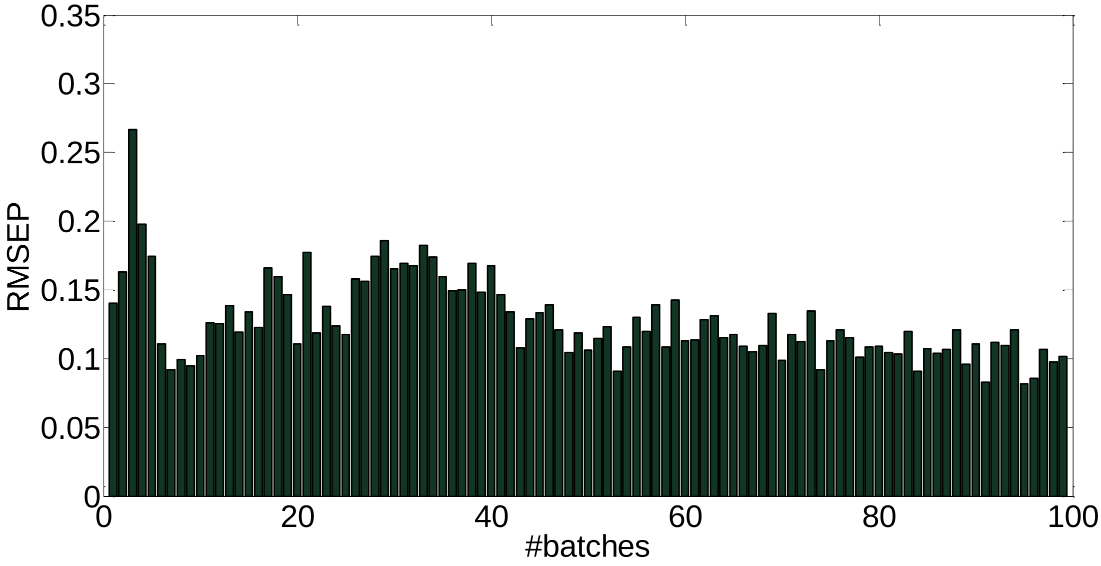


Figure 5.22: Pensim Root Mean Squared Error of Prediction after the MVT optimisation over 100 experiments of 100 batches each.

The RMSEP shown in Figure 5.22 did not decrease as much as in the Saccha simulation as soon as new batch were added into the MPLS model. However, the accuracy gradually improved from initial RMSEP values near 0.2 (g/l), to values of 0.1 (g/l) at the 100th batch. This may be because the MPLS model could not accurately predict the batch dynamics from batch to batch. The Error/Output relationship was 4.7%, which is higher than the added output noise. However, the overall control methodology showed improvements in the end-point quality.

5.4 Summary

This chapter proposed an innovative batch-to-batch optimisation technique that can be used to improve the productivity of batch processes. The technique initially identifies an MPLS model of a batch process. This model is used with an optimisation function to identify the trajectory of any manipulated variables in the process, such as the target end-point quality. By constraining the changes allowed in the identified trajectories of the manipulated variables and by updating the PLS model at the end of each batch, the productivity of the batch was gradually increased from one batch to the next.

The predictive capabilities and optimisation convergence of the proposed technique were demonstrated through their application to two benchmark fed-batch fermentation simulators: Saccharomyces cerevisiae and Pensim. For both simulators, the proposed batch-to-batch optimisation technique was found to increase productivity considerably over a limited number of batches.

Chapter 6: Validity Constraints in Batch-to-batch Optimisation

This chapter proposes and implements the addition of Validity Constraints (VC) in the batch-to-batch optimisation scheme. The objective of this extension is to restrict the QP optimisation to the space only described by useful predictions of the MPLS model. This chapter is divided as follows: First, Section 6.1 presents a brief theoretical framework. Then, Section 6.2 formulates the extension to the main batch-to-batch optimisation to consider validity constraints. In Section 6.3, this formulation is implemented for 2 case studies and their results are discussed. Finally, Section 6.4 provides a summary of the relevant points in this chapter.

6.1 VALIDITY OF PREDICTIONS

The optimisation proposed in Chapter 5 increased the yield from one batch to the next. As this happens the MPLS model moves from one region of operation to another. Since the MPLS algorithm represents a local linear approximation of the process, it is necessary to know the region over which the model will produce sensible predictions. Once this is achieved then the control action could use this information to restrict the MVT that are calculated to regions where the accuracy for the model should be good. In this thesis, this is achieved by introducing constraints in the QP optimisation, which restricts the solution space to a region spanned by the operating conditions experienced in previous batches used to identify the MPLS model. This should ensure that the model provides reliable estimates of the end-point quality prediction within the MVT optimisation.

Some statistical parameters have already been proposed for measuring the quality of predictions of the MPLS model in Chapter 6, such as RMSEP and RMSECV. These parameters use the mean prediction error (MSE) between the predicted end-point quality and the measured one from a whole population of batches. However, these parameters alone cannot asses the quality of the prediction for a single new batch using on-line measurements within the batch. The quality of predictions can be assessed using confidence intervals similar to those used in multivariate control charts [8]. The next section describes some commonly used multivariate control charts along with their statistical indicators. These indicators are used later in the methodology proposed in this chapter.

6.1.2 Multivariate control charts

Multivariate control charts can be used to detect shifts in the mean or the covariance between variables within a process. In PCA and PLS, these charts are used to detect *outliers* from the normal behaviour of a process. Statistical indicators are used to determine if the PC or LV of a certain batch lies within the limits of those considered as NOC.

Two widely used multivariate control charts for monitoring and diagnosis in batch processes [64] are the T^2 and the Square Prediction Error (SPE), described in the next paragraphs:

1. The T^2 control chart is based on multivariate, normal distribution theory. This chart, also called Hotelling’s statistic chart, is used to detect shifts in the Latent Variables or Principal Components from what is considered the norm relative to the identification dataset. This statistic can give an indication as to

whether the process variables are within the region of NOC. The T^2 index using PCA for the kth batch is given in Equation 6.1 [26] .

$$T^2{}_k = \boldsymbol{t_k S^{-1} t_k}^{\boldsymbol{T}} \tag{6.1}$$

where S is the estimated covariance matrix of the scores from the batches in NOC. This is a diagonal matrix such that each element is the variance of the score of each batch in the identification dataset.

The confidence limit at significance level α for the T^2 index can be calculated using the Snedecor F-distribution as shown in Equation 6.2.

$$T^2{}_\alpha = \frac{m(m-A)}{A(m^2-1)} \sim F_{k,m-k,\alpha} \tag{6.2}$$

where m is the number of batches used in the identification data-set under NOC and A is the number of principal components retained in the model.

2. The SPE chart is particularly useful in PCA to determine the process variable that is responsible for a certain shift in the chart. It utilises an index, also known as Q index, which contains information about the predictor residual of a given batch. The SPE statistic for the kth batch is given in Equation 6.3 [118].

$$SPE_k = \boldsymbol{e_k e_k^T} = \sum_{i=1}^{JK} \left(x(i)_j - \hat{x}(i)_j\right)^2 = \boldsymbol{x_k(I - PP^T)x_k}^{\boldsymbol{T}} \tag{6.3}$$

The confidence limit for the SPE index level α can be obtained using Equation 6.4.

$$SPE^2{}_{\alpha} = \left(\frac{v}{2\mu}\right)\chi^2_{\chi^2_{2\mu^2/v,\alpha}} \qquad (6.4)$$

where $\chi^2_{2\mu^2/v,\alpha}$ is the critical value of the chi-squared variable with $2m^2/v$ degrees of freedom, v is the variance and μ the mean of the SPE identification dataset.

The same approach adopted for multivariate charts constructed from MPCA can be extended directly to MPLS for monitoring, according to Nomikos and MacGregor [28]. However, the batch-to-batch optimisation proposed in Chapter 5 changes continuously the number of batches and the number of LVs used for the identification. In addition, the forgetting factor λ in the MPLS update from one batch to the next presents a complication to obtain the confidence limits. This optimisation thus required another strategy to calculate confidence limits in the predictions.

6.1.3 Validity Restrictions

A simple method to constrain the solution of an optimisation problem to the region of score space defined by previous operations is to add model validity constraints. Constraints of this type were first proposed by Russel et al. [119]. In their article, the T^2 index shown in Equation 6.1 was used as a soft constraint for an MPC optimisation problem. This methodology was later employed in the work published in [6] to restrict the changes in the score $\Delta \boldsymbol{t}$ over a MPC optimisation as shown in Equation 6.5.

$$\min_{\Delta\boldsymbol{t}} \ \frac{1}{2}\Delta\boldsymbol{t}^{\boldsymbol{T}}\boldsymbol{M}\Delta\boldsymbol{t} + \boldsymbol{f}^{\boldsymbol{T}}\Delta\boldsymbol{t} + \gamma T^{\boldsymbol{2}}$$
$$\text{where} \quad T^{\boldsymbol{2}} = \sum_{i}^{A}\frac{(\Delta\boldsymbol{t}+\hat{\boldsymbol{t}})_i^2}{s_i^2} \tag{6.5}$$

where γ is a weighting factor which determines how tightly the solution is to be constrained to the region of the score space defined by previous batches, $\boldsymbol{M}$ is the Hessian matrix of their QP formulation and $\boldsymbol{f}$ is a function related to the end-point quality in the score space.

A similar approach was used in the work published in [10] to optimise the estimated score. In addition, a quadratic error relating to the residual, based in the SPE index from Equation 6.3 was used as a validity indicator in the form of Equation 6.6.

$$\min_{\boldsymbol{t}} \ \frac{1}{2}\Delta\boldsymbol{t}^{\boldsymbol{T}}\boldsymbol{M}\Delta\boldsymbol{t} + \boldsymbol{f}^{\boldsymbol{T}}\Delta\boldsymbol{t} + \lambda_t J_t + \lambda_e J_e$$
$$\text{where} \qquad J_t = \boldsymbol{t}(\boldsymbol{S}_{\boldsymbol{\alpha}}^{\boldsymbol{2}})^{-\boldsymbol{1}}\boldsymbol{t}'$$
$$J_e = \boldsymbol{e}\boldsymbol{e}^{\boldsymbol{T}} \tag{6.6}$$

where the validity indicators J_t and J_e are two indices that determine the validity of the model, and are functions of the score $\boldsymbol{t}$ and the residuals $\boldsymbol{e}$. To provide confidence limits to these indicators, the authors proposed the normalization shown in Equation 6.7.

$$\lambda_t = \frac{1}{J_{tmax}}$$
$$\lambda_e = \frac{1}{J_{emax}} \tag{6.7}$$

where $J_{tmax}$is the value of the $T^{\boldsymbol{2}}$ index and J_{emax} is the value of SPE index that includes 95% of the previous batches in the identification dataset. This is analogue to the use of distributions to calculate the confidence intervals in the previous section.

This approach, although useful, does not ensure that the constraint will be respected. To address this, hard constraints were applied to the cost function in the approach adopted in [8]. This optimisation function used in this approach is shown in Equation 6.8.

$$\begin{aligned} &\min_{\boldsymbol{t}} \ \frac{1}{2}\boldsymbol{t}^T\boldsymbol{H}\boldsymbol{t} + \boldsymbol{f}^T\boldsymbol{t} \\ &\boldsymbol{s.t.}\begin{cases} J_t \leq \mathbf{1} \\ J_e \leq \mathbf{1} \end{cases} \end{aligned} \tag{6.8}$$

The advantage of using validity indicators as hard constraints is that it ensures the validity of predictions by avoiding extrapolation of the QP solution into a region where the model has not been identified. However, it may also lead to infeasibility in the minimisation if the process moves away from the region where the past batches were identified. This is certainly the case for the proposed batch-to-batch optimisation in Chapter 5, as the yield increases iteratively from one batch to the next. This problem can be solved by allowing extrapolation and choosing a value greater than 1.0 for the limit of the validity indicators.

6.2 Validity constrains in the MVT optimisation

In the previous section, validity measures have been used as a weighting or as constraints within the cost function to prevent erratic changes in the score vector [6], [10]. However, in this thesis, rather than applying the constraints in the LV space, they were applied in the real space. Equation 6.9 shows the addition of the hard validity constraints in the MVT optimisation proposed in Chapter 5.

$$\min_{\Delta u} \frac{1}{2}\Delta \boldsymbol{u}^{\boldsymbol{T}}\boldsymbol{H}\Delta \boldsymbol{u} + \boldsymbol{f}^{\boldsymbol{T}}\Delta \boldsymbol{u}$$
$$s.t.\begin{cases} \boldsymbol{H} = \boldsymbol{V}_{\boldsymbol{uf}}\boldsymbol{Q}^{\boldsymbol{T}}\boldsymbol{Q}\boldsymbol{V}_{\boldsymbol{uf}}{}^{\boldsymbol{T}} + \boldsymbol{M} \\ \boldsymbol{f}^{T} = \left(\boldsymbol{\eta}\boldsymbol{V}\boldsymbol{Q}^{\boldsymbol{T}} - \boldsymbol{y}_{\boldsymbol{sp}}\right)\boldsymbol{Q}\boldsymbol{V}_{\boldsymbol{uf}}{}^{\boldsymbol{T}} \\ \boldsymbol{lb} \leq \boldsymbol{u}_{\boldsymbol{n}} + \Delta \boldsymbol{u} \leq \boldsymbol{ub} \\ \\ J_t(\Delta \boldsymbol{u}) \leq L_t \\ J_e(\Delta \boldsymbol{u}) \leq L_e \end{cases} \tag{6.9}$$

where the score validity indicator J_t and the residuals validity indicator J_e for kth batch must be less than their pre-defined limits L_t and L_e. These limits define how far the scores of the batch can move away from the scores of the calibration data. The limits are chosen to have a value of 1 if the scores are to be constrained within the calibration range or greater than 1 if some extrapolation is to be allowed. The validity indicator J_t is based on the Hotelling's statistic T^2 and provides a measure of the deviation of the score $\boldsymbol{t}_{\boldsymbol{k}}$ from the region covered by the identification dataset. Equations 6.10 and 6.11 provide the mathematical definition of J_t.

$$J_t = \frac{\boldsymbol{t}_{\boldsymbol{k}}(\boldsymbol{S}_{\boldsymbol{\alpha}}^{\boldsymbol{2}})^{-\boldsymbol{1}}\boldsymbol{t}_{\boldsymbol{k}}{}^{\boldsymbol{T}}}{J_{tmax}} \tag{6.10}$$

$$\boldsymbol{t}_{\boldsymbol{k}} = \boldsymbol{\eta}\boldsymbol{V} + \Delta\boldsymbol{u}\boldsymbol{V}_{\boldsymbol{uf}} \tag{6.11}$$

where the diagonal matrix of covariance $\boldsymbol{S}_{\boldsymbol{\alpha}}^{\boldsymbol{2}}$ contains the covariance of each LV in the identification dataset and the normalization variable J_{tmax} is the value of the T^2 statistic taken at a 95% confidence limit [8]. The validity indicator J_e provides a quadratic measure of the error between the predictor vector, $\boldsymbol{x}_{\boldsymbol{k}}$ for the current batch and its value when reconstructed from the scores in the LV space. Equations 6.12 and 6.13 show the definition of the validity indicator J_e.

$$J_e = \frac{\boldsymbol{e_k}\boldsymbol{e_k}^{\boldsymbol{T}}}{J_{emax}} \tag{6.12}$$

$$\boldsymbol{e_k} = \boldsymbol{\eta}(\boldsymbol{I} - \boldsymbol{V}\boldsymbol{P}^{\boldsymbol{T}}) + \Delta\boldsymbol{u}(\boldsymbol{I} - \boldsymbol{V}_{\boldsymbol{uf}}\boldsymbol{P}^{\boldsymbol{T}}) \tag{6.13}$$

where the normalization variable J_{emax} is equal to the value of the square of the error, $\boldsymbol{e}\boldsymbol{e}^{\boldsymbol{T}}$ at the 95% confidence limit [8].

This section has formulated the extension to the MVT optimisation considering validity constraints in Equations 6.9 to 6.13. This extension proposes to replace Equation 5.11 with Equation 6.9 in the batch-to-batch optimisation scheme presented in Chapter 5. The results from this methodology over two case studies will be discussed in the next section.

6.3 Results and discussion

The output data used to evaluate the batch-to-batch control performance was analysed from several experiments to confirm reproducibility using the same policy as Chapter 5. Table 6.1 summarizes the experiment design.

In addition, this chapter will include different limits in the validity constraints to observe the response of the batch-to-batch optimisation scheme to different degrees of *restriction* in the MVT optimisation. Table 6.2 describes the labels used in the graphs shown in this section.

Table 6.1: Experiment design for this section

Simulation	Data structure	Additional Constraints	Initial variability	Output noise
Saccha	100 experiments each consisting of 100 batches	$0\ (\mathrm{l/h}) \leq \boldsymbol{u} \leq 0.6\ (\mathrm{l/h})$ $V_{max} \leq 9\ (\mathrm{l})$	3% in the biomass concentration, active cell material and acetaldehyde dehydrogenase	1% of the amplitude
Pensim	30 experiments each consisting of 100 batches	$0\ (\mathrm{l/h}) \leq \boldsymbol{u} \leq 0.2\ (\mathrm{l/h})$	3% in the biomass concentration, glucose concentration and temperature	1% of the amplitude

Table 6.2: Label notation

Abbreviation	Description
No V. C.	No Validity Constraints were used for these experiments; the results are the same as those in Chapter 5.
Je<1 Jt<1	The limits for J_e and $J_t$were defined to be less than or equal to 1, exactly as those spanned by the identification data set.
Je<4 Jt<4	The limits for J_e and $J_t$were defined to be less than or equal to 4, to allow extrapolation from the identification data set.
Je<4	The limit for J_e was defined to be less than or equal to 4, to observe the effects of J_e alone.
Jt<4	The limit for J_t was defined to be less than or equal to 4, to observe the effects of J_t alone.

6.3.1 Results of case study: Saccha

The results presented in this section provide a comparison of the results obtained when the Validity Constraints (VC) introduced in Section 2.3 were introduced. Figure 6.1 shows the final biomass concentration obtained for the Saccha simulation when various limits were placed on the values of J_e and J_t.

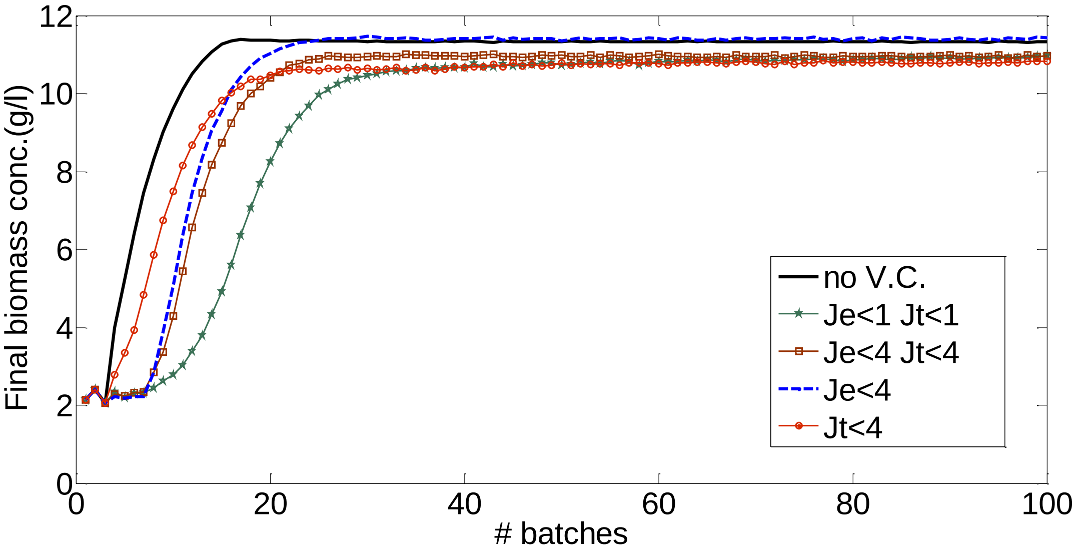


Figure 6.1: Final biomass concentrations mean under varying validity constraints for the Saccha simulator.

Table 6.3 displays the average of the mean and the standard deviation of the last 5 batches of the graphs shown in Figure 6.1. These statistics provide an indication of the mean and the standard deviation after the process has converged. Additionally, the worst and best values are marked for each study case, with red and green colours respectively. It can be observed that the end-point variation when constraints were introduced on to J_t alone was the highest, whilst the variation was lowest when constraints on J_e alone were introduced.

Table 6.3: Saccha standard deviations and means average of the final biomass concentration for the 5 last batches of Figure 6.1.

	No VC	Je<1 Jt<1	Je<4 Jt<4	Je<4	Jt<4
Standard deviation	0.194	0.541	0.742	0.125	0.817
Mean	11.30	10.93	11.01	11.32	10.88

The results presented in Figure 6.1 show that the effect of introducing the validity constraints is to reduce the rate at which biomass increases during the first 20 batches. This is to be expected as the constraints will have the effect of limiting the allowable change in MVT from one batch to the next, which will slow the rate of convergence of the optimisation strategy. It can also be observed that in most cases, the introduction of validity constraints results in a slight reduction in the final biomass concentration. This is because the introduction of the validity constraints has the effect of limiting the solution space over which the optimisation algorithm searches over and hence it may not identify the same solution as was obtained with no validity constraints.

A possible reason for the increase in the standard deviation when validity constraints were introduced is that the solution of the QP program is more constrained in the optimization search when using few LVs at the beginning of the 100 batch optimisation. This may prevent the optimisation algorithm from reaching the optimal solution as fast as without it. The exception lies with the use of J_e where the standard deviation decreases. The reason for this decrease may be that the model achieves better predictions and hence the performance of the controller is improved.

Figure 6.2 shows how the trajectory of the substrate feed changes during a typical experiment over 100 batches with the validity constraints of Je<4 Jt<4 after the initial data-set (3 batches). This figure was included to observe the constraining effect of the validity constraints over the batch-to-batch evolution. This figure shows that the trajectory changes considerably from the first through to the final batch. However, when compared with Figure 5.16 it is clear that the constraints in the score space also constrained the MVT evolution, particularly during the first few batches.

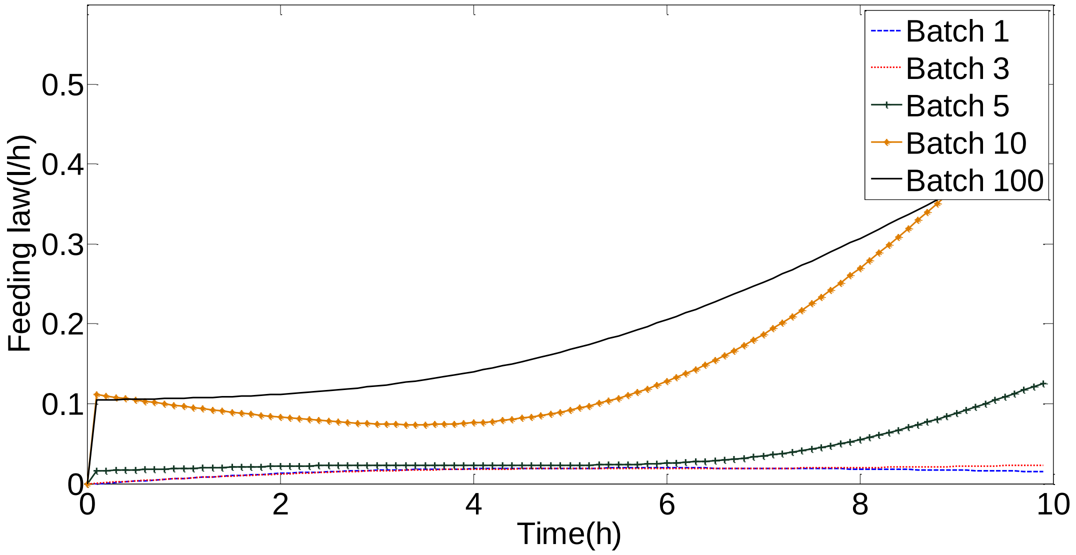


Figure 6.2: Saccha MVT evolution from batch to batch for a single experiment of 100 batches after the initial data-set with the validity constrains Je<4 Jt<4 shown in Figure 6.1.

Figure 6.3 shows the RMSEP of the MPLS model during the 100 batch test. This figure shows that the introduction of the validity constraints did not improve significantly the accuracy of the model. The RMSEP for the three cases shown in Figure 6.3 gradually decreases to values close to 0.2 g/l at the 100th batch. The most constrained case (Je<1 Jt<1) shows the lowest overall prediction error through the 100 batches run, while the unconstrained case showed the highest error.

A likely explanation for the results observed in Figure 6.3 is that the prediction accuracy of the MPLS model for the two constrained cases is superior to the non-constrained case, as the model is kept within the region in the score space spanned by previous batches.

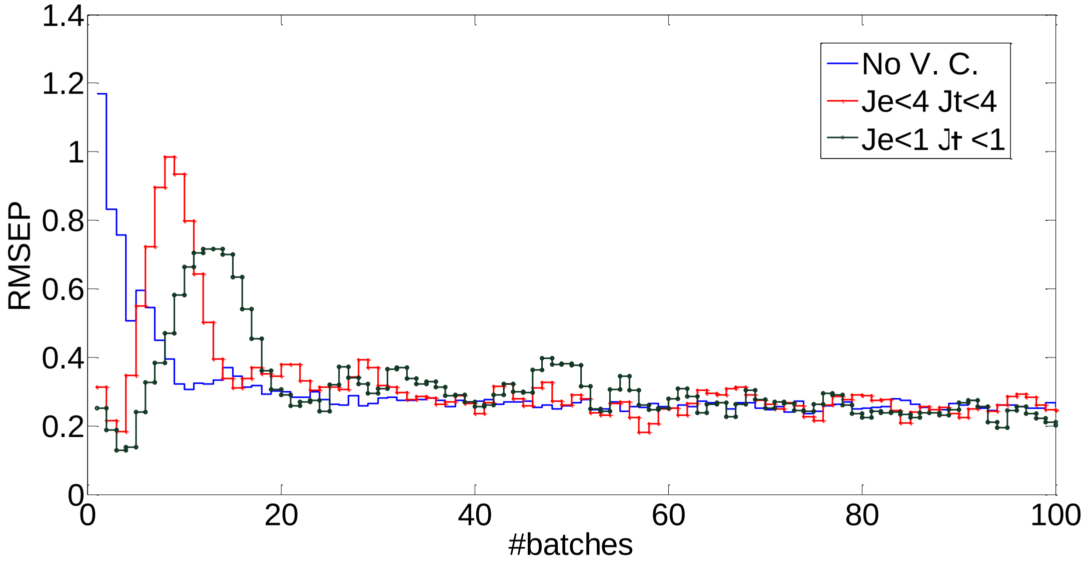


Figure 6.3: Saccha Root Mean Squared Error of Prediction after the MVT optimisation.

Figure 6.3 also reveals a gradual decrease in the RMSEP at the start of the batch-to-batch run when not using validity constraints. However there is an increase of the error after a few batches in the constrained cases. This may be caused by the Validity Constraints, which are mainly working at the start of the batch-to-batch run when few batches are used to build a MPLS model and the new batch trajectory cannot be accurately optimised quality too far from that used in the initial dataset. This knowledge could be used to design a multi-stage optimisation which could switch off the validity constraints until the model has enough information. However, this work was not included in this thesis and will be considered for future work.

6.3.2 Results of case study: Pensim

Figure 6.4 shows the final penicillin concentration when varying validity constraints were applied to the Pensim simulator. As with the Saccha simulator, this figure shows that the batch-to-batch optimisation strategy converges at a slower rate when the constraints are introduced.

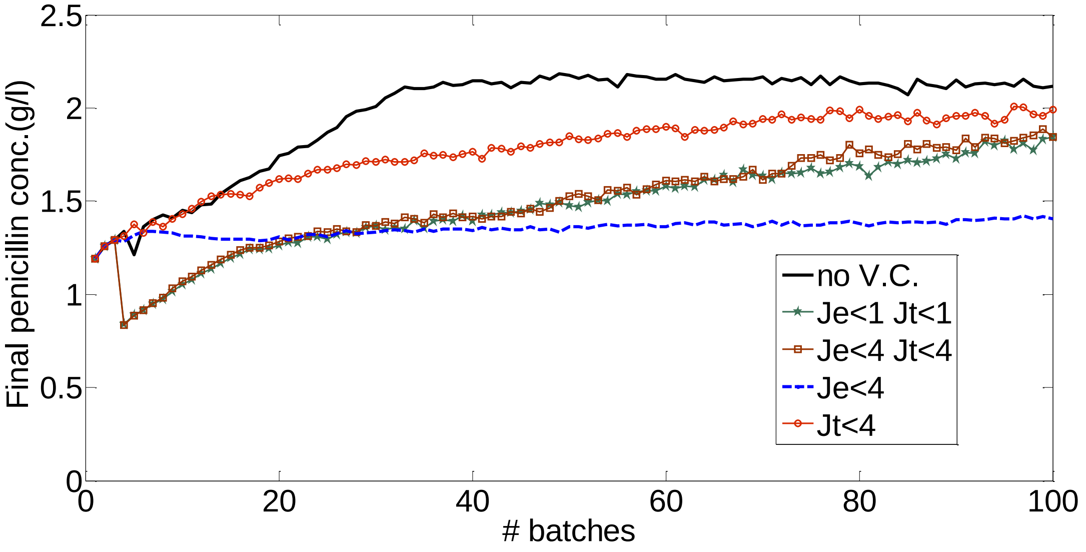


Figure 6.4 Final biomass concentrations mean under varying validity constraints for the Saccha simulator.

Table 6.4 displays the mean of the standard deviation of the last 5 batches of the graphs shown in Figure 6.4, where similarly to the Saccha simulator, it can be observed that the values when just J_t was present had the highest value while the ones where just J_e was included had the lowest value.

Table 6.4: Pensim standard deviations and means average of the final biomass concentration for the 5 last batches of Figure 6.4.

	No VC	Je<1 Jt<1	Je<4 Jt<4	Je<4	Jt<4
Standard deviation	0.203	0.237	0.299	0.141	0.321
Mean	2.11	1.86	1.93	1.42	1.97

Although the introduction of the validity constraints has the effect of slowing convergence and reducing the end-point yield slightly, they have the positive benefit of reducing variation in the results. For example, variation was minimised when the constraint that $J_e < 4$ was introduced; the standard deviation when this constraint was implemented was 0.125 g/l for Saccha and 0.141 g/l for Pensim, which compares with 0.19 g/l and 0.20 g/l when no validity constraints were considered. It is anticipated that the validity constraints would be of significant benefit with any real application where the plant operator would only be willing to see slight variations made to operating conditions from one batch to the next.

Figure 6.5 shows the trajectory of the substrate feed changes during a typical experiment over 100 batches with the validity constrains Je<4 Jt<4 from Figure 6.4. When compared with Figure 5.19 it is clear, as with the Saccha simulator, that the constraints in the score space also constrained the MVT evolution, especially during the first few batches.

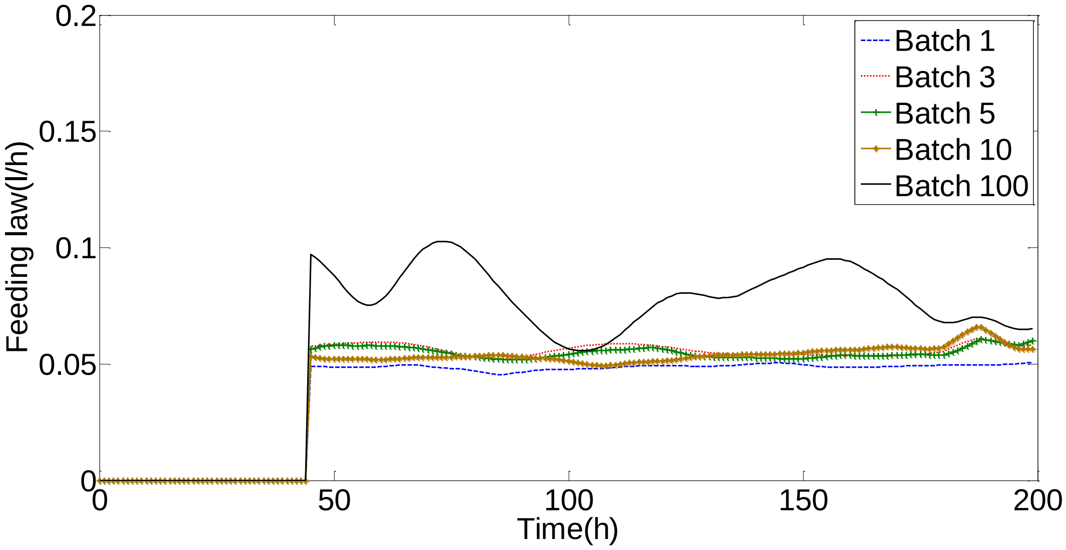


Figure 6.5: Pensim MVT evolution from batch to batch for a single experiment of 100 batches with the validity constrains Je<4 Jt<4 shown in Figure 6.4.

Figure 6.6 shows the RMSEP after the MVT optimisation for 3 validity constraints limits cases shown in Figure 6.4. The RMSEP for the two constrained cases in this figure show a very different trajectory to the non-constrained case.

The data for the experiments in Figure 6.6 was later analysed and it was found that the error in the SPE in the adaptive identification set was very small compared to that in the batches near the end of the 100 batches run; therefore, the optimisation was substantially constrained by the validity constraint J_e. The reason for this could be that the model was *overfitted* as the information of new batches was added and the number of LVs increased.

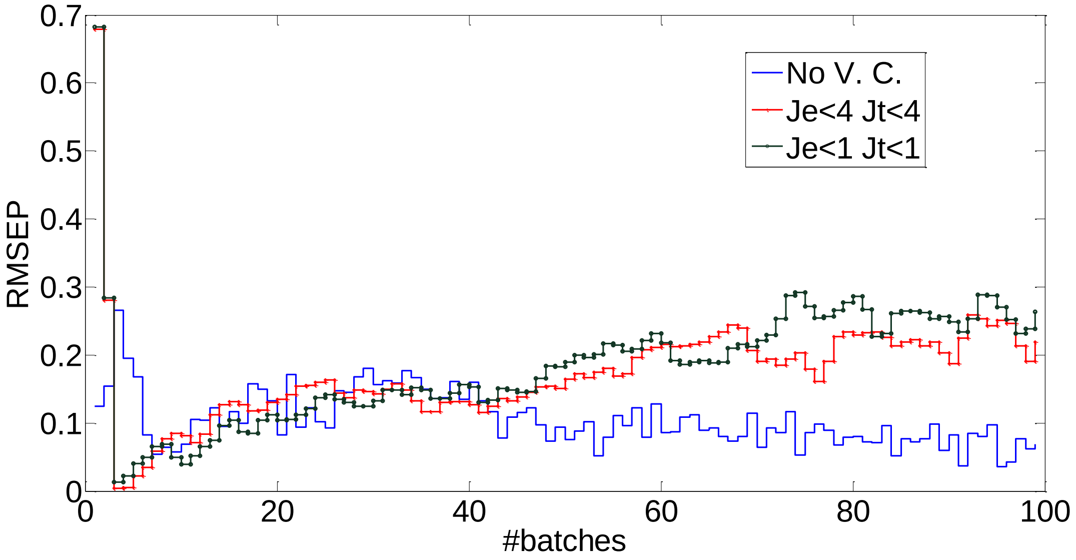


Figure 6.6: Pensim Root Mean Squared Error of Prediction after the MVT optimisation.

This could be solved by decreasing the number of LVs for identification. However, decreasing the number of LVs could cause a problem in the PMP methodology employed to estimate the future values in the QP optimisation, as it needs a significant number of LVs to have good predictions. A possible way to solve this problem is to consider the QP formulation using 2 models: one developed using PCA for the PMP estimation and a second developed using PLS for quality prediction. This was not investigated in this thesis and would be considered as an option for future developments.

6.4 Summary

In this chapter, validity constraints were introduced in to the batch-to-batch optimisation technique to ensure that the optimisation algorithm restricted its search space to the region where the MPLS was valid.

The relevant points from this chapter are listed next:

- In Section 6.1 the essential theory required to formulate the validity constraints extension in the QP formulation was described, such theory included: Statistical index, such as T^2 and SPE, confidence limits and validity constraints.

- In Section 6.2, validity constraints in the MVT optimisation from Chapter 5 were formulated based in the work published in [11]. This formulation included validity indicators as hard constraints in the QP problem.

- In Section 6.3, the introduction of validity constraints resulted in a slower rate of convergence and in some case a slight reduction in final yield. However, the constraints did improve consistency, which would be a major requirement for any industrial application.

Chapter 7: Smoothing in Batch-to-batch Optimisation

During the development of the main batch-to-batch optimisation approach, the performance was improved when PLS was combined with filtering techniques in the MVT. Therefore, this chapter details the investigation of a variety of smoothing techniques used in MPLS modelling. Firstly, section 7.1 introduces smoothing in MS and provides a brief explanation of the smoothing techniques used in this chapter. Section 7.2 then presents the predicted performance results for the two case studies formulated in this thesis using the different smoothing techniques. In section 7.3, results of the optimisation performance are presented. Finally, section 7.4 provides a brief summary of the relevant findings of this chapter.

7.1 Introduction

In MS process control, smoothing is often used to solve problems caused by the effect of random non-systematic noise within a process [24]. This can be achieved to a certain degree through the use of smoothing techniques, such as moving average filters, spline filters and Fourier analysis, among others. The main objective of smoothing in dynamic processes is to reduce the high frequency noise caused by external sources. In contrast, the low frequency noise often represents information from the process itself and is therefore more difficult to tackle. Moreover, Flores-Cerrillo and Macgregor [6] stated that the operation of batches may require smooth MVTs in order to reduce physical constraints in the actuator's MVT.

Identification methods, such as PLS or PCA, can reduce the noise effect by averaging the noise over several identification objects (batches). However, noise within the calibration data can create estimation errors in the regression vector defined by the model, and can therefore cause systematic errors in future predictions [96]. This can be negated by considering the addition of further smoothing techniques in the model identification.

Evidence of improvement in batch optimisation using smoothing techniques combined with MPLS models has previously been published [90], [102]. Filtering the process variables can significantly improve the yield obtained at the end of the batch; however, these results have yet to be sufficiently explained. In a recent article by Camacho et al. [103], a number of smoothing techniques were evaluated. These included post-filtering, Penalized Partial Least Squares (PPLS) and Functional Partial Least Squares (FPLS). These techniques were evaluated over a gradient batch-to-batch optimisation technique that was proposed in the same article. The findings indicated that improvements in batch optimisation were likely caused by a reduction in the algorithm search space rather than in the model prediction accuracy.

Using a similar analysis approach to that used by Camacho et al. [103], the investigation detailed in this chapter aimed to corroborate the findings of Camacho et al. through the use of the batch-to-batch optimisation technique formulated in this thesis. Sections 7.1.1 to 7.1.3 briefly describe the smoothing techniques evaluated.

7.1.1 Filtering

Many techniques exist to filter undesired information from a signal. Among these are the following [96]:

- Moving average filter: This is the simplest type of smoothing. This is achieved by replacing each variable sample with a weighted average of itself and its nearest neighbours.

- Spline filters: This is based on spline functions. This type of filter is built under the assumption that at small intervals most functions can be fitted by low degree polynomials. Therefore, they achieve smoothing by dividing the spectrum into sections and fitting polynomials to each section, under the condition that the resulting composite polynomial is a continuous function.

- Fourier analysis filters: The input signals are first transformed to the frequency space. The undesired frequencies are then weighted down and the inverse Fourier transform is used to regenerate the data in the input signals. Smoothing is achieved by multiplying each frequency factor by a weight, determining the frequency to which this should count.

The filter used for the experiments in this thesis consisted of a Finite Impulse Response (FIR) filter. This type of filter is based on the principles of the moving average and Fourier analysis filters. Equation 7.1 shows the output in discrete samples of a FIR filter.

$$\boldsymbol{filt}(n) = \sum_{i=o}^{C} \boldsymbol{c_i} \cdot \boldsymbol{x}(n-i) \tag{7.1}$$

where $\boldsymbol{x}(n)$ is the input signal, C is the filter order and $\boldsymbol{c_i}$ is the value of the filter coefficient at each discrete time *i*. The finite impulse response for this filter using Equation 7.1 can be thus defined in the frequency domain, as seen in Equation 7.2.

$$\boldsymbol{H}(\omega) = \sum_{i=o}^{C} \boldsymbol{c_i} \cdot \left(e^{j\omega}\right)^{-i} \tag{7.2}$$

where ω is the frequency in *(radians/sample)* and $e^{j\omega}$is Euler's formula for a complex number.

Smoothing is performed by choosing values from $\boldsymbol{c_i}$ in Equation 7.2 to obtain a response that achieves the desired frequency response. This can be achieved using numerical optimisation. In this thesis, the coefficients of $\boldsymbol{c_i}$ were identified using the FDA tool from Matlab to create a zero-phase low-pass FIR filter. Therefore, there was no filter delay. The cut-off frequency was chosen to be 10% of the maximum frequency (Nyquist frequency) of the input signal. Experimental results showed that there was little difference when the maximum frequency was varied between 3% and 10%.

7.1.2 Functional PLS

Functional PLS (FPLS) is based on the principles of spline filters. The main objective of FPLS is to alleviate the ill-posed nature of ordinary PLS, which often consists of non-smooth linear regression coefficients $\boldsymbol{\beta}$, due to the Wiener-Hopf equation. This is achieved by projecting the PLS model onto a set of B-spline basis functions [54]. B-splines are piecewise polynomial functions which can be expressed as a linear combination of other B-splines.

The estimation procedure for functional linear regression coefficients using PLS consists of two steps [100]:

1. The spline coefficients α are estimated from discrete observations such that the basis functions are appropriate to Equation 7.3.

$$\boldsymbol{X}(t) \approx \sum_{b=1}^{B} \alpha_b \cdot \boldsymbol{\phi_b}(t) \tag{7.3}$$

Where B is the number of nodes in the B-splines and $\boldsymbol{\phi_b}(t)$ are the basis functions. Using Equation 7.3, the linear regression can then be described by Equation 7.4.

$$\boldsymbol{Y} = (\boldsymbol{\Phi\alpha})^{\boldsymbol{T}}\boldsymbol{\beta} + \boldsymbol{e} \tag{7.4}$$

Where for all the B knots, $\boldsymbol{\alpha}$ is a vector containing the basis coefficients α_b of $\boldsymbol{X}(\boldsymbol{t})$, $\boldsymbol{\Phi}$ is the symmetric matrix of the inner products of the spline functions, and $\boldsymbol{\beta}$ is the vector containing the spline coefficients β_b for the linear regression, as shown in Equation 7.5.

$$\boldsymbol{\beta}(t) \approx \sum_{b=1}^{B} \beta_b \cdot \boldsymbol{\phi_b}(t) \tag{7.5}$$

2. The PLS regression of $\boldsymbol{Y}$ over $\boldsymbol{X}$ can then be reduced to the functional regression of $\boldsymbol{Y}$ over the matrix, as shown in Equation 7.6.

$$\boldsymbol{\Lambda} = \boldsymbol{\Phi}^{1/2}\boldsymbol{\alpha} \tag{7.6}$$

This approach was used in each object (row or batches) of the matrix of predictors $\boldsymbol{X}$ to obtain the results here marked as FPLS. This approach is consistent with that used to obtain the results presented in Camacho et al. [103].

7.1.3 Penalized PLS

PPLS smooths the linear regression coefficients by incorporating a multiplicative penalty term $\boldsymbol{L}$ to the optimisation criterion of the NIPALS algorithm for an univariate response vector $\boldsymbol{y}$ [101]. This is achieved by identifying the solution to the problem in Equation 7.7.

$$\arg\max_{\boldsymbol{w}} \frac{\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{y}\boldsymbol{y}^{\boldsymbol{T}}\boldsymbol{X}\boldsymbol{w}}{\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{w} + \boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{L}\boldsymbol{w}} \tag{7.7}$$

Using Lagrange multipliers, the penalty term was obtained (which was then applied to the iterative calculation of weights of each latent variable), by adding a matrix $\boldsymbol{C}$ into the NIPALS algorithm. This can be observed in Equation 7.8.

$$\mathbf{w}_{\boldsymbol{a}} = \frac{\boldsymbol{C}\boldsymbol{X}_{\boldsymbol{a}}^{\boldsymbol{T}} \cdot \boldsymbol{y}}{||\boldsymbol{C}\boldsymbol{X}_{\boldsymbol{a}}^{\boldsymbol{T}} \cdot \boldsymbol{y}||} \tag{7.8}$$

where a is the index corresponding to the $a\text{-}th$ latent variable and the matrix $\boldsymbol{C}$ is defined as in Equation 7.9.

$$\mathbf{C} = (\mathbf{I} + \mathbf{L})^{-\mathbf{1}} \tag{7.9}$$

In Equation 7.9, $\mathbf{I}$ is the identity matrix of size $(JK \times JK)$ and $\mathbf{L}$ is the penalty matrix expressed in Equation 7.10.

$$\mathbf{L} = \xi(\boldsymbol{D_{h-2}} \cdot \boldsymbol{D_{h-1}})^{\boldsymbol{T}} \cdot \boldsymbol{D_{h-2}} \cdot \boldsymbol{D_{h-1}} \tag{7.10}$$

Where ξ is the penalty coefficient, and the matrices $\boldsymbol{D_{h-2}}$ of size $(JK-2) \times (JK-1)$ and $\boldsymbol{D_{h-1}}$ of size $(JK-1) \times (JK)$ each correspond to the first order difference operator, as given by Equation 7.11.

$$\mathbf{D}_{\boldsymbol{h}} = \begin{bmatrix} 1 & -1 & \cdot & \cdot \\ \cdot & 1 & -1 & \cdot \\ \cdot & \cdot & \ddots & \vdots \\ \cdot & \dots & 1 & -1 \end{bmatrix} \tag{7.11}$$

In order to implement this approach, instead of the SIMPLS algorithm provided by Matlab, it was necessary to build and modify a kernel NIPALS algorithm as shown in Equations 7.8 to 7.11. This approach was used in order to obtain the results marked as PPLS in this thesis, and was also used to obtain the results presented by Camacho et al. [103].

7.2 Prediction performance from batch-to-batch

This section presents the results and analysis of the prediction performance of several smoothing techniques applied in the batch-to-batch optimisation scheme proposed in Chapter 5. The results were obtained from several experiments, which were performed in order to confirm reproducibility, using the same policy as that used in previous chapters. This policy is presented in Table 6.1. In addition, Table 7.1 describes the labels used in the graphs presented in this section.

Table 7.1: Label notation

Abbreviation	Description
PLS	These experiments performed the batch-to-batch optimisation without filtering the optimised MVT prior to the batch run.
PLS+F	These experiments performed the batch-to-batch optimisation filtering the optimised MVT prior to the batch run.
FPLS	These experiments performed the batch-to-batch optimisation using a Functional PLS model.
PPLS	These experiments performed the batch-to-batch optimisation using a Penalized PLS model.

The experiments described in this chapter used the same smoothing parameters as those described in Camacho et al. [103], using the penalty coefficient for the PPLS algorithm, namely ξ =1000, and a number of nodes equal to B = 20. Although these values were used for the purpose of comparison, they were able to be identified at the beginning of each 100 batches run.

The objective of the experiments described in this section was to determine whether the accuracy of the predictions made from one batch to the next was affected by the smoothing of the inputs of the MPLS model.

7.2.1 Prediction of case study: Saccha

The prediction accuracy of the adaptive MPLS model through batch-to-batch optimisation was assessed from the Root Mean Squared Error of Prediction

(RMSEP), as explained in Chapter 5. Therefore, the RMSEP of 100 experiments in 100 batches were obtained to observe the prediction response through the batch-to-batch run.

Figure 7.1 shows the RMSEP after the MVT optimisation for the three smoothing techniques (PLS+F, FPLS, PPLS) explained in section 7.1. This figure shows the RMSEP without the application of any filtering technique (PLS) to the optimisation.

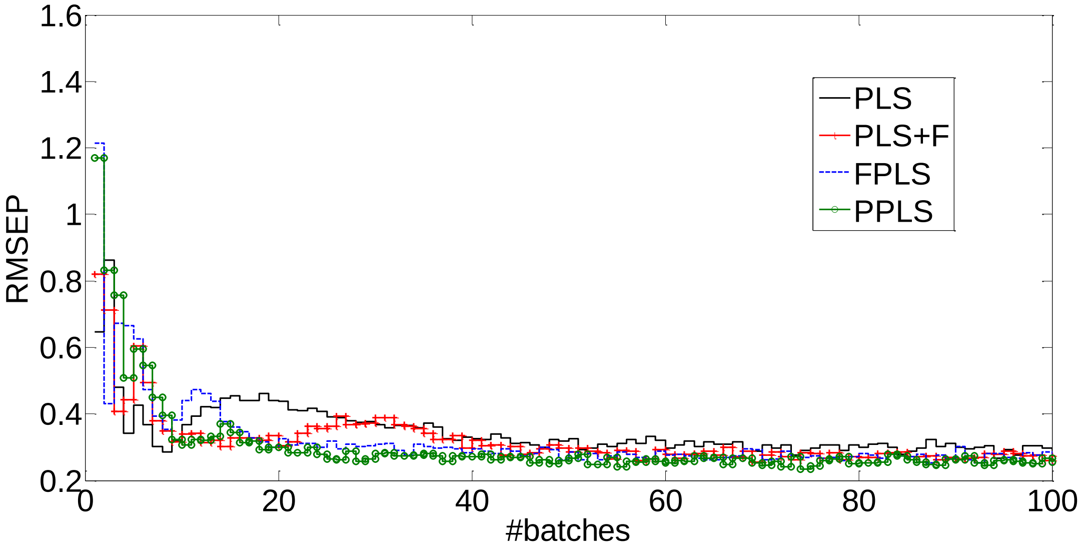


Figure 7.1: Saccha RMSEP after the MVT optimisation using different smoothing techniques

This figure illustrates the prediction accuracy of the MPLS model for each technique after the performance of MVT optimisation. As expected, the RMSEP fell as more batches were added to the model. The RMSEP for the four cases gradually decreased to values close to 0.2 g/l at the 100th batch, with a similar speed observed in all smoothing techniques.

An easier method with which to observe statistical differences in the RMSEP for each approach is that of using *boxplots*. This is a convenient way to observe the statistics of a collection of numerical data, as the spacing between the boxes indicates the degree of data dispersion.

Figure 7.2 shows the boxplot of the data from the RMSEP used in Figure 7.1. The central mark on each boxplot represents the median, and the edges of the box represent the 25th and 75th percentiles. The lines projecting out of the boxes, also known as *whiskers*, indicate the direction of the most extreme data points that were not considered to be outliers. Finally, the crosses represent the outliers not considered in the range bordered by the box edges.

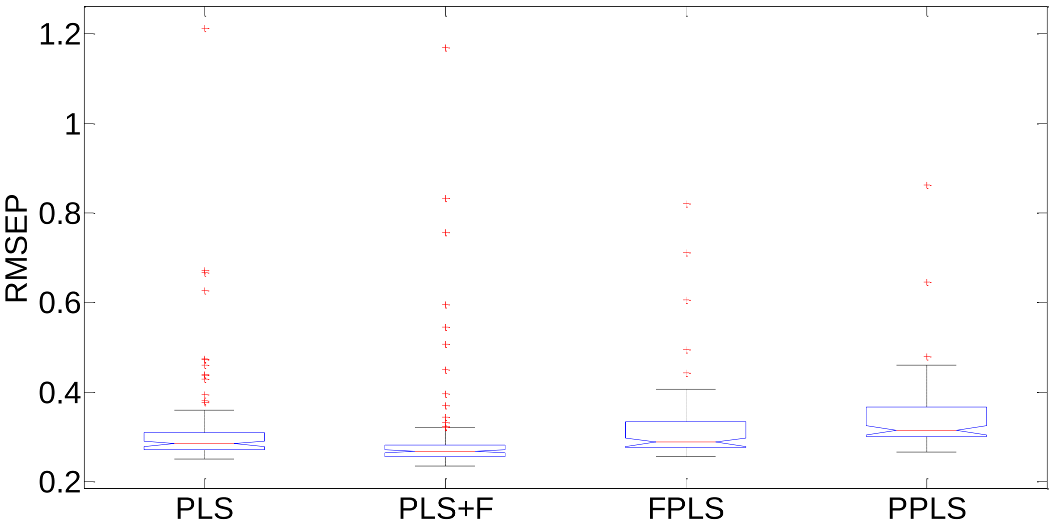


Figure 7.2 Boxplot of Saccha RMSEP after MVT optimisation using different smoothing techniques

The boxplot for each approach presented in Figure 7.2 shows a minor difference in the median of all smoothing techniques; however, the number of outliers using either the PLS or PLS+F techniques were much larger than observed in the FPLS and PPLS cases. A possible explanation for this is that the smoothed approaches within the PLS model (FPLS and PPLS) improved the accuracy of the prediction when the LV number was small. However, in the following section, it is observed that this was not always the case. In conclusion, there did not appear to be a significant difference in the prediction accuracy from one batch to the next using smoothing techniques alone in the Saccha simulator.

7.2.2 Prediction of case study: Pensim

Figure 7.3 shows the RMSEP after MVT optimisation in the three smoothing techniques and the non-smoothed approach for the Pensim simulator.

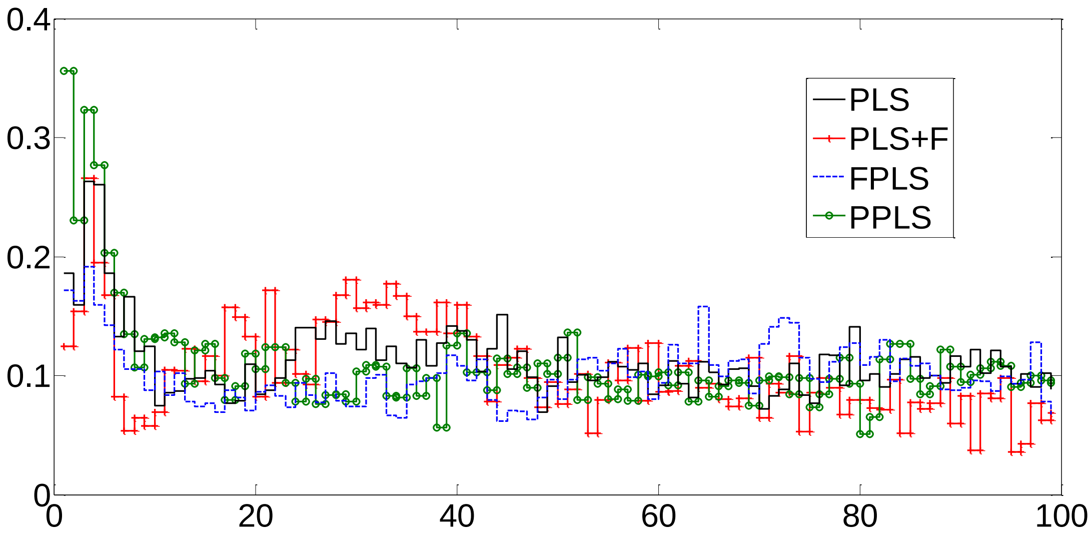


Figure 7.3: Pensim RMSEP after the MVT optimisation using different smoothing techniques

The RMSEP for the techniques under investigation (Figure 7.3) showed a similar response to the Saccha case. The accuracy gradually improved for each of the techniques, from RMSEP values close to 0.2 g/l in the commencing batch, to 0.1 g/l in the 100th batch. The PLS+F technique showed the greatest variation in response, yet also showed the lowest RMSEP at the end of the batch run.

Figure 7.4 shows a boxplot of the data from the RMSEP (Figure 7.3) for the different smoothing techniques.

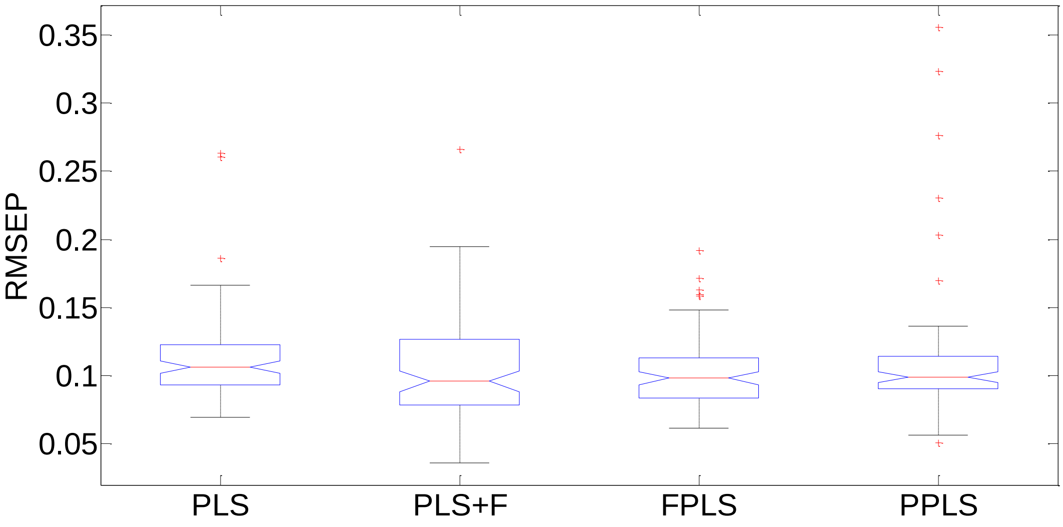


Figure 7.4 Boxplot of Pensim RMSEP after MVT optimisation using different smoothing techniques

Figure 7.4 shows little difference in the median of all smoothing techniques, as with the Saccha case; in contrast, the FPLS and PPLS techniques in the Pensim simulator had a significantly larger number of outliers than the PLS and PLS+F cases. Therefore, the accuracy of prediction in the initial batches likely depended to a greater extent on the case study than on the smoothing technique used in the batch-to-batch optimisation. Therefore, there did not appear to be a significant difference in the prediction accuracy from one batch to the next using smoothing techniques alone for the Pensim simulator.

7.3 Optimisation performance from batch-to-batch

This section presents the results of the batch-to-batch optimisation performance of several smoothing techniques. Relevant findings will also be discussed. The objective of these experiments was to observe how the numerical optimisation is affected by smoothing the inputs of the MPLS model. As in the previous chapter, this section is divided into two parts: the Saccha simulator results and the Pensim simulator results, presented in sections 7.3.1 and 7.3.2, respectively.

7.3.1 Optimisation of case study: Saccha

Figure 7.5 illustrates how the end-point penicillin concentration increased over 100 batches for the smoothing techniques and the non-smoothing approach used in the Pensim simulator.

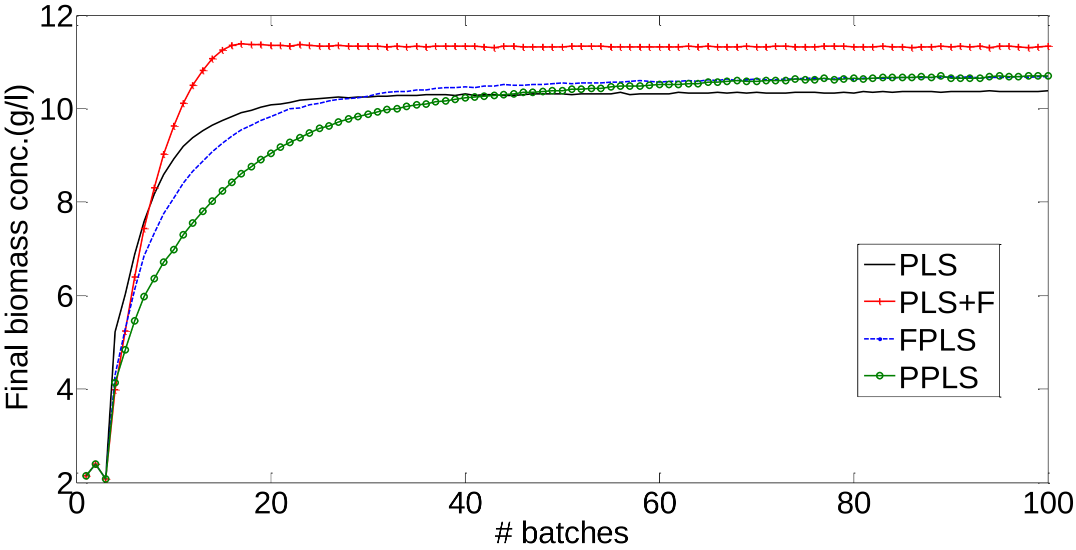


Figure 7.5 Final biomass concentration means using smoothing techniques in the Saccha simulator

The results in the figure above (Figure 7.5) show significant differences between the results of batch-to-batch optimisation and different smoothing techniques. The PLS+F technique obtained the highest yield, of approximately 2.0 g/l to 11.3 g/l. After approximately 15 batches, the yield stabilised at this output concentration. Subsequently, the FPLS and PPLS techniques obtained similar end-point concentrations, of approximately 10.8 g/l after approximately 60 batches. Finally, the non-smoothing PLS approach reached approximately 10.4 g/l by the 15th batch.

Table 7.2 provides the standard deviation means of the final five batches of the graphs shown in Figure 6.1. In addition, the smallest and largest values are indicated for each study case in green and red, respectively.

Table 7.2: Saccha standard deviations and means average of the final biomass concentration for the 5 last batches of Figure 7.5.

	PLS	PLS+F	FPLS	PPLS
Standard deviation	0.201	0.194	0.194	0.192
Mean	9.96	11.30	10.43	10.41

The results presented in Figure 7.5 and Table 7.2 show that the optimisation performances from batch-to-batch were likely influenced by the smoothness of the input MVT, and that the deviations were similar in all the smoothed cases. The slow convergence observed in the FPLS and PPLS smoothing techniques may have been caused by the penalty functions in the PPLS and the smoothing of all the predictor variables in the FPLS technique. The reason for this may be that the batch-to-batch optimisation scheme proposed in this thesis is highly dependent not only on the prediction accuracy of the quality but also on the predicted future values of the predictors. As the FPLS smooths these predictors, and as the penalty function of the PPLS techniques directly affects the weights of the predictors, the model is expected to predict the smoothed versions of the variables and not the variables themselves. However, it also appeared that having at least this degree of smoothness improved the optimisation more than the non-smoothing case overall.

7.3.2 Optimisation of case study: Pensim

The final penicillin concentration (when varying validity constraints) was applied to the Pensim simulator (Figure 7.6). This figure shows that the batch-to-batch optimisation strategy converged at similar speeds for each of the proposed approaches.

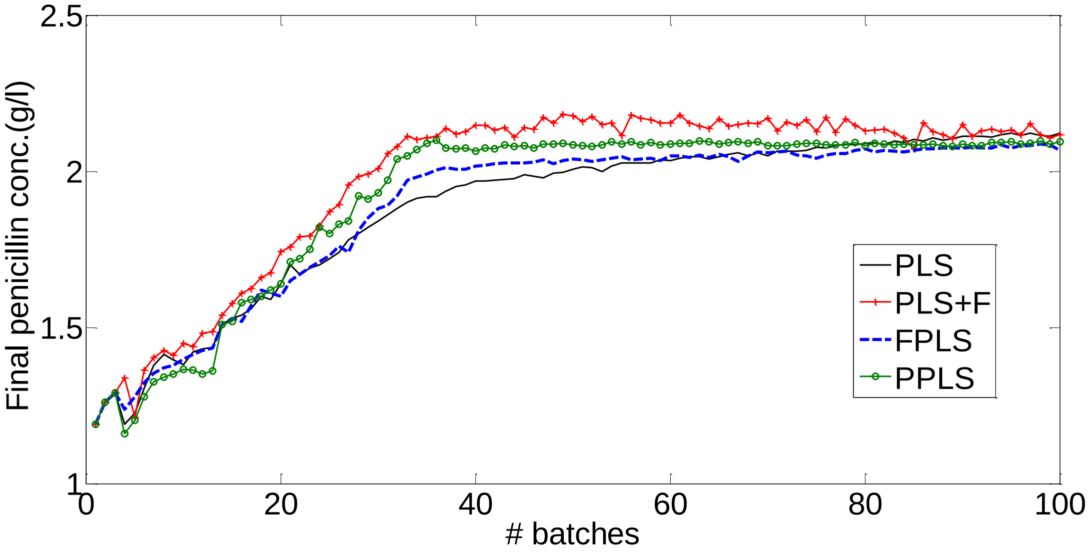


Figure 7.6 Final penicillin concentration means using smoothing techniques in the Pensim simulator

Figure 7.6 shows small differences in the results from the various smoothing techniques used in the Pensim simulator. As with the Saccha case, the PLS exhibited the slowest convergence rate and performance in general. However, in the Pensim simulations, the yield obtained at the end of the batch-to-batch run was approximately equal for each technique used. The FPLS and PPLS produced a slightly reduced yield of 2.08 g/l, compared to the 2.11 g/l produced by the PLS and PLS+F. The smoothing techniques (PLS+F, FPLS and PPLS) in general converged faster than the non-smoothing PLS technique used in this case study.

The standard deviations observed in the final five batches, as presented in the graphs in Figure 7.6, can be found in Table 7.3. In this figure, the deviations of the smoothing and non-smoothing techniques revealed a slight difference between all the smoothing approaches.

Table 7.3: Pensim standard deviations and means average of the final biomass concentration for the 5 last batches of Figure 7.6.

	PLS	PLS+F	FPLS	PPLS
Standard deviation	0.199	0.203	0.212	0.198
Mean	2.12	2.11	2.06	2.08

From the results presented in Figure 7.6 and Table 7.3, it can be concluded that the smoothing optimisation for both the Saccha and the Pensim cases appeared to have favourably affected, to some degree, the batch-to-batch optimisation methodology proposed in this thesis. A possible cause of this may have been the simplification in the search space for the QP optimisation function after smoothing. The final yield at the end of the 100 batches run was similar for all smoothing techniques used in the Pensim simulation, and significantly better in the Saccha simulation employing the PLS+F technique. The standard deviation did not appear to be affected by this improvement. This may indicate that the prediction accuracy of the model did not change considerably due to the smoothing techniques.

7.4 Summary

This chapter presented an investigation into the possible causes of the improvement observed after the introduction of MVT smoothing. This chapter also provided results and discussed several smoothing techniques as applied to the batch-to-batch optimisation proposed in Chapter 5.

The findings of this chapter are summarised as follows:

- The main theoretical framework for smoothing in batch optimisation was described in section 7.1. This included techniques such as FPLS, PPLS and FIR filtering of the MVT.

- Section 7.2 detailed several experiments performed to observe the effect of smoothing techniques on prediction accuracy in batch-to-batch optimisation. From the results of these experiments, it was clear that the MPLS model prediction accuracy did not significantly improve with the use of smoothing techniques.

- Similarly, section 7.3 presented results of the batch-to-batch optimisation of two case studies using smoothing techniques. However, in this section, the objective of these experiments was to examine the effect of different smoothing techniques in the batch-to-batch optimisation scheme. The results of this section exhibited significant improvements using smoothing techniques, especially in the Saccha simulator.

The case of the exceptional improvement in yield in the Saccha simulator during the initial batches using the PLS+F technique may have been due to the smoothness in the optimal feeding profiles leading to an improved numerical optimisation from the start of the batch-to-batch optimisation, as stated in Camacho et al. [103].

Chapter 8: Conclusions and Future Work

This chapter provides conclusions of the work presented in this thesis, organized according to the main subjects of research:

- Section 8.1 presents conclusions about the batch-to-batch optimisation scheme proposed in Chapter 5.
- Section 8.2 presents conclusions about the validity constraints included in the QP optimisation in Chapter 6.
- Section 8.3 presents conclusions about the smoothing techniques implemented in Chapter 7.

Finally, Section 8.4 suggests possible lines of research related to this work.

8.1 CONCLUSIONS OF BATCH-TO-BATCH OPTIMISATION

A novel batch-to batch optimization technique was proposed and implemented to two benchmark fed-batch fermentation simulators: Saccha and Pensim. Its performance, compared with other published work, showed that this methodology produced results with slightly higher end-point quality and the speed of convergence was considerably faster than alternative techniques

The technique consists of several stages developed for real manufacturing conditions:

- First, the process is identified, using MPLS, from a set of open-loop batch runs containing some form of excitation, a *golden batch* trajectory or even an initial guess at the optimal MVT; For the proposed technique to be applied to the simulated process investigated in this study, only 3 initial batches of data were required.

- This identified MPLS model is then used within a QP optimisation to calculate the trajectory of the MVT for the next batch. The objective function for this optimisation includes end-point quality as the target to minimise.

- The next batch is then run with the optimised MVT and the output data used to update the MPLS model. This allowed the productivity of the batch to be gradually increased from one batch to the next.

By using this strategy, approximately 15 further batches were required before the algorithm converged to an optimal MVT for the Saccha simulator and 40 batches for the Pensim simulator. The most important objective achieved in this design, was the improvement of the benefit/cost relationship of simulations that were designed to test the system under varying initial conditions. On the whole, this method is likely to be well suited for high cost and/or low quantity batch processes.

8.2 CONCLUSIONS OF VALIDITY CONSTRAINTS

Validity constraints were introduced in to the batch-to-batch optimisation technique to ensure the validity of predictions made by the MPLS model in the MVT optimisation. The introduction of these constraints resulted in general to a slower rate of convergence for the two case studies and in a considerable reduction of the yield in some cases for the Pensim simulation. However, the constraints slightly improved the consistency of predictions for some cases.

Mathematically the validity constraints were successfully included in the QP optimization; however, in numeric calculations, the use of the SPE index in the Pensim simulation constrained the QP optimization excessively and slowed considerably the convergence rate of the end-point quality. In summary, the constraints did reduce the standard deviation of the end-point quality measure and also reduced the prediction error under certain circumstances; however, the limits for the validity constraints would need to be adaptively tuned for each application in real manufacturing conditions.

8.3 CONCLUSIONS OF SMOOTHING TECHNIQUES

A selection of smoothing techniques (FPLS, PPLS and FIR filtering of the MVT) were successfully included in the batch-to-batch optimisation scheme and multiple experiments were carried out to observe the effect of smoothing techniques on the prediction accuracy and numerical optimisation from one batch to the next over two case studies.

From the results of these experiments, it was clear that the MPLS model prediction accuracy did not significantly improve with the use of smoothing techniques. However the batch-to-batch end-pint quality exhibited slight improvements using smoothing techniques, especially in the Saccha simulator. In general the results and conclusions presented in this thesis are consistent to those in [103]. Both investigations found the use of filtering techniques in the predictor variables to be beneficial in applications to general batch processes.

8.4 RECOMMENDATIONS FOR FUTURE WORK

The most promising directions of research are listed as follows:

- In this thesis, we choose PRBS as the input signal for system identification because it contains many distinct frequencies, in an attempt to make the input data informative of the dynamics relevant to the batch process. However, other alternatives should be analysed in future work to include methodologies such as the design of preconditioning data proposed in [120], [121]. The main reason for this recommendation is that it was observed that the initial data-set used for the calibration of the MPLS model considerably affected the yield for the batch-to-batch optimisation technique.

- Analyse the feasibility of the proposed optimisation technique for multiple quality outputs and inputs, in theory the algorithm still works for these systems, but some research must be done to assign the constraints and importance of each variable.

- Consider a numerical iterative scheme to calculate the optimal number of LVs to be used for model identification, when having a small number of batches in the initial calibration set, for the batch-to-batch optimisation methodology proposed in this thesis.

- Consider disturbance rejection applying the proposed scheme not at the beginning but at some intermediate time-point in the batch. Promising results were observed during some experiments done in the research used to write this thesis.

- Develop a multi-block model for the QP optimisation using PCA model for the estimation of future predictor values and a PLS model for quality prediction.

- Fit the designed adaptive methods to biomedical systems; the author in [82] suggest that the techniques of adaptive learning control could be highly exploitable in this field considering the repetitive nature of biological organisms. Some research has been presented in [122] and [123].

Appendix 1: Complete QP Formulation

The cost function used to optimise the MVT is formulated as:

$$\min_{\Delta\boldsymbol{u}}\left(\widehat{\boldsymbol{y}}-\boldsymbol{y}_{sp}\right)^{T}\left(\widehat{\boldsymbol{y}}-\boldsymbol{y}_{sp}\right)+\ \Delta\boldsymbol{u}^{T}\boldsymbol{M}\Delta\boldsymbol{u}$$
$$s.t.\begin{cases}\widehat{\boldsymbol{y}}=\left(\boldsymbol{\eta}\boldsymbol{V}+\Delta\boldsymbol{u}\boldsymbol{V}_{\boldsymbol{uf}}\right)\boldsymbol{Q}^{T}\\ \boldsymbol{lb}\leq\boldsymbol{u}_{\boldsymbol{n}}+\Delta\boldsymbol{u}\leq\boldsymbol{ub}\end{cases}$$

Substituting in the main cost function:

$$\min_{\Delta\boldsymbol{u}}\left(\left(\boldsymbol{\eta}\boldsymbol{V}+\Delta\boldsymbol{u}\boldsymbol{V}_{\boldsymbol{uf}}\right)\boldsymbol{Q}^{T}-\boldsymbol{y}_{sp}\right)^{T}\left(\left(\boldsymbol{\eta}\boldsymbol{V}+\Delta\boldsymbol{u}\boldsymbol{V}_{\boldsymbol{uf}}\right)\boldsymbol{Q}^{T}-\boldsymbol{y}_{sp}\right)+\ \Delta\boldsymbol{u}^{T}\boldsymbol{M}\Delta\boldsymbol{u}$$
$$s.t.\{\ \boldsymbol{lb}\leq\boldsymbol{u}_{\boldsymbol{n}}+\Delta\boldsymbol{u}\leq\boldsymbol{ub}$$

Expanding the previous equations yields to:

$$\min_{\Delta\boldsymbol{u}}\left(\boldsymbol{\eta}\boldsymbol{V}\boldsymbol{Q}^{T}+\Delta\boldsymbol{u}\boldsymbol{V}_{\boldsymbol{uf}}\boldsymbol{Q}^{T}-\boldsymbol{y}_{sp}\right)^{T}\left(\boldsymbol{\eta}\boldsymbol{V}\boldsymbol{Q}^{T}+\Delta\boldsymbol{u}\boldsymbol{V}_{\boldsymbol{uf}}\boldsymbol{Q}^{T}-\boldsymbol{y}_{sp}\right)+\ \Delta\boldsymbol{u}^{T}\boldsymbol{M}\Delta\boldsymbol{u}$$
$$s.t.\{\ \boldsymbol{lb}\leq\boldsymbol{u}_{\boldsymbol{n}}+\Delta\boldsymbol{u}\leq\boldsymbol{ub}$$

This is equal to:

$$\min_{\Delta\boldsymbol{u}}\left(\boldsymbol{\eta}\boldsymbol{V}\boldsymbol{Q}^{T}+\Delta\boldsymbol{u}\boldsymbol{V}_{\boldsymbol{uf}}\boldsymbol{Q}^{T}-\boldsymbol{y}_{sp}\right)^{T}\left(\boldsymbol{\eta}\boldsymbol{V}\boldsymbol{Q}^{T}+\Delta\boldsymbol{u}\boldsymbol{V}_{\boldsymbol{uf}}\boldsymbol{Q}^{T}-\boldsymbol{y}_{sp}\right)+\ \Delta\boldsymbol{u}^{T}\boldsymbol{M}\Delta\boldsymbol{u}$$
$$s.t.\{\ \boldsymbol{lb}\leq\boldsymbol{u}_{\boldsymbol{n}}+\Delta\boldsymbol{u}\leq\boldsymbol{ub}$$

Expanding again:

$$\begin{aligned}\min_{\Delta\boldsymbol{u}}\ & (\boldsymbol{\eta VQ}^{T})^{\boldsymbol{T}}\boldsymbol{\eta VQ}^{T} + (\boldsymbol{\eta VQ}^{T})^{\boldsymbol{T}}\Delta\boldsymbol{uV}_{\boldsymbol{uf}}\boldsymbol{Q}^{T} - (\boldsymbol{\eta VQ}^{T})^{\boldsymbol{T}}\boldsymbol{y}_{sp} \\ & + \left(\Delta\boldsymbol{uV}_{\boldsymbol{uf}}\boldsymbol{Q}^{T}\right)^{\boldsymbol{T}}\boldsymbol{\eta VQ}^{T} \\ & + \left(\Delta\boldsymbol{uV}_{\boldsymbol{uf}}\boldsymbol{Q}^{T}\right)^{\boldsymbol{T}}\Delta\boldsymbol{uV}_{\boldsymbol{uf}}\boldsymbol{Q}^{T} - \left(\Delta\boldsymbol{uV}_{\boldsymbol{uf}}\boldsymbol{Q}^{T}\right)^{\boldsymbol{T}}\boldsymbol{y}_{sp} - \left(\boldsymbol{y}_{sp}\right)^{\boldsymbol{T}}\boldsymbol{\eta VQ}^{T} \\ & - \left(\boldsymbol{y}_{sp}\right)^{\boldsymbol{T}}\Delta\boldsymbol{uV}_{\boldsymbol{uf}}\boldsymbol{Q}^{T} + \left(\boldsymbol{y}_{sp}\right)^{\boldsymbol{T}}\boldsymbol{y}_{sp} + \Delta\boldsymbol{u}^{T}\boldsymbol{M}\Delta\boldsymbol{u}\end{aligned}$$

$$s.t.\{\ \boldsymbol{lb} \le \boldsymbol{u_n} + \Delta\boldsymbol{u} \le \boldsymbol{ub}$$

Keeping the terms only relevant to $\Delta\boldsymbol{u}$:

$$\begin{aligned}\min_{\Delta\boldsymbol{u}}\ & (\boldsymbol{\eta VQ}^{T})^{\boldsymbol{T}}\Delta\boldsymbol{uV}_{\boldsymbol{uf}}\boldsymbol{Q}^{T} + \left(\Delta\boldsymbol{uV}_{\boldsymbol{uf}}\boldsymbol{Q}^{T}\right)^{\boldsymbol{T}}\boldsymbol{\eta VQ}^{T} \\ & + \left(\Delta\boldsymbol{uV}_{\boldsymbol{uf}}\boldsymbol{Q}^{T}\right)^{\boldsymbol{T}}\Delta\boldsymbol{uV}_{\boldsymbol{uf}}\boldsymbol{Q}^{T} - \left(\Delta\boldsymbol{uV}_{\boldsymbol{uf}}\boldsymbol{Q}^{T}\right)^{\boldsymbol{T}}\boldsymbol{y}_{sp} \\ & - \left(\boldsymbol{y}_{sp}\right)^{\boldsymbol{T}}\Delta\boldsymbol{uV}_{\boldsymbol{uf}}\boldsymbol{Q}^{T} + \Delta\boldsymbol{u}^{T}\boldsymbol{M}\Delta\boldsymbol{u}\end{aligned}$$

$$s.t.\{\ \boldsymbol{lb} \le \boldsymbol{u_n} + \Delta\boldsymbol{u} \le \boldsymbol{ub}$$

Grouping similar terms:

$$\min_{\Delta\boldsymbol{u}} \left(\Delta\boldsymbol{u}^{T}\boldsymbol{V_{uf}Q^{T}QV_{uf}^{T}}\Delta\boldsymbol{u}\right) + 2\left(\left(\boldsymbol{\eta VQ}^{T} - \boldsymbol{y_{sp}}\right)\boldsymbol{QV_{uf}^{T}}\right) + \Delta\boldsymbol{u}^{T}\boldsymbol{M}\Delta\boldsymbol{u}$$

$$s.t.\{\ \boldsymbol{lb} \le \boldsymbol{u_n} + \Delta\boldsymbol{u} \le \boldsymbol{ub}$$

Multiplying for ½:

$$\min_{\Delta \boldsymbol{u}} \frac{1}{2}\left(\Delta \boldsymbol{u}^T \boldsymbol{V}_{\boldsymbol{uf}} \boldsymbol{Q}^{\boldsymbol{T}} \boldsymbol{Q} \boldsymbol{V}_{\boldsymbol{uf}}^{\boldsymbol{T}} \Delta \boldsymbol{u}\right) + \left(\left(\boldsymbol{\eta} \boldsymbol{V} \boldsymbol{Q}^T - \boldsymbol{y}_{\boldsymbol{sp}}\right) \boldsymbol{Q} \boldsymbol{V}_{\boldsymbol{uf}}^{\boldsymbol{T}}\right) \Delta \boldsymbol{u} + \frac{1}{2} \Delta \boldsymbol{u}^T \boldsymbol{M} \Delta \boldsymbol{u}$$

$$s.t.\{\ \boldsymbol{lb} \leq \boldsymbol{u}_{\boldsymbol{n}} + \Delta \boldsymbol{u} \leq \boldsymbol{ub}$$

This gives the complete the QP formulation:

$$\min_{\Delta \boldsymbol{u}} \ \frac{1}{2} \Delta \boldsymbol{u}^T \boldsymbol{H} \Delta \boldsymbol{u} + \boldsymbol{f}^T \Delta \boldsymbol{u}$$

$$s.t. \begin{cases} \boldsymbol{H} = \boldsymbol{V}_{\boldsymbol{uf}} \boldsymbol{Q}^{\boldsymbol{T}} \boldsymbol{Q} \boldsymbol{V}_{\boldsymbol{uf}}^{\boldsymbol{T}} + \boldsymbol{M} \\ \boldsymbol{f}^T = \left(\boldsymbol{\eta} \boldsymbol{V} \boldsymbol{Q}^T - \boldsymbol{y}_{\boldsymbol{sp}}\right) \boldsymbol{Q} \boldsymbol{V}_{\boldsymbol{uf}}^{\boldsymbol{T}} \\ \boldsymbol{lb} \leq \boldsymbol{u}_{\boldsymbol{n}} + \Delta \boldsymbol{u} \leq \boldsymbol{ub} \end{cases}$$

References

[1] M. Barker, J. Rawtani, and S. Mackay, *Practical Batch Process Management*. Elsevier Ltd, 2005.

[2] D. Bonvin, B. Srinivasan, and D. Hunkeler, "Control and optimization of batch processes: improvement of process operation in the production of specialty chemicals," *IEEE Control Systems Magazine*, no. December 2006, pp. 34–45, Dec-2006.

[3] Y. Yabuki, T. Nagasawa, and J. F. Macgregor, "An industrial experience with product quality control in semi-batch processes," *Computers & Chemical Engineering*, vol. 24, pp. 585–590, 2000.

[4] F. Lei, M. Rotbøll, and S. B. Jørgensen, "A biochemically structured model for Saccharomyces cerevisiae.," *Journal of biotechnology*, vol. 88, no. 3, pp. 205–21, Jul. 2001.

[5] E. Korovessi and A. A. Linninger, *Batch Processes*. Taylor & Francis Group,LLC, 2006.

[6] 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, no. 5, pp. 539–553, Aug. 2004.

[7] J. Flores-Cerrillo and J. F. MacGregor, "Latent variable MPC for trajectory tracking in batch processes," *Journal of process control*, vol. 15, pp. 651–663, 2005.

[8] D. Laurí, J. Sanchis, M. Martínez, and a. Hilario, "Latent variable based model predictive control: Ensuring validity of predictions," *Journal of Process Control*, vol. 23, no. 1, pp. 12–22, Jan. 2013.

[9] J. Wan, O. Marjanovic, and B. Lennox, "Disturbance rejection for the control of batch end-product quality using latent variable models," *Journal of Process Control*, vol. 22, pp. 643–652, 2012.

[10] D. Laurí, J. a. Rossiter, J. Sanchis, and M. Martínez, "Data-driven latent-variable model-based predictive control for continuous processes," *Journal of Process Control*, vol. 20, no. 10, pp. 1207–1219, Dec. 2010.

[11] D. Laurí, B. Lennox, and J. Camacho, "Model predictive control for batch processes: Ensuring validity of predictions," *Journal of Process Control*, vol. 24, no. 1, pp. 239–249, 2014.

[12] T. F. Edgar, "Control and operations: When does controllability equal profitability?," *Computers and Chemical Engineering*, vol. 29, no. 1, pp. 41–49, 2004.

[13] P. Li, H. Garcia, G. Wozny, and E. Reuter, “Optimization of a semibatch distillation process with model validation on the industrial site,” *Industrial & Engineering Chemistry Research*, vol. 5885, no. 97, pp. 1341–1350, 1998.

[14] O. Abel, a. Helbig, W. Marquardt, H. Zwick, and T. Daszkowski, “Productivity optimization of an industrial semi-batch polymerization reactor under safety constraints,” *Journal of Process Control*, vol. 10, no. 4, pp. 351–362, Aug. 2000.

[15] S. Russell, D. Robertson, and J. Lee, “Control of product quality for batch nylon 6, 6 autoclaves,” *Chemical engineering*, vol. 53, no. 21, 1998.

[16] C. Hsu and J. Ward, “The best objective function for seeded batch crystallization,” *AIChE Journal*, vol. 59, no. 2, pp. 15–23, 2012.

[17] W. Wu, S.-Y. Lai, M.-F. Jang, and Y.-S. Chou, “Optimal adaptive control schemes for PHB production in fed-batch fermentation of Ralstonia eutropha,” *Journal of Process Control*, vol. 23, no. 8, pp. 1159–1168, Sep. 2013.

[18] H. Pham and G. Larsson, “Growth and energy metabolism in aerobic fed-batch cultures of Saccharomyces cerevisiae: Simulation and model verification,” *Biotechnology and Bioengineering*, vol. 60, no. 4, pp. 474–482, 1998.

[19] A. M. Jobé, C. Herwig, M. Surzyn, B. Walker, I. Marison, and U. von Stockar, “Generally applicable fed-batch culture concept based on the detection of metabolic state by on-line balancing.,” *Biotechnology and bioengineering*, vol. 82, no. 6, pp. 627–39, Jun. 2003.

[20] A. Cinar, “A modular simulation package for fed-batch fermentation: penicillin production,” *Computers &amp; chemical engineering*, vol. 26, pp. 1553–1565, 2002.

[21] R. K. Bajpai and M. Reuß, “A mechanistic model for penicillin production,” *Journal of Chemical Technology and Biotechnology*, vol. 30, no. 1, pp. 332–344, 1980.

[22] S. Goldrick, S. Andrei, D. Lovett, G. Montague, and B. Lennox, “The development of an industrial-scale fed-batch fermentation simulation,” *Journal of Biotechnology*, vol. 193, pp. 70–82, 2015.

[23] G. C. Paul and C. R. Thomas, “A structured model for hyphal differentiation and penicillin production using Penicillium chrysogenum,” *Biotechnology and bioengineering*, vol. 51, no. 5, pp. 558–572, 1996.

[24] T. E. Marlin, *Process Control: Designing Processes and Control Systems for Dynamic Performance.*, 2nd Editio. 2000.

[25] J. L. Marchetti, “Referential process-reaction curve for batch operations,” *AIChE Journal*, vol. 50, no. 12, pp. 3160–3168, Dec. 2004.

[26] P. Nomikos and J. MacGregor, “Monitoring batch processes using multiway principal component analysis,” *AIChE Journal*, vol. 40, no. 8, pp. 1361–1375, 1994.

[27] T. Kourti, P. Nomikos, and J. F. MacGregor, "Analysis, monitoring and fault diagnosis of batch processes using multiblock and multiway PLS," *Journal of Process Control*, vol. 5, no. 4, pp. 277–284, Aug. 1995.

[28] P. Nomikos and J. F. MacGregor, "Multi-way partial least squares in monitoring batch processes," *Chemometrics and Intelligent Laboratory Systems*, vol. 30, no. 95, pp. 97–108, Nov. 1995.

[29] C. K. Yoo, D. S. Lee, and P. A. Vanrolleghem, "Application of multiway ICA for on-line process monitoring of a sequencing batch reactor.," *Water research*, vol. 38, no. 7, pp. 1715–32, Apr. 2004.

[30] M. J. Willis, G. A. Montague, C. Di Massimo, M. T. Tham, and A. J. Morris, "Artificial neural networks in process estimation and control," *Automatica*, vol. 28, no. 6, pp. 1181–1187, Nov. 1992.

[31] K. Desai, Y. Badhe, S. S. Tambe, and B. D. Kulkarni, "Soft-sensor development for fed-batch bioreactors using support vector regression," *Biochemical Engineering Journal*, vol. 27, no. 3, pp. 225–239, Jan. 2006.

[32] J. Yu, "Multiway discrete hidden Markov model-based approach for dynamic batch process monitoring and fault classification," *AIChE Journal*, vol. 58, no. 9, pp. 2714–2725, Sep. 2012.

[33] K. Pearson, "On lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 1, pp. 559–572, 1901.

[34] H. Hotelling, "Analysis of a complex of statistical variables into principal components.," *British Journal of Educational Psychology*, vol. 24, pp. 417–520, 1932.

[35] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1–3, pp. 37–52, Aug. 1987.

[36] S. Wold, P. Geladi, K. Esbensen, and J. Öhman, "Multi-way principal components-and PLS-analysis," *Journal of Chemometrics*, vol. 1, no. 1, pp. 41–56, 1987.

[37] J. MacGregor and A. Cinar, "Monitoring, fault diagnosis, fault-tolerant control and optimization: Data driven methods," *Computers & Chemical Engineering*, vol. 47, pp. 111–120, Dec. 2012.

[38] T. Næs and H. Martens, "Principal component regression in NIR analysis: Viewpoints, background details and selection of components," *Journal of Chemometrics*, vol. 2, no. 2, pp. 155–167, Apr. 1988.

[39] S. Wold, A. Ruhe, H. Wold, W. J. Dunn, III, and W. D. III, "The collinearity problem in linear regression. The partial least squares (PLS) approach to generalized inverses," *SIAM Journal on Scientific and Statistical Computing*, vol. 5, no. 3, pp. 735–743, Sep. 1984.

[40] B. M. Wise, N. B. Gallagher, and W. Windig, *Chemometrics Tutorial for PLS _ Toolbox and Solo*. Wenatchee: Eigenvector Research, 2006.

[41] M. Otto and W. Wegscheider, “Spectrophotometric multicomponent analysis applied to trace metal determinations,” *Analytical Chemistry*, vol. 57, no. 1, pp. 63–69, Jan. 1985.

[42] P. Geladi and B. Kowalski, “Partial least-squares regression: a tutorial,” *Analytica chimica acta*, vol. 186, pp. 1–17, 1986.

[43] S. De Jong and S. de Jong, “SIMPLS: an alternative approach to partial least squares regression,” *Chemometrics and Intelligent Laboratory Systems*, vol. 18, no. 3, pp. 251–263, Mar. 1993.

[44] A. Alin, “Comparison of PLS algorithms when number of objects is much larger than number of variables,” *Statistical Papers*, pp. 711–720, 2009.

[45] S. Wold, N. Kettaneh-Wold, and B. Skagerberg, “Nonlinear PLS modeling,” *Chemometrics and Intelligent Laboratory Systems*, vol. 7, pp. 53–65, 1989.

[46] F. Yacoub and J. F. MacGregor, “Product optimization and control in the latent variable space of nonlinear PLS models,” *Chemometrics and Intelligent Laboratory Systems*, vol. 70, no. 1, pp. 63–74, Jan. 2004.

[47] K. Helland, H. Berntsen, and O. Borgen, “Recursive algorithm for partial least squares regression,” *Chemometrics and intelligent laboratory Systems*, vol. 14, no. 1–3, pp. 129–137, 1992.

[48] S. Qin, “Partial least squares regression for recursive system identification,” *Decision and Control, 1993., Proceedings of the*, vol. 00, no. 1988, pp. 2617–2622, 1993.

[49] B. S. Dayal and J. F. MacGregor, “Recursive exponentially weighted PLS and its applications to adaptive control and prediction,” *Journal of Process Control*, vol. 7, no. 3, pp. 169–179, Jan. 1997.

[50] S. Joe Qin, “Recursive PLS algorithms for adaptive data modeling,” *Computers & Chemical Engineering*, vol. 22, no. 4–5, pp. 503–514, Jan. 1998.

[51] C. Preda and G. Saporta, “Clusterwise PLS regression on a stochastic process,” *Computational Statistics and Data Analysis*, vol. 49, pp. 99–108, 2005.

[52] X. Li, X. Lu, J. Tian, P. Gao, H. Kong, and G. Xu, “Application of fuzzy c-means clustering in data analysis of metabolomics.,” *Analytical chemistry*, vol. 81, no. 11, pp. 4468–75, Jun. 2009.

[53] K. Faber and B. Kowalski, “Propagation of measurement errors for the validation of predictions obtained by principal component regression and partial least squares,” *Journal of chemometrics*, vol. 11, no. January 1996, pp. 181–238, 1997.

[54] B. McWilliams and G. Montana, “A PRESS statistic for two-block partial least squares regression,” *Workshop on Computational Intelligence (UKCI)*, no. Cv, pp. 1–6, 8–10, 2010.

[55] R. Ergon, “PLS score-loading correspondence and a bi-orthogonal factorization,” *Journal of Chemometrics*, no. 16, pp. 368–373, 2002.

[56] B. Li, J. Morris, and E. B. Martin, “Model selection for partial least squares regression,” *Chemometrics and Intelligent Laboratory Systems*, vol. 64, pp. 79–89, 2002.

[57] M. Forina, C. Casolino, and E. M. Almansa, “The refinement of PLS models by iterative weighting of predictor variables and objects,” *Chemometrics and Intelligent Laboratory Systems*, vol. 68, no. 1–2, pp. 29–40, Oct. 2003.

[58] R. Ergon, “Informative PLS score-loading plots for process understanding and monitoring,” no. 1, pp. 1–14, 2004.

[59] R. Rosipal and N. Krämer, “Overview and recent advances in partial least squares,” *Subspace, Latent Structure and Feature Selection*, pp. 34–51, 2006.

[60] J. Peng, S. Peng, and Y. Hu, “Partial least squares and random sample consensus in outlier detection.,” *Analytica chimica acta*, vol. 719, pp. 24–9, Mar. 2012.

[61] Z. Ge, Z. Song, F. Gao, and P. Wang, “Information-Transfer PLS Model for Quality Prediction in Transition Periods of Batch Processes,” *Industrial & Engineering Chemistry Research*, vol. 52, no. 15, pp. 5507–5511, Apr. 2013.

[62] M. Ronkko and J. Evermann, “A Critical Examination of Common Beliefs About Partial Least Squares Path Modeling,” *Organizational Research Methods*, vol. 16, no. 3, pp. 425–448, Mar. 2013.

[63] D. Laurí and B. Lennox, “Expanded PLS algorithm: Modeling of batch processes,” *Chemometrics and Intelligent Laboratory Systems*, vol. 132, no. MARCH 2014, pp. 111–123, 2014.

[64] S. J. Qin, “Survey on data-driven industrial process monitoring and diagnosis,” *Annual Reviews in Control*, vol. 36, no. 2, pp. 220–234, Dec. 2012.

[65] T. Clarke-Pringle and J. F. MacGregor, “Nonlinear adaptive temperature control of multi-product, semi-batch polymerization reactors,” *Computers & Chemical Engineering*, vol. 21, no. 12, pp. 1395–1409, Jan. 1997.

[66] N. Aziz, M. Hussain, and I. Mujtaba, “Performance of different types of controllers in tracking optimal temperature profiles in batch reactors,” *Computers & Chemical Engineering*, vol. 24, pp. 1069–1075, 2000.

[67] E. N. M. Van sprang, H.-J. -J Ramaker, J. A. Westerhuis, A. K. Smilde, and D. Wienke, “Statistical batch process monitoring using gray models,” *AIChE Journal*, vol. 51, no. 3, pp. 931–945, Mar. 2005.

[68] S. Zhang, F. Wang, D. He, and R. Jia, “Batch-to-batch control of particle size distribution in cobalt oxalate synthesis process based on hybrid model,” *Powder Technology*, vol. 224, pp. 253–259, Jul. 2012.

[69] G. M. Troup and C. Georgakis, “Process systems engineering tools in the pharmaceutical industry,” *Computers & Chemical Engineering*, vol. 51, pp. 157–171, Apr. 2013.

[70] J. Flores-Cerrillo and J. F. MacGregor, “Within-Batch and Batch-to-Batch Inferential-Adaptive Control of Semibatch Reactors: A Partial Least Squares

Approach," *Industrial & Engineering Chemistry Research*, vol. 42, no. 14, pp. 3334–3345, Jul. 2003.

[71] D. J. Kozub and J. F. MacGregor, "Feedback control of polymer quality in semi-batch copolymerization reactors," *Chemical Engineering Science*, vol. 47, no. 4, pp. 929–942, 1992.

[72] A. S. Kumar and Z. Ahmad, "Model Predictive Control (Mpc) and Its Current Issues in Chemical Engineering," *Chemical Engineering Communications*, vol. 199, no. 4, pp. 472–511, 2012.

[73] M. Morari and J. H. Lee, "Model predictive control: past, present and future," *Computers & Chemical Engineering*, vol. 23, no. 4–5, pp. 667–682, May 1999.

[74] J. Valappil and C. Georgakis, "State estimation and nonlinear model predictive control of end-use\nproperties in batch reactors," *Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148)*, vol. 2, no. 3, pp. 999–1004, 2001.

[75] J. Schäfer and A. Cinar, "Multivariable MPC system performance assessment, monitoring, and diagnosis," *Journal of Process Control*, vol. 14, no. 2, pp. 113–129, Mar. 2004.

[76] S. Perk, F. Teymour, and A. Cinar, "Adaptive Agent-Based System for Process Fault Diagnosis," *Industrial & Engineering Chemistry Research*, vol. 50, no. 15, pp. 9138–9155, Aug. 2011.

[77] B. Roffel and B. Betlem, *Process Dynamics and Control*, First. Wiley, 2006.

[78] B. Zhang, a. Abbas, and J. a. Romagnoli, "Automatic image-based estimation of texture analysis as a monitoring tool for crystal growth," *Chemometrics and Intelligent Laboratory Systems*, vol. 121, pp. 42–51, Feb. 2013.

[79] P. Fu and J. Barford, "Simulation of an iterative learning control system for fed-batch cell culture processes," *Cytotechnology*, vol. 10, pp. 53–62, 1992.

[80] J. Flores-Cerrillo and J. F. MacGregor, "Iterative Learning Control for Final Batch Product Quality Using Partial Least Squares Models," *Industrial & Engineering Chemistry Research*, vol. 44, no. 1, pp. 9146–9155, 2005.

[81] J. Zhang and J. Nguyan, "Batch to batch iterative learning control of a fed-batch fermentation process using linearised models," in *10th International Conference on Control, Automation, Robotics and Vision*, 2008, no. December, pp. 745–750.

[82] Y. Wang, F. Gao, and F. J. Doyle, "Survey on iterative learning control, repetitive control, and run-to-run control," *Journal of Process Control*, vol. 19, no. 10, pp. 1589–1600, Dec. 2009.

[83] J. Li, S. Jiping, and C. Min-Sen, "Neuro-fuzzy-based dynamic quadratic criterion-iterative learning control for batch process," *Transactions of the Institute of Measurement and Control*, vol. 35, no. 1, pp. 92–101, Dec. 2011.

[84] J. Jewaratnam, J. Zhang, J. Morris, and A. Hussain, "Batch-to-batch iterative learning control using linearised models with adaptive model updating," in

UKACC International Conference on Control, 2012, no. September, pp. 3–5.

[85] D. H. Owens and J. Hätönen, “Iterative learning control — An optimization paradigm,” *Annual Reviews in Control*, vol. 29, no. 1, pp. 57–70, Jan. 2005.

[86] S. Arimoto, S. Kawamura, and F. Miyazaki, “Bettering operation of dynamic systems by learning: A new control theory for servomechanism or mechatronics systems,” in *The 23rd IEEE Conference on Decision and Control*, 1984, pp. 1064–1069.

[87] T. Ao, X. Dong, and M. Zhizhong, “Batch-to-batch iterative learning control of a batch polymerization process based on online sequential extreme learning machine,” *Industrial & Engineering Chemistry …*, vol. 48, pp. 11108–11114, 2009.

[88] C. Li, J. Zhang, and G. Wang, “Batch-To-Batch Optimal Control of Batch Processes Based on Recursively Updated Nonlinear Partial Least Squares Models,” *Chemical Engineering Communications*, vol. 194, no. 3, pp. 261–279, Mar. 2007.

[89] K. S. Lee and J. H. Lee, “Iterative learning control-based batch process control technique for integrated control of end product properties and transient profiles of process variables,” *Journal of Process Control*, vol. 13, no. 7, pp. 607–621, Oct. 2003.

[90] J. Camacho, J. Pico, and A. Ferrer, “Self-tuning run to run optimization of fed-batch processes using unfold-PLS,” *AIChE journal*, vol. 53, no. 7, 2007.

[91] S. A. Vander Wiel, W. T. Tucker, F. W. Faltin, and N. Doganaksoy, “Algorithmic Statistical Process Control: Concepts and an Application,” *Technometrics*, vol. 34, no. 3, pp. 286–297, Aug. 1992.

[92] C. Ündey, S. Ertunç, and A. Çinar, “Online batch/fed-batch process performance monitoring, quality prediction, and variable-contribution analysis for diagnosis,” *Industrial & Engineering Chemistry Research*, vol. 42, pp. 4645–4658, 2003.

[93] R. Reiss, W. Wojsznis, and R. Wojewodka, “Partial least squares confidence interval calculation for industrial end-of-batch quality prediction,” *Chemometrics and Intelligent Laboratory Systems*, vol. 100, no. 2, pp. 75–82, Feb. 2010.

[94] M. Boiret, L. Meunier, and Y.-M. Ginot, “Tablet potency of Tianeptine in coated tablets by near infrared spectroscopy: model optimisation, calibration transfer and confidence intervals.,” *Journal of pharmaceutical and biomedical analysis*, vol. 54, no. 3, pp. 510–6, Feb. 2011.

[95] A. AlGhazzawi and B. Lennox, “Model predictive control monitoring using multivariate statistics,” *Journal of Process Control*, vol. 19, no. 2, pp. 314–327, Feb. 2009.

[96] H. Martens and T. Naes, *Multivariate Calibration*. Wiley, 1989.

[97] P. T. Reiss and R. T. Ogden, “Functional Principal Component Regression and Functional Partial Least Squares,” *Journal of the American Statistical*

Association, vol. 102, no. December 2014, pp. 984–996, 2007.

[98] S. Wold, “Spline Fucntions in Data Analysis,” *Technometrics*, vol. Vol. 16. pp. 1–11, 1974.

[99] W. Saeys, B. De Ketelaere, and P. Darius, “Potential applications of functional data analysis in chemometrics,” *Journal of Chemometrics*, vol. 22, no. 5, pp. 335–344, 2008.

[100] A. M. Aguilera, M. Escabias, C. Preda, and G. Saporta, “Using basis expansions for estimating functional PLS regression. Applications with chemometric data.,” *Chemometrics and Intelligent Laboratory Systems*, vol. 104, no. 2, pp. 289–305, 2010.

[101] N. Krämer, A. L. Boulesteix, and G. Tutz, “Penalized Partial Least Squares with applications to B-spline transformations and functional data,” *Chemometrics and Intelligent Laboratory Systems*, vol. 94, pp. 60–69, 2008.

[102] C. Duran-Villalobos and B. Lennox, “Iterative Learning Modelling and Control of Batch Fermentation Processes,” in *10th IFAC International Symposium on Dynamics and Control of Process Systems*, 2013, pp. 511–516.

[103] J. Camacho, D. Lauri, B. Lennox, M. Escabias, and M. Valderrama, “Evaluation of smoothing techniques in the run to run optimization of fed-batch processes with u-PLS,” *Journal of Chemometrics*, vol. 29, no. 6, pp. 338–348, 2015.

[104] H. Zhang and Z. Zhang, “Evolutionary optimization of a fed-batch penicillin fermentation process,” *Computer Communication Control and Automation*, pp. 403–406, 2010.

[105] H. Zhang and B. Lennox, “Integrated condition monitoring and control of fed-batch fermentation processes,” *Journal of Process Control*, vol. 14, no. 1, pp. 41–50, Feb. 2004.

[106] Y. Li, Z. Wang, and J. Yuan, “On-line fault detection using SVM-based dynamic MPLS for batch processes,” *Chinese Journal of Chemical Engineering*, vol. 14, no. 60574038, pp. 754–758, 2006.

[107] L. Yan and B. Lennox, “The application of nonlinear partial least square to batch processes,” *10th IFAC International Symposium on Dynamics and Control of Process Systems*, pp. 289–294, 2013.

[108] S. Stubbs, J. Zhang, and J. Morris, “Multiway Interval Partial Least Squares for Batch Process Performance Monitoring,” *Industrial & Engineering Chemistry Research*, vol. 52, no. 35, pp. 12399–12407, Sep. 2013.

[109] B. Srinivasan, C. J. Primus, D. Bonvin, and N. L. Ricker, “Run-to-run optimization via control of generalized constraints,” *Control Engineering Practice*, vol. 9, pp. 911–919, 2001.

[110] T. F. Edgar, D. M. Himmelblau, and L. S. Lasdon, *Optimization of chemical processes*, 2nd Editio. McGraw-Hill, 2001.

[111] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, *Optimization*, vol. 1. Elsevier, 1989.

[112] H. Martens, “Reliable and relevant modelling of real world data: a personal account of the development of PLS Regression,” *Chemometrics and Intelligent Laboratory Systems*, vol. 58, no. 2, pp. 85–95, Oct. 2001.

[113] J.-M. Lee, C. Yoo, and I.-B. Lee, “On-line Batch Process Monitoring Using Different Unfolding Method and Independent Component Analysis.,” *Journal of Chemical Engineering of Japan*, vol. 36, no. 11, pp. 1384–1396, 2003.

[114] F. Arteaga and A. Ferrer, “Dealing with missing data in MSPC: several methods, different interpretations, some examples,” *Journal of chemometrics*, pp. 408–418, 2002.

[115] H. Lin, “The Application of Multivariate Statistical Analysis and Batch Process Control in Industrial Processes (Doctoral thesis),” Universtity of Manchester, 2010.

[116] M. Stone, “Cross-Validatory Choice and Assessment of Statistical Predictions,” *Journal of the Royal Statistical Society*, vol. 36, no. 2, pp. 111–147, 1974.

[117] S. Wold, “Cross-validatory estimation of the number of components in factor and principal components models,” *Technometrics*, vol. 20, no. 4, pp. 397–405, 1978.

[118] P. Nomikos and J. F. Macgregor, “Multivariate SPC Charts for Batch Monitoring Processes,” *Technometrics*, vol. 37, no. 1, pp. 41–59, 1995.

[119] S. Russell, P. Kesavan, and J. Lee, “Recursive data-based prediction and control of batch product quality,” *AIChE journal*, vol. 44, no. 11, pp. 2442–2458, 1998.

[120] C. Rojas and J. Aguero, “Robustness in experiment design,” *IEEE Trans. Autom. Control*, vol. 57, no. 4, pp. 860–874, 2012.

[121] M. Deflorian and S. Zaglauer, “Design of experiments for nonlinear dynamic system identification,” in *IFAC 18th World Congress, Milano*, 2011, pp. 13179–13184.

[122] Y. Wang, H. Zisser, F. J. D. Iii, E. Dassau, and L. Jovanovic, “Model Predictive Control with Learning-Type Set-Point : Application to Artificial Pancreatic b-Cell,” *AIChE journal*, vol. 56, no. 6, pp. 1510–1518, 2010.

[123] Y. Wang, E. Dassau, and F. J. Doyle, “Closed-loop control of artificial pancreatic Beta -cell in type 1 diabetes mellitus using model predictive iterative learning control.,” *IEEE transactions on bio-medical engineering*, vol. 57, no. 2, pp. 211–9, Feb. 2010.