Data-Driven Modeling and Control of Batch and Continuous Processes using Subspace Methods

Data-Driven Modeling and Control of Batch and Continuous Processes using Subspace Methods

by

Nikesh Patel, B.Eng B.Biosci

A Thesis
Submitted to the School of Graduate Studies
in Partial Fulfillment of the Requirements for
the Degree Doctor of Philosophy

Doctor of Philosphy(2022)	McMaster University
(Chemical Engineering)	Hamilton, Ontario, Canada

TITLE:	Data-Driven Modeling and Control of Batch and Continuous Processes using Subspace Methods
AUTHOR:	Nikesh Patel, B.Eng B.Biosci
	(McMaster University, Hamilton, ON)
SUPERVISOR:	Dr. Prashant Mhaskar
NUMBER OF PAGES:	xxi, 198

LAY ABSTRACT

An important consideration of chemical processes is the maximization of production and product quality. To that end developing an accurate controller is necessary to avoid wasting resources and off-spec products. All advance process control approaches rely on the accuracy of the process model, therefore, it is important to identify the best model. This thesis presents two novel subspace based modeling approaches the first using first principles based constraints and the second handling missing data approaches. These models are then applied to a modified state space model with a predictive control strategy to show that the improved models lead to improved control. The approaches in this work are tested on both simulation (polymethyl methacrylate) and industrial (bioreactor) processes.

ABSTRACT

This thesis focuses on subspace based data-driven modeling and control techniques for batch and continuous processes. Motivated by the increasing amount of process data, data-driven modeling approaches have become more popular. These approaches are better in comparison to first-principles models due to their ability to capture true process dynamics. However, data-driven models rely solely on mathematical correlations and are subject to overfitting. As such, applying first-principles based constraints to the subspace model can lead to better predictions and subsequently better control. This thesis demonstrates that the addition of process gain constraints leads to a more accurate constrained model. In addition, this thesis also shows that using the constrained model in a model predictive control (MPC) algorithm allows the system to reach desired setpoints faster. The novel MPC algorithm described in this thesis is specially designed as a quadratic program to include a feedthrough matrix. This is traditionally ignored in industry however this thesis portrays that its inclusion leads to more accurate process control. Given the importance of accurate process data during model identification, the missing data problem is another area that needs improvement. There are two main scenarios with missing data: infrequent sampling/ sensor errors and quality variables. In the infrequent sampling case, data points are missing in set intervals and so correlating between different batches is not possible as the data is missing in the same place everywhere. The quality variable case is different in that quality measurements require additional expensive test making them unavailable for over 90% of the observations at the regular sampling frequency. This thesis presents a novel subspace approach using partial least squares and principal component analysis to identify a subspace model. This algorithm is used to solve each case of missing data in both simulation (polymethyl methacrylate) and industrial (bioreactor) processes with improved performance.

ACKNOWLEDGEMENTS

Firstly, I would like to start by thanking my supervisor Dr. Prashant Mhaskar. His guidance in both technical and personal matters helped to define the researcher and person I have become today. Without his constant support and willingness to explain concepts in numerous ways I would not have managed to accomplish as much as I have in these past seven years. In fact, without Dr. Mhaskar's belief in my abilities I would not be pursuing my doctoral degree in the first place. It was because of his unwavering faith in my abilities and his support of my career path that I got the amazing opportunity to teach at McMaster University. I have always regarded him as a mentor and a friend and I can never thank him enough for setting me on this path and for providing me with numerous opportunities to succeed. I have tried my best to learn everything I could from him and I hope to apply his principles in my own life moving forward. I would also especially like to thank Dr. Jake Nease who has been a true friend and mentor and perhaps the best squash player McMaster has seen in a while. Even before becoming a member of my committee and my supervisor Jake has always made time for me and was always willing to help me with any problems I had. Jake seamlessly sets an impossibly high bar both as an academic but also as a professor. Throughout me PhD as an instructor and while conducting research Jake was always there to guide me by holding me accountable to those very same standards and challenging me to be my best. I can honestly say that getting to know Jake changed my life and made me better both academically and as an individual. I want to thank my other committee member Dr. Fengjun Yan for his support and guidance along the way. I am grateful also for all the industrial collaborators I have had to help with my research. They have provided me with a lot of insight into the field of chemical engineering and helped me to develop my own research skills along the way. In particular, I would like to thank Dr. Brandon Corbett for his friendship and guidance these past few years. He went above and beyond his role as an instructor to help me with my research and to develop and test some the methods presented in my

thesis. His expertise and willingness to put my research first has helped me immensely. Finally, to the other instructors I have had that helped mold me and guided me to where I am today, thank you for your help. Even before I started graduate school and throughout my life, I would like to acknowledge the endless support provided by my parents and my brother. Their unconditional love and belief in me have helped me to keep going no matter how difficult life got. Without their understanding of the long hours and continuous work, this thesis would not have been possible. Their continued sacrifices to help me realize my dreams will never be forgotten. Finally, I would like to thank my loving fiancée Kishoree. Her willingness to listen to boring presentations and provide feedback and support throughout my PhD will always be appreciated. She has always been the person I counted on when things were difficult and she was the first person I want tell to share good news with. Thank you for always being there for me even when I did not know I needed it.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

As technological advancements have led to increases in automation and computational power the chemical engineering industry has also evolved. Traditionally, chemical processes have relied on first-principles based models to design small scale systems that can be converted to large scale industrial processes. While these approaches are effective, they are difficult to develop and often rely on assumptions and simplifications that fail to fully encompass the true dynamics. To that end, data-driven modeling has seen an increase in popularity as these techniques can analyze all the historical data, collected from different modes of operation, to develop a comprehensive model. data-driven modeling approaches take several forms and this thesis focuses on adapting well-studied subspace identification methods for modeling and control. The purpose of this section is to explain the basics behind batch subspace identification techniques along with the contributions made to the area of batch modeling and control starting with the definitions of batch processes. Batch processes are uniquely characterized by their finite duration, their predefined recipe, and their termination based on a process outcome. Batch dynamics differ from continuous processes in that they do not reach steady state conditions which represent the desired operating state for the process. Instead, batch processes focus on achieving a terminal product quality as the desired outcome. This poses a challenge for process control strategies because quality variables require additional testing and are not available as an "online" measurement. Online measurements are variables, such as temperature and pH, that can be measured continuously and can be used for control. Quality variables, which are not measured directly, are not used in process control strategies and so batch control relies on trajectory tracking approaches based on a successful trial. While input tracking can be simple to implement it does not utilize any process information or dynamics making these approaches unable to handle variations in initial conditions. To handle process variations and disturbances an advanced process control strategy

must be developed and implemented. In these strategies, such as model predictive control (MPC), a process model must first be developed. Thus, an accurate model is integral to an effective control strategy. To develop a model there are two main schools of thought: first-principles models and data-driven models. First-principles models rely on the scientific knowledge of the system and require intensive study of the process and expected interactions. These models are well explored however, they are difficult to develop and often rely on assumptions to simplify the equations. Given the abundance of historical data, data-driven modeling approaches, which rely on the mathematical relationships in the data, are a better alternative. While numerous data-driven modeling techniques exist, subspace identification is a natural choice due to its computational efficient algorithm and ability to generate a simple linear time invariant model capable of modeling both batch and continuous processes. Subspace identification also has drawbacks as relying only on the mathematical relationships in the training data can lead to overfitting and the matrix manipulations require full rank data. In consideration of the above data-driven modeling and control shortcomings, this thesis demonstrates novel improvements to subspace based techniques to improve the modeling and control of these processes. These approaches are applied to several industrially relevant processes such as seeded batch polymerization and continuous bioreactors. The remainder of this section provides a brief outline of each of the chapters and the contributions within.

1.2 Outline of the thesis

The purpose of the first manuscript is to address issues with data-driven modeling approaches identifying incorrect process trends. When attempting to identify a data-driven model through subspace identification the goal is to generate a linear time invariant model that minimizes the error in the training data. This is done through a

series of regression steps designed to identify a state trajectory and the system matrices of the model. However, in subspace identification, the states are a transformation of the physical states and as such, the first principles dynamics are not inherently present in the final model. This work introduces an approach that can introduce first principles knowledge into the data-driven model using constraints. The approach is particularly effective at correcting process gains that have been misidentified. In Chapter 3 the work on the constrained models is extended to focus on control strategies. Traditional state space model predictive control (MPC) approaches omit the feedthrough term when calculating the outputs. This is done since most industrial processes do not have outputs that are directly affected by the inputs. However, subspace identification identifies a feedthrough term that can capture unobserved effects and is necessary for improved control. This work firstly introduces a state space MPC algorithm that is capable of handling a feedthrough term and secondly shows the benefits of the feedthrough matrix. Chapter 4 presents a completely novel subspace model identification approach to handle the missing data problem. This approach is motivated by the need to handle scenarios where process data is unavailable from numerous problems such as sensor faults and different sampling rates. The key to this work lies in the close relationship between subspace identification techniques and regression based approaches of partial least squares (PLS) and principal component analysis (PCA). In particular, PCA and singular value decomposition result in identical solutions. This work demonstrates that the regression steps of subspace identification can be replaced by PCA and PLS steps using the non-iterative partial least squares (NIPALS) regression technique. To test the strength of the proposed approach a polymethyl methacrylate (PMMA) polymerization reactor is used as the test-bed. The missing data subspace model is compared against linear interpolation and mean replacement techniques in a receding horizon MPC approach. The simulation results show the improved performance of the missing data approach. In Chapter 5, an application of the missing data algorithm is presented using a bioreactor process

as the case study. The key challenges in the bioreactor problem are the large amounts of missing data due to different sampling rates and the discrete addition of glucose as an input. This paper provides a starting point to expand the missing data algorithms to industrial applications. Specifically, the bioreactor industry is one where advanced process control strategies are seldom used because any deviations from the plant-model mismatch can potentially lead to a lost batch due to the delicate nature of the cells. This work utilizes the missing data algorithm and state space MPC approach to model and control the bioreactor in order to improve batch quality. Chapter 6 presents all of the above techniques developed in this thesis to control a perfusion bioreactor for Sartorius. By extending work from previous chapters this chapter utilized constrained subspace identification techniques with the missing data algorithm to identify a subspace model of the bioreactor. The state space MPC is then used to control the bioreactor to maximize production and the control approach is currently being implemented at Sartorius. The last chapter is used to make concluding remarks and summarize the approaches developed in this thesis. Future work is also proposed that can be continued with other students under my supervision.

Chapter 2

Integrating Data-Driven Modeling with First-Principles Knowledge

This first chapter presents a novel subspace model identification approach to handle problems where data-driven modeling incorrectly identifies process trends such as the gain between certain inputs and outputs. The problem results when data-driven modeling approaches minimize the prediction error in the training set and overfit the model with incorrect process trends. The key idea in this work is to impose constraints based on first-principles knowledge in the model identification stage to generate an improved subspace model. Furthermore, an iterative process is utilized to ensure that the states of the model are representative of the process. This work was completed in collaboration with industrial partners Siam Aumi, Chris Ewaschuk and Jay Luo from Corning. They provided technical support and direction for the experiments and identification approach.

Patel, N., Nease, J., Aumi, S., Ewaschuk, C., Luo, J., & Mhaskar, P. (2020). Integrating Data-Driven Modeling with First-Principles Knowledge. *Industrial & Engineering Chemistry Research*, 59(11), 5103-5113

2.1 Abstract

This paper addresses the problem of integrating subspace based model identification with first principles modeling for handling scenarios where the subspace model identifies spurious relationships between inputs and outputs. The key motivation is to suitably synergize the two approaches while retaining the simplicity of subspace based model identification. In the proposed methodology, as is done with traditional subspace identification, state trajectories that best describe the input-output data are first computed (which implicitly correspond to an underlying linear time invariant model). In computing the system matrices using the state trajectories, constraints derived from first principles understanding, are incorporated into the optimization problem. To reconcile the resulting mismatch between the state trajectories and the system matrices, an iterative process is utilized. First, the system matrices computed from the optimization problem are utilized to re-estimate the state trajectories (this time utilizing a state estimator and the input and output trajectories). The state trajectories are, in turn, utilized to re-solve the system matrices using the input-output data. The process is repeated until convergence between successive state trajectories, thus yielding state trajectories and 'consistent' system matrices. The efficacy of the proposed approach is shown via simulations using a nonlinear process example.

2.2 Introduction

Models that capture and sufficiently represent the underlying dynamics of a process are critical for model-based control. Fundamentally, there are two modeling approaches available: first principles, mechanistic-based modeling, and empirical, data-driven modeling. First principles models are sought for their ability to capture inherent process dynamics. [15, 2, 10]These models rely on conservation equations,

such as mass, mole, and energy balances. However, they can be high-dimensional and add complexity, making them difficult to develop and maintain or use for control design. In many cases, the availability of historical process data has led to the development of data-driven models which are easier to develop and implement. One of the more prevalent techniques in data-driven modeling is the partial least squares (PLS) method [7]. In this approach, data from each experiment/process run is collected and projected onto a lower dimension (latent variable space) with guaranteed latent variable independence, where inferences about the nature of the process can be drawn [MacGregor et al.]. The data is oriented in a series of runs where columns represent measurements and rows represent observations. PLS models can be interpreted as time-dependent linear models around historical run trajectories and require special techniques to handle data from runs of non-uniform length. In PLS techniques, first principles knowledge can be accounted for by additional variables calculated using first-principles equations and appending the columns to the data matrix.

In addition to PLS, there are other types of data-driven modeling techniques such as prediction error methods (PEM) [16, 12, 21]. These methods solve the problem of minimizing the sum of square of the error between the predicted and measured outputs, to compute the system matrices. The PEM methods can readily incorporate additional constraints in solving the system matrices. The difficulty with PEMs lies in the computational complexity. The traditional PEM methods require solving a non-convex optimization problem to compute the system matrices, and successful model identification can rely on initial parameter values being sufficiently close to that of the solution.[12]. Unlike subspace identification which calculates states from the data, PEM methods must compute and compare the states of the system along with the predicted outputs as part of the optimization problem. The incorporation of first principles model based constraints can make the PEM based approaches even more computationally demanding.

Subspace identification, another model identification approach, on the other hand, is intrinsically more computationally tractable. [14, 8, 19] The method involves a two step procedure where the first step is to use data projection to identify a state trajectory and the second step is to compute the system matrices. Subspace identification algorithms have different techniques from canonical variate analysis[11], numerical algorithms for subspace state space identification [18]and multivariate output error state space algorithms [20]. These subspace identification algorithms can be classified by their use of singular value decomposition of matrices under different weightings schemes. Recent developments of the traditional subspace method have allowed for data from multiple experiments to be analyzed using the same singular value decomposition method.[4]

While some efforts have been made to incorporate first principles knowledge into subspace identification approaches by including constraints, these approaches are computationally expensive. One approach [1] proposes a constrained least squares solution while using a series of weighted constraints to turn the problem into a regular least squares solution. However, it is limited to equality constraints. Additionally, the method requires the system matrices to be solved in an intermediate step rather than being solved simultaneously[1], thus not necessarily resulting in system matrices (and the identified dynamics) being consistent with the constraints. Other approaches that consider prior information utilize parameter estimation techniques using a Bayesian framework to introduce steady state gains. [17] However, the constraints are proposed as soft constraints in the optimization problem. In another approach, the moment of the transfer function, i.e., the value of a point on the complex plane along with higher order derivatives, can be constrained using a weighted constraints approach and a quadratic optimization problem. This is done by using the Sylvester equation in conjunction with subspace identification. [9] In summary, limited formulations exist that enable incorporation of first principles knowledge explicitly as constraints, especially in the context of subspace based dynamic models.

Motivated by these considerations, this paper presents an identification method utilizing notions from subspace identification [3], but enabling incorporation of first principles knowledge. Compared to the existing subspace identification procedure, this work introduces a novel blending of first-principles based constraints in the subspace identification procedure to generate a constrained model. The integration of first principles knowledge in subspace identification demonstrates the ability to identify models that capture the underlying dynamics via data-driven modeling and reject spurious predictions as a result of incorporating first principles constraints. Our approach is to utilize first-principle knowledge of the process dynamics in order to add constraints to the subspace identification procedure. To achieve this, first, a nonlinear optimization problem is used to solve for the system matrices (via state estimates generated using an existing subspace identification technique) with constraints based on first-principles knowledge, hereafter referred to as the constrained model. Then, an iterative procedure is used to refine the resulting system matrices and reconcile them with the state trajectories. The rest of the paper is organized as follows: Section 2.3 presents two chemical stirred tank reactors (CSTR) in series as a motivating example, followed by an overview of subspace identification methods. Section 2.3.2 reviews the subspace identification approach that does not require the training data from multiple runs to be of uniform length. The proposed optimization based approach is then presented in Section 2.4.1 with the iterative algorithm described in Section 2.4.2. Section 2.4.3 presents the validation results. In Section 2.5, an application of the proposed approach to the CSTR example is presented, and the ability to incorporate first principles knowledge demonstrated. Finally, concluding remarks are made in Section 2.6.

2.3 Preliminaries

In this section we first present an example to motivate our results, followed by a review of existing subspace identification approaches.

2.3.1 Motivating Example: 2 Chemical Stirred Tank Reactors (CSTRs) in Series

Consider a process which has two CSTRs in series (see Figure 2.1) where the effluent from the first CSTR feeds into the second CSTR. Each CSTR has two inputs that can be manipulated, the fresh feed concentration and the amount of heat added, and two outputs that are measured: the effluent concentration and temperature. A first principles model describing the evolution of the concentration of A, C_{A_i}, and the reactor temperature, T_i, $i = 1, 2$ for each CSTR, results in the following 4 ordinary differential equations:

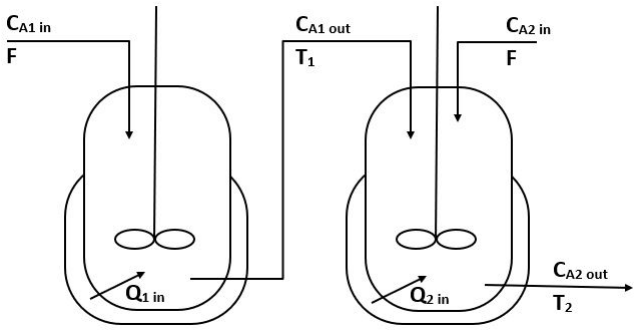


Figure 2.1: The schematic of the CSTR model

$$
\begin{aligned}
\frac{dC_{A1}}{dt} &= -\frac{F}{V}(C_{A1,in} - C_{A1,ss} + C_{A,0} - C_{A1}) - k_o e^{\frac{-E}{RT_1}} C_{A1in} \\
\frac{dT_1}{dt} &= -\frac{F}{V}(T_0 - T_1) - \frac{\Delta H}{C_p \rho} k_o e^{\frac{-E}{RT_1}} C_{Ain} + \frac{Q_{1in}}{C_p} \rho V \\
\frac{dC_{A2}}{dt} &= -\frac{F}{V}(C_{A2,in} - C_{A2,ss} + C_{A,0} - C_{A2}) - k_o e^{\frac{-E}{RT_2}} C_{A1in} + \frac{F}{V}(C_{A1} - C_{A2}) \\
\frac{dT_2}{dt} &= -\frac{F}{V}(T_1 - T_2) - \frac{\Delta H}{C_p \rho} k_o e^{\frac{-E}{RT_2}} C_{A2,in} + \frac{Q_{2in}}{C_p} \rho V + \frac{F}{V}(T_0 - T_2)
\end{aligned}
\tag{2.1}
$$

where ρ is the density of the fluid, F is the inlet flow rate to the first CSTR, V is the volume of each CSTR, R is the gas transfer coefficient, E is the activation energy of the reaction, k_0 is the value of the Arrhenius constant, T_0 is the inlet feed temperature, $C_{A,0}$ is the inlet feed concentration, ΔH is the heat of reaction, C_p is the heat capacity of the solution (see Table 2.1 for the parameter values). The input vector u includes $C_{A1,in}$,Q_{1in}, $C_{A2,in}$,Q_{1in} and the output vector includes y as C_{A1}, T_1, C_{A2}, T_2.

Table 2.1: Parameter values for the motivating process example

Parameter	Value	Unit	Parameter	Value	Unit
V	0.1	m^3	ρ	1000	kg/m^3
R	8.314	$J/(mol \cdot K)$	E	8.314×10^4	$\mu m/s$
$C_{A,0}$	2	mol/L	k_0	7.2×10^{10}	K
ΔH	4.78×10^4	kJ/kg	C_p	0.239	$kJ/K \cdot kg$
T_0	310	K	F	100	L/s

The process set up shown in Figure 2.1 is such that the inputs of the second CSTR do not effect the outputs of the first CSTR. Process noise and nonlinearities in the process can mislead existing subspace identification procedure into identifying non-casual relationships between the inputs of the second CSTR and the outputs of the first CSTR (see 2.3.3 section for a demonstration).

2.3.2 Subspace Identification

This subsection provides an overview of existing system identification methods that use iterative optimization algorithms involving the minimization of prediction errors. System identification methods rely on data from the measured outputs and manipulated inputs in order to generate a model of the system. Some of the more widely studied approaches that minimize prediction errors include maximum likelihood estimation (MLE) and expectation minimization (EM) methods[16, 12, 21]. These techniques are well-adapted to handle a wide range of model parameterizations including linear time invariant (LTI) models, linear regression models, and non-linear models.

Batch subspace identification techniques have also been widely studied as a way to identify a LTI state space model.[3, 5] The deterministic identification problem can be described as follows: If s measurements (where s represents the length of the data) of the input $u^{(b)}[k] \in \mathbb{R}^m$ and the output $y^{(b)}[k] \in \mathbb{R}^l$ are available for each run, then a model with order n can be identified in the following format:

$$\begin{aligned} \hat{\mathbf{x}}^{(b)}[k+1] &= \mathbf{A}\mathbf{x}^{(b)}[k] + \mathbf{B}\mathbf{u}^{(b)}[k], \\ \mathbf{y}^{(b)}[k] &= \mathbf{C}\hat{\mathbf{x}}^{(b)}[k] + \mathbf{D}\mathbf{u}^{(b)}[k], \end{aligned} \tag{2.2}$$

where the objective is to determine the order n of this unknown system and the system matrices $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times m}$, $\mathbf{C} \in \mathbb{R}^{l\times n}$, $\mathbf{D} \in \mathbb{R}^{l\times m}$.

We denote the measured outputs as $\mathbf{y}^{(b)}[k]$, where k is the sampling time from when the run is initialized and b denotes the run number. Thus, the Hankel matrix, after aligning each run b, is laid out as follows:

$$\mathbf{Y}_{1|i}^{(b)} = \begin{bmatrix} \mathbf{y}^{(b)}[1] & \mathbf{y}^{(b)}[2] & \cdots & \mathbf{y}^{(b)}\left[j^{(b)}\right] \\ \vdots & \vdots & & \vdots \\ \mathbf{y}^{(b)}[i] & \mathbf{y}^{(b)}[i+1] & \cdots & \mathbf{y}^{(b)}\left[i+j^{(b)}-1\right] \end{bmatrix} \quad \forall\, b = 1, \ldots, nb \tag{2.3}$$

where nb is the total number of runs used for identification and $\mathbf{Y}_{1|i}^{(b)}$ represents the output Hankel matrix for each run.

A single Hankel matrix by itself would not allow data from multiple experiments or runs to be utilized and the simple concatenation of the outputs from all of the runs would generate a data set where the initial condition of a subsequent run is the end point of the previous run which is also incorrect. Therefore, when concatenating the data it is important to generate a matrix where this assumption is not necessary to solve for the states. This can be achieved by horizontally concatenating the Hankel matrices from each run to generate our pseudo-Hankel matrix for both the input and output variables. This pseudo-Hankel matrix for the output data is defined as follows:

$$\mathbf{Y}_{1|i} = \begin{bmatrix} \mathbf{Y}_{1|i}^{(1)} & \mathbf{Y}_{1|i}^{(2)} & \cdots & \mathbf{Y}_{1|i}^{(nb)} \end{bmatrix} \tag{2.4}$$

Similarly, a pseudo-Hankel matrix for input data can be generated. A key consideration of this approach is that horizontal concatenation of data allows for runs of varying lengths to be identified without aligning the variables. The use of these pseudo-Hankel matrices for input and output data allows for data from multiple runs to be analyzed to compute the state trajectory using any subspace identification technique, such as the deterministic method used in this approach.[14] A consequence of horizontal concatenation is that the identified state trajectories also consist of horizontally concatenated state estimates from each run which can be represented as:

$$\hat{\mathbf{X}}_{i+1}^{(b)} = \begin{bmatrix} \hat{\mathbf{x}}^{(b)}[i+1] & \cdots & \hat{\mathbf{x}}^{(b)}\left[i+j^{(b)}\right] \end{bmatrix} \quad \forall\, b = 1, \ldots, nb \tag{2.5}$$

$$\hat{\mathbf{X}}_{i+1} = \begin{bmatrix} \hat{\mathbf{X}}_{i+1}^{(1)} & \hat{\mathbf{X}}_{i+1}^{(2)} & \cdots & \hat{\mathbf{X}}_{i+1}^{(nb)} \end{bmatrix} \tag{2.6}$$

where nb is the total number of training runs used for identification. Finally, once the state trajectory matrix is determined, the system matrices can be estimated using methods such as ordinary least squares as shown below:

$$\mathbf{Y}_{reg}^{(b)} = \begin{bmatrix} \hat{\mathbf{x}}^{(b)}[i+2] & \cdots & \hat{\mathbf{x}}^{(b)}\left[i+j^{(b)}\right] \\ \mathbf{y}^{(b)}[i+1] & \cdots & \mathbf{y}^{(b)}\left[i+j^{(b)}-1\right] \end{bmatrix} \tag{2.7}$$

$$\mathbf{X}_{reg}^{(b)} = \begin{bmatrix} \hat{\mathbf{x}}^{(b)}[i+1] & \cdots & \hat{\mathbf{x}}^{(b)}\left[i+j^{(b)}-1\right] \\ \mathbf{u}^{(b)}[i+1] & \cdots & \mathbf{u}^{(b)}\left[i+j^{(b)}-1\right] \end{bmatrix} \tag{2.8}$$

$$\begin{bmatrix} \mathbf{Y}_{reg}^{(1)} & \cdots & \mathbf{Y}_{reg}^{(nb)} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{X}_{reg}^{(1)} & \cdots & \mathbf{X}_{reg}^{(nb)} \end{bmatrix} \tag{2.9}$$

$$\mathbf{Y} = \theta \mathbf{X} \tag{2.10}$$

where the existing subspace identification approach would yield the **A**, **B**, **C** and **D** as the state-space model matrices and are henceforth collectively referred to as the unconstrained model.

Remark 1. *While the proposed approach utilizes a continuous process as a motivating example it is important to note that the proposed approach is applicable to numerous scenarios. The first scenario is where a continuous process is initiated from different initial conditions and operated until steady state (creating multiple runs). The second is where the process is operated as a continuous operation for a sufficiently long period of time while making appropriate step tests (thus producing a single run). Finally, the proposed approach is also set up to handle data from different batches (runs). To clearly*

convey the method's applicability to continuous and batch operations, the data sets are referred to as 'runs'.

Remark 2. *The key consideration with subspace identification for multiple data sets compared to a single data set is in the generation of the state trajectory. The risk of not using a pseudo-Hankel matrix structure through a concatenation of the data can result in a single state trajectory for the data set, where the initial point of the next run is incorrectly linked to the end point of the previous run. The subspace identification approach allows for the correct identification of the separate state trajectories from the training data to be used for model identification, thus enabling the usage of multiple runs during training.*

2.3.3 Current Modeling Limitations

This section illustrates how current modeling techniques can result in spurious predictions to provide a contrast with the proposed modeling technique presented later in the paper. To this end, we present modeling and validation results using the MATLAB system identification toolbox to fit first order plus dead time (FOPDT) models, and then using existing subspace identification techniques. The training set for the following case studies consisted of 10 runs where $s = 500$ minutes. The input sequence was a series of steps with each input move being held for 50 minutes. To simulate process variability, the initial condition of each run was varied over a range of temperatures (320 K to 350 K) and concentrations from (0.5 M to 3 M). The data was used to generate a transfer function model of the system with input delays chosen based on the best validation results. For the purpose of determining a subspace based model, the data is arranged into the Hankel matrix format described in section 2.3.2 (for the present application, the number of states $n = 4$ was chosen based on cross

validation; the problem of a more systematic method of choosing the number of states is considered elsewhere [6]).

To test the validity of the model, especially against spurious relationships, a step change was carried out where only the heat added to the second CSTR is changed (see Figure 2.3) . As expected, there is no change in the outlet concentrations and temperatures from the first CSTR. Both the FOPDT and subspace identification models, however, predict poorly in this scenario. The FOPDT model and subspace model predict a change in the outlet concentration and temperature and a new steady state for both variables (see Figure 2.2 for the process and predicted outputs and Figure 2.3 for the validation inputs). The result is not entirely unexpected. Note that the identification techniques 'fit' the process data presented to them at the training stage, and in an effort to capture the nonlinear dynamics as best as possible, could include relationship between variables that are inconsistent with the physical process. This provides the motivation behind crafting a data driven modeling framework that incorporates constraints at the model identification stage to ensure that known physical relationships are obeyed.

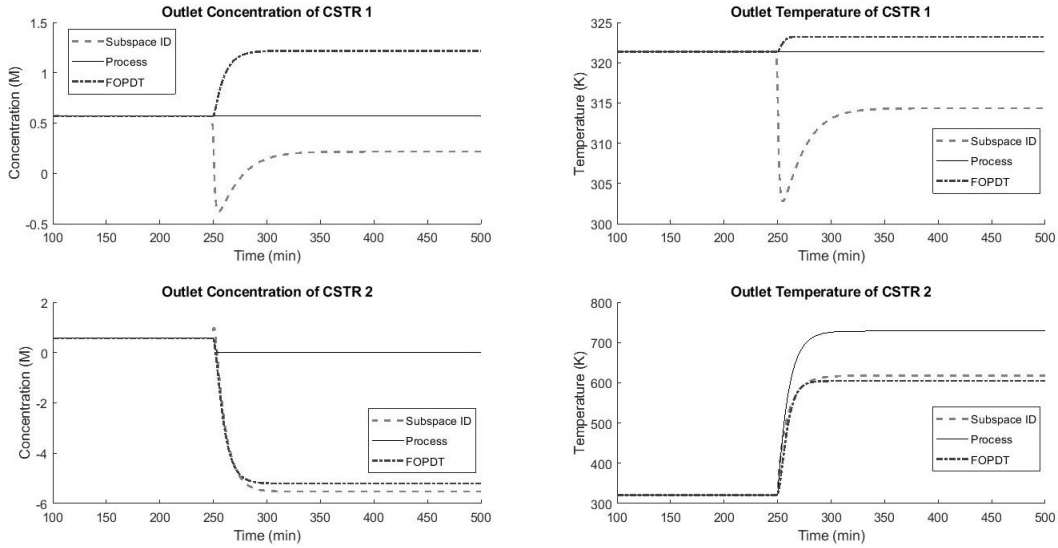


Figure 2.2: Step test results using the standard subspace identification techniques. The solid black line represents the process, the grey dashed line represents the predictions by the subspace model and the black dashed dot line represents the first order plus dead time model predictions.

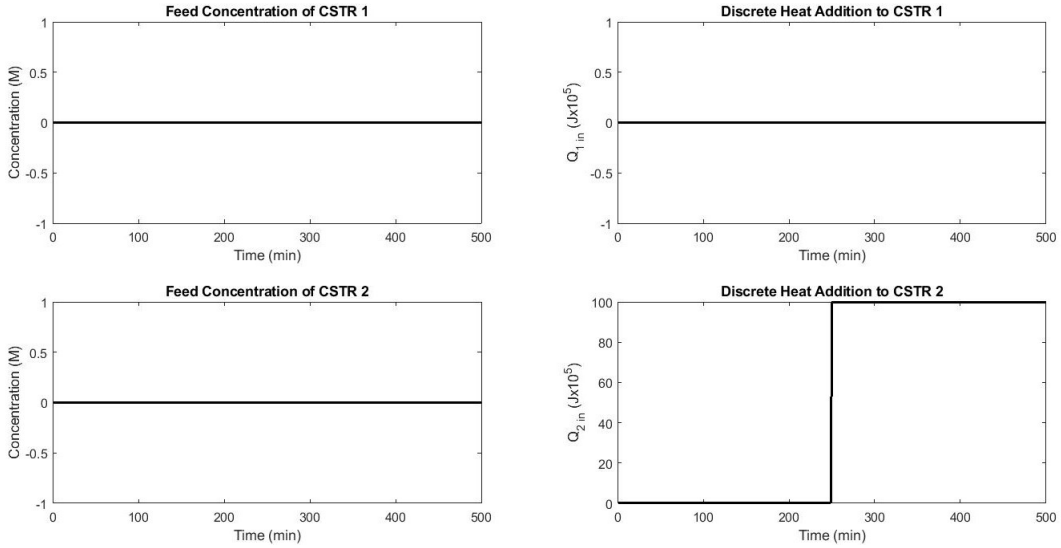


Figure 2.3: The input profiles used to generate the data to conduct targeted validation.

2.4 Proposed Modeling Approach

The first step in the proposed approach is to identify an LTI state space model of the system using data from historical runs. The state space model is found using a modified subspace identification algorithm that is able to handle multiple runs without aligning the variables.[3] This model is used as an initial guess in an optimization problem with the objective of minimizing the prediction error (similar to PEM approach) between the "constrained" model and the process while respecting first principles based constraints. The next step is to compare the new states of the constrained model with the original states and then iterating until the states converge to within a user-defined tolerance. The final converged model is referred to as the constrained model.

2.4.1 Optimization Problem Formulation

The state space model that has been identified using the subspace identification approach mentioned above is designed without first principles knowledge of the system. As illustrated in Section 2.3.3, the identification procedure can result in spurious predictions. The purpose of the constrained approach is to utilize first principles knowledge to identify a model that is consistent with what is physically realizable. One piece of the proposed approach involves solving a nonlinear optimization problem with appropriate constraints to determine the system matrices, given a state trajectory. A generalized formulation of the proposed optimization problem is shown below:

$$\begin{aligned} &\min_{\theta}(Y-\theta X)^T(Y-\theta X) \\ &s.t.\ \textit{First principles knowledge based constraints} \end{aligned} \tag{2.11}$$

$$\theta = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \tag{2.12}$$

where Y and X are described in Equation 2.10, θ is comprised of the system matrices, and the first principles based knowledge is imposed as appropriate constraints in the optimization problem. This specific constraint can take many forms, including steady state gains, dynamic gains, or other means of characterizing known/expected behavior between variables.

This nonlinear optimization problem utilizes the system matrices identified through traditional subspace approaches as an initial guess. Note that the solution of the optimization problem yields a set of system matrices that are computed using the state trajectories determined from the standard subspace identification procedure. Thus, while the system matrices respect the constraints, they are not necessarily 'consistent' with the state trajectories. The next section presents an iterative procedure to generate a consistent set of system matrices and state trajectories.

Remark 3. *The optimization problem, depending on the nature of the first principles knowledge, could very well be a non-convex optimization problem (as in the case of the illustrative example). Thus, any solution is not guaranteed to be the global solution. However, a global optimum, while desired, is not necessary to ensure the incorporation of the first principles knowledge. Any potential local solution that respects the constraints enables the incorporation of the first principles knowledge. Note that the implementation is also 'warm*

started' using the solution from the standard subspace identification approach. This work utilized MATLAB's fmincon solver to solve the optimization problem however, any other appropriate nonlinear programming solver can readily be used.

Remark 4. *With regard to the complexity of the optimization problem, it is important to note that a direct implementation of the prediction error minimization model would require the system matrices along with the entire state trajectory to be computed (thus not only increasing the number of decision variables, but also requiring more nonlinear constraints). In particular, the system matrices along with the inputs and outputs determining the state trajectory result in a highly nonlinear optimization problem. Such an optimization would take the following form (written in pseudocode for the case of a single run):*

$$\begin{aligned} &\min_{\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D},\mathbf{x_0}} \sum_{k=1}^{s} ||y[:,k] - \hat{y}[:,k]||^2 \\ &\quad s.t.\ \textit{First principles knowledge based constraints} \end{aligned} \tag{2.13}$$

$$\begin{aligned} x[:,k+1] &= Ax[:,k] + Bu[:,k] \\ \hat{y}[:,k] &= Cx[:,k] + Du[:,k] \\ x[:,1] &= x_0 \end{aligned} \tag{2.14}$$

where input $u \in \mathbb{R}^{m\times s}$ and the output $y \in \mathbb{R}^{l\times s}$ are available and the decision variables are the system matrices $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times m}$, $\mathbf{C} \in \mathbb{R}^{l\times n}$, $\mathbf{D} \in \mathbb{R}^{l\times m}$ and the initial value of the states x_0. The subsequent predicted output matrix $\hat{y} \in \mathbb{R}^{l\times s}$ and state matrix $x \in \mathbb{R}^{n\times s}$ comes from solving Equation 6 for $k = 1..s$.

The above optimization problem is significantly more nonlinear (the decision variables are multiplied to each other, and raised up-to the power of N, where N is the number of samples in the run), than the proposed, iterative algorithm described in the next section. The proposed method offers a trade-off between calculating the complete solution (as could be done with a modified PEM implementation) and completely ignoring the first principles based constraints (as with the subspace based identification approaches, including the recently proposed approaches for handling data from multiple runs).

2.4.2 Iterative algorithm

Presently, the way Problem 11 is solved is as follows: The given data is first utilized to solve the traditional subspace identification problem- this yields the system matrices and the state trajectories, that are consistent with each other. Problem 11 then **utilizes** these state trajectories, and the system matrices as initial guesses, and computes a new set of system matrices that respect the constraints. In doing so, however, the computed system matrices are no longer 'consistent' with the state trajectories- thus an iterative procedure is followed.

The key idea of the proposed approach is to compute system matrices that are consistent with the constraints. While that is seemingly achieved via the constraints in the optimization problem, the optimization problem utilizes state trajectories in the computations, which are in turn determined by the (unconstrained) subspace identification approach. Next we describe an algorithm as shown in Figure 2.4 that enables achieving a set of system matrices and state trajectories that are consistent.

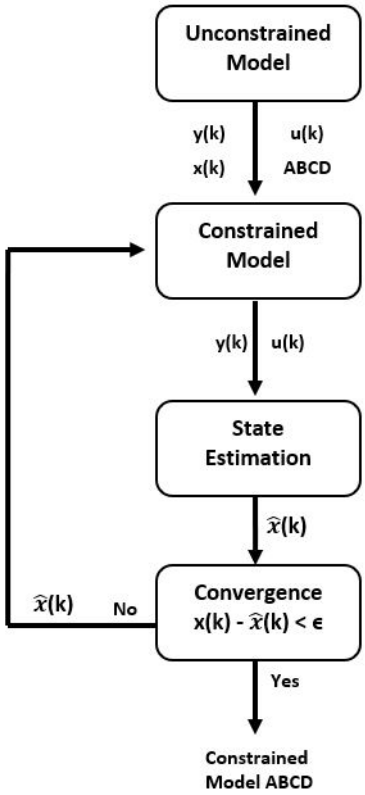


Figure 2.4: The Iterative algorithm used to converge the state sequence

State Convergence Algorithm

1. Identify the states of the system and an unconstrained model by utilizing a subspace identification technique.[14]

2. Using the state trajectory and an appropriate initial guess (the unconstrained model the first time, and the previous solution in subsequent iterations) for the optimization problem proposed in 2.4.1, identify a new constrained model.

3. Use the newly computed system matrices, and the input output data to compute a new state sequence, using say a Luenberger observer, but the comparison is carried out only for corresponding state values, past the time point of convergence (see Section 2.4.3 for details on the observer).

4. Compare the new states estimated using the constrained model with the states at the previous iteration at the corresponding time points (take the absolute value of the difference and then divide it by the length of the new state sequence) and, if they have not, use the new state sequence and iterate (go back to step 2) until the variation is less than a prescribed tolerance. Note that the state sequence shrinks during each iteration based on the time it takes the state observer to converge. However, the new states are directly compared to their corresponding states in the original sequence. For example, if the original state sequence is estimated at 100 time points and the observer takes 2 iterations to converge, the state sequence from time points 3-100 is compared with the new state vector.

5. Once the state sequences have converged, the resultant system matrices are denoted as the final constrained model.

Remark 5. *For the simple cases where the first principles based constraints are linear in the decision variables, the optimization problem reduces to a convex optimization problem, ensuring a globally optimal solution. In such cases, the convergence of the algorithm can be readily guaranteed. For application purposes, the algorithm is terminated after a predetermined set of iterations. In the present applications, this number was chosen as 30, with the algorithm converging well within this threshold for all the test cases. Note that the incorporation of the physics based constraints do not require the iterative algorithm to converge (at every step of the iteration, the constrained model that is generated respects the additional constraints): the convergence enables 'consistency' between the identified state trajectory and the identified model.*

Remark 6. *An important modeling parameter to consider further is the number of states- both for the constrained and unconstrained model. Specifically, there could be scenarios where the analysis of the constrained model reveals*

that using the same number of states, albeit with the physics based constraints incorporated, either renders some of the states redundant, or necessitates more states. In the present implementation, the number of states determined during the standard identification procedure is retained for the rest of the identification procedure. Future implementations would explore further the choice with the number of states (and specifically, using the constrained model identification to guide the choice).

2.4.3 Model Validation

Model validation is the key step in model identification, given that success with validation scenarios ultimately enable confidence in the developed model. In state-space models, where the initial state for a new data set is not know *apriori*, it is imperative to first implement a state observer and determine a sufficiently accurate state estimate so the prediction capability of the model can be evaluated. In this work, a Luenberger observer is used. Specifically, during the initial part of a new dataset, the Luenberger observer is used to find a good estimate of the current system states before using the state space model. The observer has the following form:

$$\hat{\mathbf{x}}[k+1] = \mathbf{A}\hat{\mathbf{x}}[k] + \mathbf{B}\mathbf{u}[k] + \mathbf{L}(\mathbf{y}[k] - \hat{\mathbf{y}}[k]) \tag{2.15}$$

where $\mathbf{L}$ is the observer gain and is chosen to ensure that $(\mathbf{A} - \mathbf{LC})$ is stable. The initial state estimate could be chosen as zero, but for this work the state estimate for each run generated from the subspace identification approach was used as the initial state for the run. Once the observer has converged, the identified model can be utilized for predicting the remainder of the trajectory.

Remark 7. *While the present illustration utilizes the Luenberger observer to estimate the states of the system (appropriate for the noise-free illustrative results in the present manuscript), in principle any observe can be utilized. In certain scenarios, such as noisy process data, other state estimation techniques such as the Kalman filter might be more applicable. Furthermore, the key idea of the algorithm is not restricted to the particular subspace identification algorithm. Any other subspace identification algorithm can be used to 'warm start' the optimization problem with the key tool being the iterative algorithm that makes the state trajectory consistent with the constrained system matrices without resorting to having to solve a complex optimization problem.*

Remark 8. *This work focuses on applying first principles knowledge that is known with 'certainty', such as steady state gains having a known value, or certain input-output channels being zero. The approach is not designed to handle situations where the 'first principles knowledge' itself has parameters that need to be identified (for instance, if it includes a reaction rate constant that needs to be determined). Handling such situations can be done through an alternative hybrid modeling approach where the two models are identified and implemented in parallel. [6].*

2.5 Application To The Motivating Example

In this section, first, the identification of a state space model for the two stage CSTR process using the novel subspace identification approach is discussed. This is followed by a comparison against the previously developed subspace identification procedure.

2.5.1 Model Identification

The constrained optimization problem for the motivating example takes the following form:

$$\begin{aligned} &\min_{\theta}(Y-\theta X)^T(Y-\theta X) \\ &\theta = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \\ &G_\theta = tf(A,B,C,D) \\ &G(1,3) = 0 \\ &G(1,4) = 0 \\ &G(2,3) = 0 \\ &G(2,4) = 0 \end{aligned} \tag{2.16}$$

where Y and X are described in Equation 2.10, θ is comprised of the system matrices, G_θ denotes the equivalent transfer function representation of the state-space system, denoted by $tf(A,B,C,D)$, with $G(i,j)$ denoting the transfer function from input i to output j. Thus, physical constraints are included which require that the inputs from the second CSTR do not affect the outputs from the first CSTR. This is accomplished by computing an equivalent transfer function model from the state space model and for the transfer functions between inputs 3 and 4 and outputs 1 and 2, constraining the numerator to be zero.

Note that the optimization problem is nonlinear and non-convex as presented. It is important to recognize that the non-convexity arises from the constraints and not the objective function. In particular, since the pre-computed state trajectory (using the standard subspace identification method) is utilized in formulating the objective

function, the objective function is quadratic in the decision variables and hence convex. In contrast, if the state trajectory was unavailable, and the initial state vector needed to be a decision variable, the objective function itself would be a highly nonlinear function of the decision variables (the positive tradeoff would be that it would result in the computation of an entirely consistent state trajectory and system matrices). Thus, a full blown PEM optimization problem would have a similar form as in Equation 2.16, with the following key additions: The optimization would also include x_0, the initial state value as the decision variable. In computing the objective function, the predicted outputs at time k would be computed as follows:

$$\begin{aligned} X_k &= A^k x_0 + \sum_{i=0}^{k-1} A^{k-(i+1)} B u_i \\ Y_k &= C X_k + D u_k \end{aligned} \tag{2.17}$$

Thus, even in the absence of the first principles based constraints, the optimization problem would be highly nonlinear and non-convex, making it difficult to compute the solution. The additional first principles based constraint, if included with the full optimization problem, could render the optimization intractable. In contrast, the proposed formulation provides a practical trade-off between computational complexity and solution accuracy in incorporating first principles based constraints in the system identification approach.

We next describe the implementation of the iterative algorithm. In the first step of the algorithm, where the standard subspace identification is utilized to generate a state sequence for input output data with $s = 500$, the state sequence that is generated is of a length of 491. Using this state sequence, the optimization problem defined above is solved to compute the 'constrained' system matrices, and using the first state value as an initial guess, a Luenberger observer is run. Recall that the system matrices, now being computed with the constraints in place, are no longer 'consistent' with the state trajectory, hence the output as predicted by the Luenberger observer

contains an error. However, the Luenberger observer converges rapidly (after 2 time steps), yielding a state trajectory of length 489 time points. This state sequence is then compared against the original states at the corresponding time points to verify whether they are within tolerance. In the present implementation, the error was not within tolerance after the first iteration and the algorithm reverted to computing a new set of 'constrained' system matrices and iterating until convergence is achieved, which took 5 iterations. The length of the state sequence from each iteration is listed in Table 2 and the state error is shown in Figure 2.5.

Table 2.2: Length of the state sequence

Iteration	State Sequence
0	491
1	489
2	487
3	486
4	481
5	478

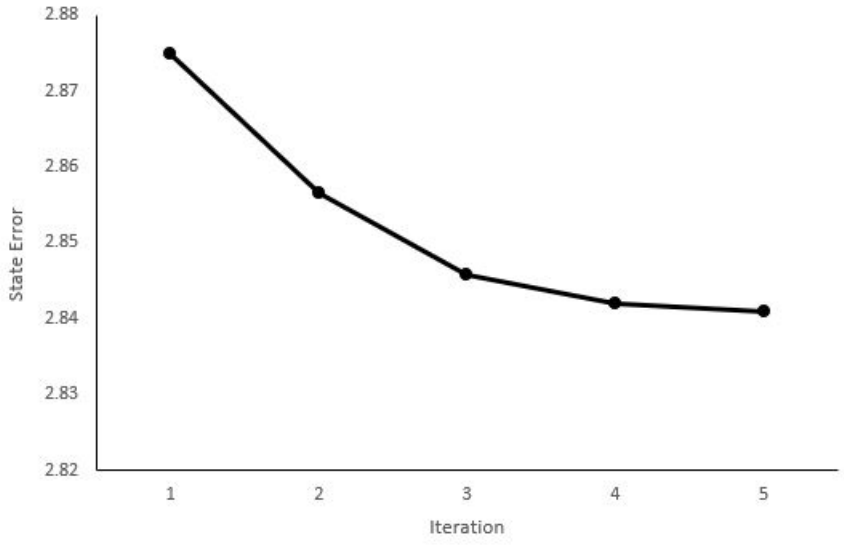


Figure 2.5: The normalized state error over the course of the iterations

2.5.2 Simulation Results

Figure 2.6 shows the training input profile for one of the runs used to identify the subspace model (the input data for validation has a similar profile). The training results using the standard subspace identification approach [14] and a FOPDT model are shown in Figure 2.7. The standard approach fits the true process reasonably well. The FOPDT model fit is not as accurate given the nature of the model structure.

The real test is for the targeted validation sequence, where only one of the inputs in the second CSTR is changed with the others held constant, (see Figure 2.8 where the heat input to the second CSTR is increased). The resulting output profiles are shown in Figure 2.9. The key plots are the top two, which show the process outputs from the first CSTR continue to be unchanged, as desired. The constrained model predictions do not change even after the step change in the input to the second CSTR. In contrast, the unconstrained model and FOPDT model predict a change in the outlet variables from the first CSTR. This is a result of the models incorrectly identifying the transfer functions between the inputs to the second CSTR and the outputs of the first CSTR. The bottom two plots in Figure 2.9 show the prediction in the variables that are expected to change (outlet variables from the second CSTR). All the models demonstrate plant-model mismatch. This is expected due to process nonlinearity however, the proposed approach still captures the dynamics of the process known with certainty, that is, the inputs to the second CSTR should not have any effect on the first CSTR.

To quantify the model effectiveness, the root mean sum errors (RMSE) between each of the scaled model outputs, $\hat{y}$, and the process outputs, y, is recorded along with the percent difference between the model errors.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)}{n}} \tag{2.18}$$

For the targeted validation results, the constrained model error was 1.8770 while the unconstrained model error was 2.5352, resulting in a difference of -25.96%. The solution time for the unconstrained model was significantly faster in comparison to the constrained approach as seen in Table 2.3. The experiment was carried out using a 4th Gen Intel Core i5-4300M processor and 4GB of memory.

Table 2.3: Solution Times For Model Identification

Case	Time(s)
Unconstrained	162.74
Constrained	21.754

The error is calculated to be consistent with how the predictions are generated, i.e., after using a Luenberger observer to identify the current state of the model and then using the estimated states and future inputs to predict the future outputs. Note that a negative percent error refers to the constrained model having a lower error than the unconstrained model as calculated by the following equation:

$$\%\ Error = \frac{constrained\ model\ error - unconstrained\ model\ error}{unconstrained\ model\ error} * 100 \tag{2.19}$$

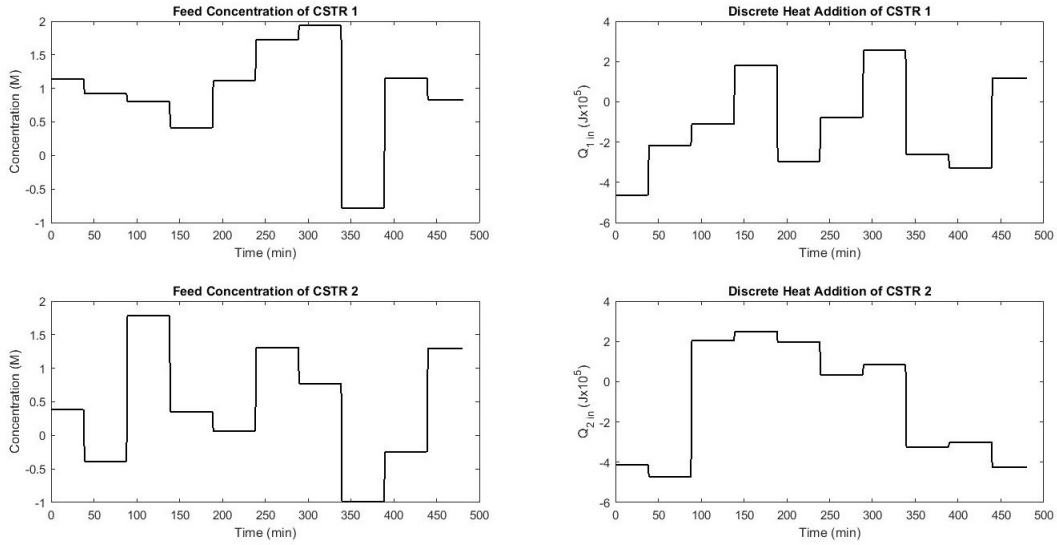


Figure 2.6: The input profiles that are used to generate the training data.

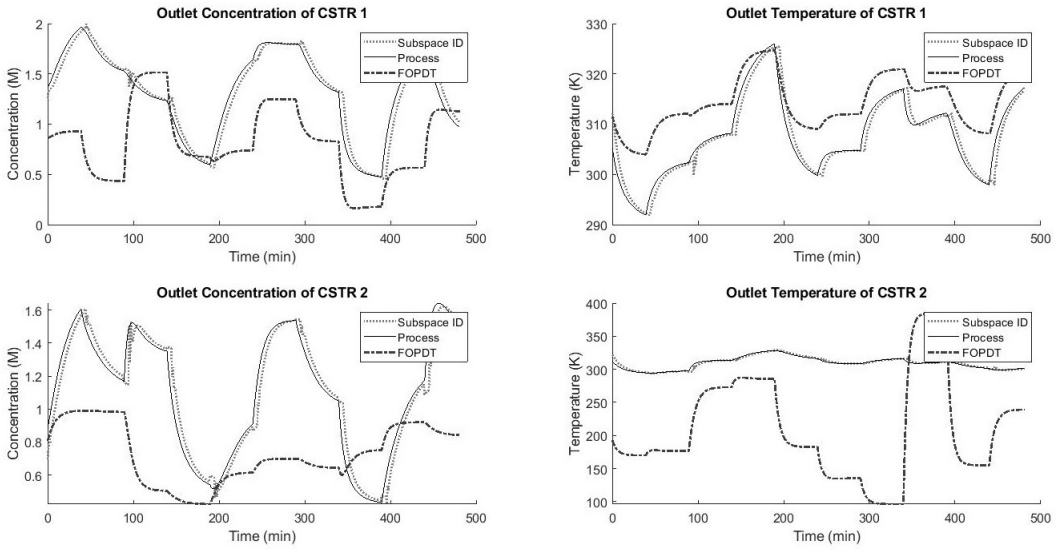


Figure 2.7: The figures shows the training fit for each output. In each figure the dashed dot line shows the FOPDT model fit, the grey dotted line is the traditional subspace approach and the solid black line is the measured outputs.

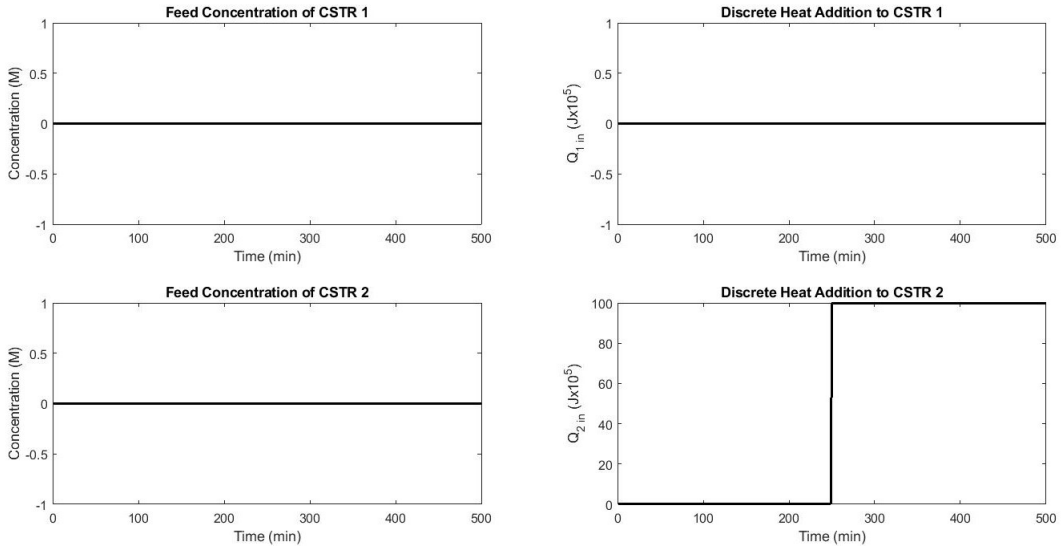


Figure 2.8: The input profiles used to generate the training data to conduct targeted validation.

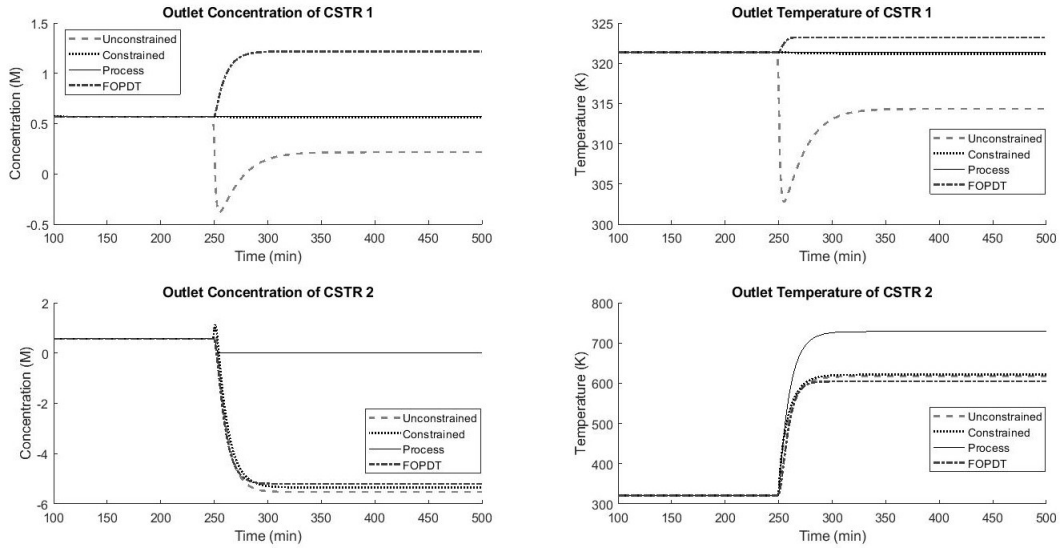


Figure 2.9: The figures show the outputs from each CSTR. In each figure the grey dashed line shows the unconstrained model predictions the black dotted line shows the constrained model predictions, the black dashed dot line shows the FOPDT model and the black line is the measured outputs. Starting in the top right and going clockwise the figures are as follows concentration leaving the first CSTR, temperature leaving the first CSTR, concentration leaving the second CSTR and finally temperature leaving the second CSTR.

While the focus of this paper is on incorporating known first principles knowledge

into the modeling technique, it is also important for the model to predict reasonably well for regular experiments. Using an input sequence similar to the training data as shown in Figure 2.10 a process run for validation was generated. Figure 2.11 shows the validation results comparing the standard and proposed subspace approaches along with the FOPDT model. The FOPDT model is compared for the sake of completeness, with the resultant plant-model mismatch being quite evident. Both subspace models perform reasonably well, and similar, however, the proposed subspace identification performs slightly better compared to the traditional approach. During validation, the proposed approach, being guided by first principles knowledge of the process, is able to make slightly better predictions.

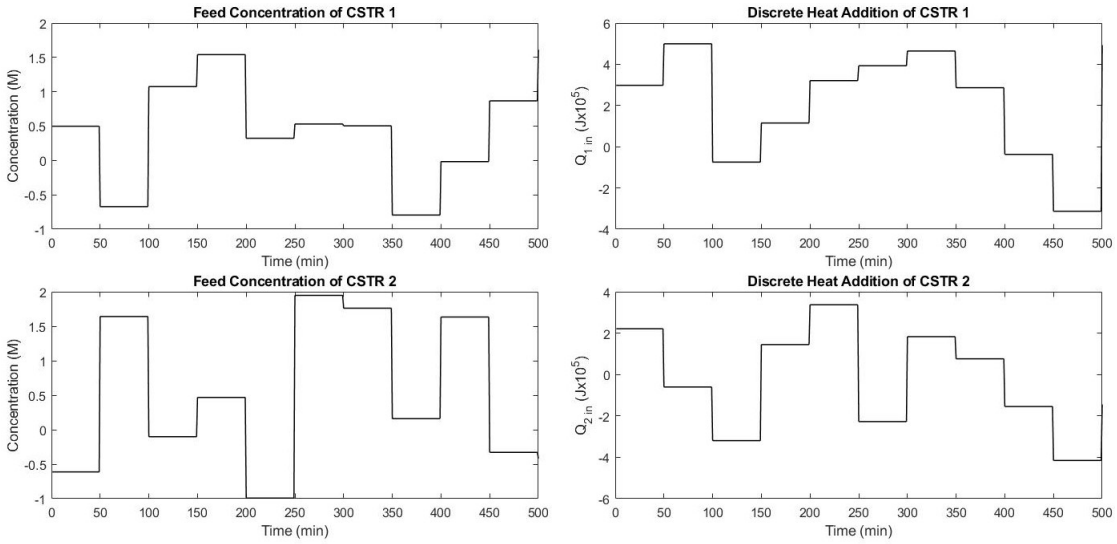


Figure 2.10: The input profiles that are used to generate the validation data.

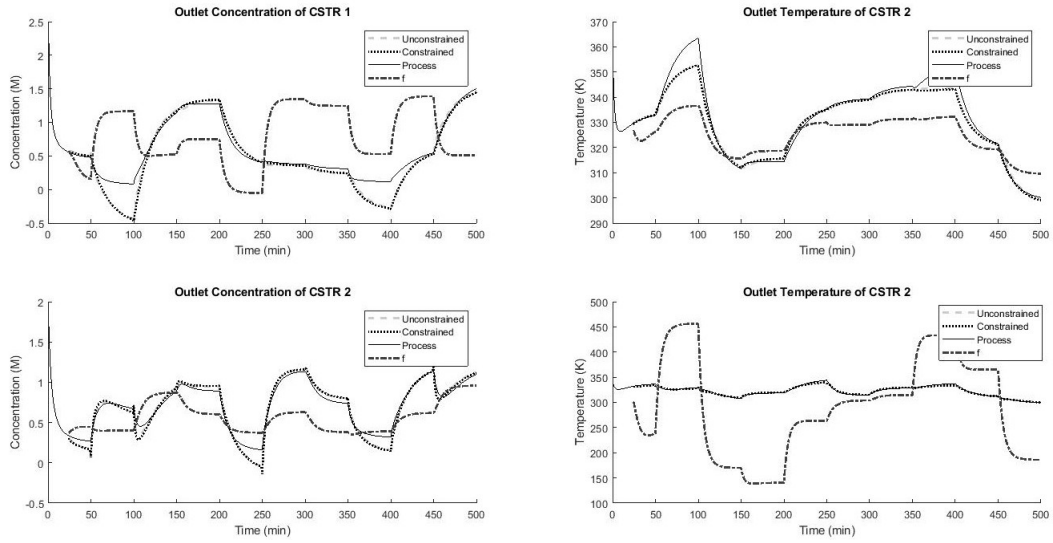


Figure 2.11: The figures show the outputs from each CSTR. In each figure the grey dashed line shows the unconstrained model predictions the black dotted line shows the constrained model predictions, the dashed and dotted line is the FOPDT model and the black line is the measured outputs. Starting in the top right and going clockwise the figures are as follows concentration leaving the first CSTR, temperature leaving the first CSTR, concentration leaving the second CSTR and finally temperature leaving the second CSTR.

Remark 9. *It is important to note that the simulation study in the present manuscript is meant to illustrate the key idea of the proposed approach. While we recognize that application to larger process examples would increase the computational time, an enabling feature is that the identification step is offline. Another point to consider is that for larger more complex problems, the tradeoff achieved by the proposed method becomes even more meaningful. Thus for larger problems, where solving the nonlinear optimization problem of the form of Eq. 2.14, that solves for the state trajectories and system matrices simultaneously in its entirety could become prohibitively expensive, the proposed approach represents a useful compromise that enables incorporation of first principles knowledge (compared to the standard subspace identification approach), and yet keeps the computational burden reasonable.*

2.6 Conclusions

In this work, a novel data driven and first principles model identification approach for subspace identification is proposed. The key idea in the proposed approach is to incorporate first principles information in the identification of a subspace-based state space model without converting it into a highly complex non-convex optimization problem. To this end, an algorithm is developed that first uses standard subspace identification method to generate state trajectories for the data. The state sequence along with the input and output data is used in a least squares minimization solution to generate the system model. The identified model is then utilized to initialize the constrained optimization problem. The states of the identified model can then be iteratively checked until they converge to the final constrained model. This constrained model has first principles knowledge of the process allowing it to predict the measured outputs more accurately as seen in the simulation results. In the simulation case study, the proposed approach was able to improve upon the traditional subspace model to predict the process outputs more accurately. When conducting a step test in the second CSTR, the traditional approach incorrectly predicts a change in the outputs of CSTR 1 while the proposed approach correctly identifies that no change occurs in the outlet from CSTR 1.

2.7 Acknowledgment

Financial support from the Ontario Graduate Scholarship Award, Corning Inc., and the McMaster Advanced Control Consortium is gratefully acknowledged.

Bibliography

[1] Alenany, A., Shang, H., Soliman, M., and Ziedan, I. (2011). Improved subspace identification with prior information using constrained least squares. *IET Control Theory and Applications*, 5(13):1568–1576(8).

[2] Bonvin, D. (2006). Control and optimization of batch processes. *IEEE control systems*, 26(6):34–45.

[3] Corbett, B. and Mhaskar, P. (2016). Subspace identification for data-driven modeling and quality control of batch processes. *AIChE Journal*, 62(5):1581–1601.

[4] Corbett, B. and Mhaskar, P. (2017). Data-driven modeling and quality control of variable duration batch processes with discrete inputs. *Industrial & Engineering Chemistry Research*, 56(24):6962–6980.

[5] Garg, A. and Mhaskar, P. (2017). Subspace Identification Based Modeling and Control of Batch Particulate Processes . *Industrial & Engineering Chemistry Research.* , 2017, Submitted.

[6] Ghosh, D., Hermonat, E., Mhaskar, P., Snowling, S., and Goel, R. (2019). A hybrid modeling approach integrating first principles models with subspace identification.

[7] Hu, B., Zhao, Z., and Liang, J. (2012). Multi-loop nonlinear internal model controller design under nonlinear dynamic pls framework using arx-neural network model. *Journal of Process Control*, 22(1):207–217.

[8] Huang, B., Ding, S. X., and Qin, S. J. (2005). Closed-loop subspace identification: an orthogonal projection approach. *Journal of process control*, 15(1):53–66.

[9] Inoue, M., Matsubayashi, A., and Adachi, S. (2015). Moment-constrained subspace identification using a priori knowledge. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 2731–2736. IEEE.

[10] Kozub, D. J. and MacGregor, J. F. (1992). Feedback control of polymer quality in semi-batch copolymerization reactors. *Chemical Engineering Science*, 47(4):929–942.

[11] Larimore, W. E. (1996). Statistical optimality and canonical variate analysis system identification. *Signal Processing*, 52(2):131 – 144. Subspace Methods, Part II: System Identification.

[12] Ljung, L. (2002). Prediction error estimation methods. *Circuits, Systems and Signal Processing*, 21(1):11–21.

[MacGregor et al.] MacGregor, J. F., Jaeckle, C., Kiparissides, C., and Koutoudi, M. Process monitoring and diagnosis by multiblock pls methods. *AIChE Journal*, 40(5):826–838.

[14] Moonen, M., De Moor, B., Vandenberghe, L., and Vandewalle, J. (1989). On-and off-line identification of linear state-space models. *International Journal of Control*, 49(1):219–232.

[15] Shi, D., El-Farra, N. H., Li, M., Mhaskar, P., and Christofides, P. D. (2006). Predictive control of particle size distribution in particulate processes. *Chemical Engineering Science*, 61(1):268–281.

[16] Söderström, T., Stoica, P., and Friedlander, B. (1991). An indirect prediction error method for system identification. *Automatica*, 27(1):183–188.

[17] Trnka, P. and Havlena, V. (2009). Subspace like identification incorporating prior information. *Automatica*, 45(4):1086–1091.

[18] Van Overschee, P. and De Moor, B. (1994). N4sid: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30(1):75–93.

[19] Van Overschee, P. and De Moor, B. (1995). A unifying theorem for three subspace system identification algorithms. *Automatica*, 31(12):1853–1864.

[20] Verhaegen, M. and Dewilde, P. (1992). Subspace model identification part 2. analysis of the elementary output-error state-space model identification algorithm. *International journal of control*, 56(5):1211–1241.

[21] Zhao, Y., Huang, B., Su, H., and Chu, J. (2012). Prediction error method for identification of lpv models. *Journal of process control*, 22(1):180–193.

Chapter 3

Model Predictive Control Using Subspace Model Identification

Where the previous chapter presented a novel integrated modeling approach this chapter continues to work with that model to present a novel model predictive control (MPC) state space algorithm. Traditional industrial (MPC) algorithms tend to ignore the feedthrough part of a state space model as there aren't direct input effects in the outputs. The key problem is that most subspace algorithms identify a feedthrough matrix due to error minimization techniques. This feedthrough matrix is therefore an important part of the subspace model and shouldn't be excluded in all cases. As such this work explored a key area of MPC algorithms to develop a quadratic program implementation that utilizes the feedthrough terms in the model.

Patel, N., Corbett, B., & Mhaskar, P. (2021). Model predictive control using subspace model identification. *Computers & Chemical Engineering*, 149, 107276.

3.1 Abstract

This paper addresses the problem of designing and implementing a data-driven model based model predictive controller (MPC). In particular, we consider the problem where a subspace identification approach is utilized to determine a state-space model, while applying first-principles based knowledge in the model identification (denoted as the constrained subspace model). The incorporation of the first-principles based constraints in the subspace matrix [22] often leads to a feed-through matrix being present. Such a model then is the best representation of the system dynamics, but does not lend itself readily to existing linear MPC formulations where the feed-through matrix is assumed to be zero. Thus, an existing linear MPC formulation is adapted to handle the feed through matrix. The superior performance of this MPC design, which can utilize the constrained subspace model, over existing approaches is demonstrated using a two tank chemical stirred tank reactor process.

3.2 Introduction

Advancements in computational power and increased automation in industry have enabled the use of advanced process control strategies to maintain plant operation at economic optimums, and to respond to market conditions. Model Predictive Control (MPC) is one such control strategy for which significant research has gone into studying the stability properties [15, 17, 27, 3] and in devising readily implementable quadratic program formulations [20] when the underlying model used in the MPC is linear.

Given the dependence on the model, it is only natural that the best possible model be identified and utilized within the MPC. To that end, there are two modeling and MPC approaches available: first principles based models that identify a model (and the associated parameters) by first starting with a model structure that is based on first principles knowledge, and utilizing them within an MPC [15, 16, 17] and data driven models which often use a simpler structure and are easier to identify and implement. [23, 12, 29, 18, 9, 26] While good first principles models are reliable and good for extrapolation, they are generally difficult to develop, and more importantly, difficult to maintain. Data driven models, on the other hand, are relatively easy to develop. Some approaches to developing linear time invariant models using data driven techniques include prediction error minimization techniques [23, 12, 29] and linear regression modeling [8]. In contrast to these approaches, which provide a computational challenge, subspace identification is intrinsically more computationally tractable.

One of the these regression modelling techniques is the partial least squares (PLS) method [8]. In this approach, data from each batch is organized and then projected onto a lower dimension (latent variable space) which has guaranteed latent variable independence, allowing the nature of the process to be determined [MacGregor et al.]. In order to handle batch data the data is oriented in series where columns represent

measurements and rows represent observations. PLS generates a time-dependent linear model but requires additional techniques to handle non-uniform batch length. The use of first-principles knowledge involves the creation of new variables calculated from the first principles equations and appending the columns to the dataset.

Other techniques such as prediction error minimization (PEM) approaches solve the problem by reducing the sum squared error between the predicted and measured outputs. These approaches are also capable of applying first principles based constraints; the difficulty arises due to the increase computational complexity. As this is a non-convex optimization problem successful model identification relies on a good initial set of parameters. [12] An additional drawback is that PEM approaches must compute and compare both the system states and the predicted outputs at each iteration. Thus the addition of highly nonlinear first principles based constraints makes these approaches more computationally demanding.

The incorporation of first principles based constraints has been achieved in subspace identification approaches with similar drawbacks. One approach [1] utilizes a constrained least squares solution using weighted constraints to solve the problem with traditional least squares techniques. The approach was limited to equality constraints and required the system matrices to be solved in an intermediate step instead of with the states.[1] This approach can result in the system matrices (and the system dynamics) being inconsistent with the constraints.

Other approaches that consider a priori knowledge with parameter estimation techniques use a Bayesian framework to impose steady state gains. [24] These constraints are only imposed as soft constraints and may not be satisfied. Another approach utilizes weighted constraints on the moment of the transfer function to solve the problem as a quadratic optimization approach. [10] In summation, there are existing approaches capable of incorporating first principles knowledge explicitly as constraints but none as specifically targeted to the constrained subspace based dynamic batch

models.

Subspace identification [18, 9, 26, 11, 25, 28] results in a linear time invariant model, and has been adapted to handle data from several runs [5, 4, 6]. Owing to the fact that these are intrinsically data driven approaches, it results in one or both of two possible scenarios occurring: firstly where the subspace model does not respect physical process constraints or secondly where a nonzero feed-through matrix is identified. The first scenario is a result of the possibility where in the process a truly 'feedthrough' term is present, but the feedthrough term has not adequately manifested itself in the training data. Data driven modeling also relies on the accuracy of the data, which could be noisy, and can therefore, be led astray when finding the best model. The first scenario is more problematic since a process model that does not match first principles knowledge, such as process gains with opposite signs compared to what is known and expected, means that control action taken is contrary to the one desired. It is necessary to introduce constraints to the data driven approach in a suitable manner to correct these gains.[1, 24, 10, 22] Previous work [22] presented a synergized approach combining first principles knowledge with data driven subspace model identification to identify a constrained subspace model. The constrained model is able to accurately capture the process dynamics and is better at predicting the process in comparison to the traditional unconstrained subspace model, especially with respect to being true to first principles knowledge. Additionally, in identifying the best model traditional (and constrained) subspace model identification may result in a nonzero feed-through matrix. In order to use these models in the MPC, the state space MPC algorithm [20] must be accordingly modified. Recall that state space MPCs have traditionally considered a subspace model where the feed-through matrix is zero and thus rely on the process outputs being a linear function of the states alone (and not the outputs). This realization is in part due to subspace identification having a background in electrical engineering systems where feed-through is more common in comparison to MPC which was developed for use in process systems engineering where the dynamics

are typically slower compared to other systems.

To address the problem of utilizing a physically consistent data driven model for improved control, this manuscript presents a state space MPC using a constrained subspace model. A constrained subspace model is identified for an illustrative two-stage chemical stirred tank reactor (CSTR) system and then MPC is used to stabilize the system from various initial conditions. The proposed model based MPC is compared with other models where the feed-through matrix is set to zero and with the traditional subspace model. By evaluating the objective function for the closed-loop implementation, the effectiveness of the model is evaluated. The constrained model based MPC has better performance in comparison to both traditional subspace models with a feed-through matrix and with the feed-through matrix set to zero. The rest of the paper is organized as follows: Section 3.3 presents a two stage CSTR process as the motivating example, followed by an overview of subspace identification methods and MPC algorithms. Section 3.5.1 presents the subspace identification approach and MPC formulation. In Section 3.5, an application of the proposed approach to the CSTR process is presented. Finally, concluding remarks are made in Section 3.6.

3.3 Preliminaries

3.3.1 Motivating Example: 2 Chemical Stirred Tank Reactors (CSTRs) in Series

The process example used to demonstrate the capabilities of the state space MPC consists of two CSTRs that are connected in series (see Figure 3.1). In this example there are four inputs: the fresh feed concentration and the amount of heat added can be manipulated for each CSTR. There are also four outputs with the effluent

concentration and temperature being measured from each CSTR. A first principles model consisting of 4 ordinary differential equations describes the evolution of the concentration of A, C_{A_i}, and the reactor temperature, T_i, $i = 1, 2$ for each CSTR as follows:

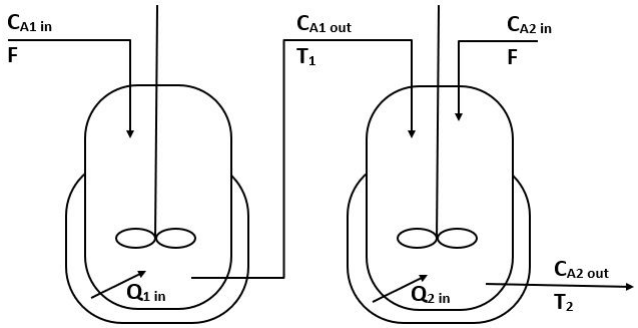


Figure 3.1: The schematic of the CSTR model

$$
\begin{aligned}
\frac{dC_{A1}}{dt} &= -\frac{F}{V}(C_{A1,in} - C_{A1,ss} + C_{A,0} - C_{A1}) - k_o e^{\frac{-E}{RT_1}} C_{A1in} \\
\frac{dT_1}{dt} &= -\frac{F}{V}(T_0 - T_1) - \frac{\Delta H}{C_p \rho} k_o e^{\frac{-E}{RT_1}} C_{Ain} + \frac{Q_{1in}}{C_p} \rho V \\
\frac{dC_{A2}}{dt} &= -\frac{F}{V}(C_{A2,in} - C_{A2,ss} + C_{A,0} - C_{A2}) - k_o e^{\frac{-E}{RT_2}} C_{A1in} + \frac{F}{V}(C_{A1} - C_{A2}) \\
\frac{dT_2}{dt} &= -\frac{F}{V}(T_1 - T_2) - \frac{\Delta H}{C_p \rho} k_o e^{\frac{-E}{RT_2}} C_{A2,in} + \frac{Q_{2in}}{C_p} \rho V + \frac{F}{V}(T_0 - T_2)
\end{aligned}
\tag{3.1}
$$

where ρ is the density of the fluid, F is the inlet flow rate to the first CSTR, V is the volume of each CSTR, R is the gas transfer coefficient, E is the activation energy of the reaction, k_0 is the value of the Arrhenius constant, T_0 is the inlet feed temperature, $C_{A,0}$ is the inlet feed concentration, ΔH is the heat of reaction, C_p is the heat capacity of the solution (see Table 3.1 for the parameter values). The input vector u includes $C_{A1,in}$, Q_{1in}, $C_{A2,in}$, Q_{1in} and the output vector includes y as C_{A1}, T_1, C_{A2}, T_2.

From Figure 3.1 (and first principles knowledge) it is clear that any changes to the fresh inlet of CSTR 2 should not have any effect on the outputs of CSTR 1. In other

Table 3.1: Parameter values for the motivating process example

Parameter	Value	Unit	Parameter	Value	Unit
V	0.1	m^3	ρ	1000	kg/m^3
R	8.314	$J/(mol \cdot K)$	E	8.314×10^4	$\mu m/s$
$C_{A,0}$	2	mol/L	k_0	7.2×10^{10}	K
ΔH	4.78×10^4	kJ/kg	C_p	0.239	$kJ/K \cdot kg$
T_0	310	K	F	100	L/s

words, in the process model, the transfer function between the respective inputs and outputs should be zero.

Remark 10. *The specific choice of 'certain' process knowledge that should be incorporated in the data driven modeling must be done carefully. The idea is to avoid introducing additional parameters that must be identified as part of the identification procedure making the identification even more complex. Only known relationships between variables, such as certain transfer functions being zero, yet others being positive or negative should be utilized as constraints in the model identification procedure. That said, if much more detailed first principles knowledge must be incorporated, it can be done in another way [7]. These alternate approaches [7] address the situation where fairly complex first principles information is available (in the form of detailed models) but may contain parameters that are difficult to estimate. In such instances, the first principles knowledge/model can be incorporated in a parallel fashion instead of as constraints in the identification procedure.*

3.3.2 Subspace Identification

This subsection provides an overview of traditional subspace identification methods used to identify a state space model using process input and output data. In this approach, given a set of process inputs $u_k \in \mathbb{R}^p$ and the outputs $y_k \in \mathbb{R}^q$ for a run of

s time steps a subspace model of the following form is identified:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k, \tag{3.2}$$

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{D}\mathbf{u}_k, \tag{3.3}$$

where the order n of this unknown system and the system matrices $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times p}$, $\mathbf{C} \in \mathbb{R}^{q\times n}$, $\mathbf{D} \in \mathbb{R}^{q\times p}$ are determined by finding the best fit to the training runs. Consider now data from a run where k is the sampling instant since the run is initiated (ie. setpoint change applied) and b denotes the run number. Then the 'standard' Hankel matrix can be appropriately modified for a single run as:

$$\mathbf{Y}_{1|i}^{(b)} = \begin{bmatrix} \mathbf{y_1}^{(b)} & \mathbf{y_2}^{(b)} & \cdots & \mathbf{y_{j^b}}^{(b)} \\ \vdots & \vdots & & \vdots \\ \mathbf{y_i}^{(b)} & \mathbf{y_{i+1}}^{(b)} & \cdots & \mathbf{y}_{\mathbf{i+j^{(b)}-1}}^{(\mathbf{b})} \end{bmatrix} \quad \forall\, b = 1, \ldots, nb \tag{3.4}$$

where nb is the number of runs being used for identification.

The above Hankel matrix is a simple representation of data collected from a single process run. In order to analyze data from multiple runs a new matrix must be constructed. Note that a simple concatenation of data from all runs would not provide a distinct separation between the end of one run and the start of the next. The key to utilizing data from multiple runs lies in building a pseudo-Hankel matrix where the data is separated by run. This is achieved by horizontally concatenating the individual Hankel sub-matrices into a single matrix for both inputs and outputs as follows:

$$\mathbf{Y}_{1|i} = \begin{bmatrix} \mathbf{Y}_{1|i}^{(1)} & \mathbf{Y}_{1|i}^{(2)} & \cdots & \mathbf{Y}_{1|i}^{(nb)} \end{bmatrix} \tag{3.5}$$

Incidentally using this approach does not require data from various runs to be identical in length. From these pseudo-Hankel matrices a deterministic algorithm [18] can be utilized to identify state trajectories that can be similarly concatenated as follows:

$$\hat{\mathbf{X}}_{i+1}^{(b)} = \begin{bmatrix} \hat{\mathbf{x}_{\mathbf{i+1}}}^{(b)} & \cdots & \hat{\mathbf{x}_{\mathbf{i+j^{(b)}}}}^{(b)} \end{bmatrix} \quad \forall\, b = 1, \ldots, nb \tag{3.6}$$

$$\hat{\mathbf{X}}_{i+1} = \begin{bmatrix} \hat{\mathbf{X}}_{i+1}^{(1)} & \hat{\mathbf{X}}_{i+1}^{(2)} & \cdots & \hat{\mathbf{X}}_{i+1}^{(nb)} \end{bmatrix} \tag{3.7}$$

Using a standard least squares solution, the system matrices can be subsequently identified as follows:

$$\mathbf{Y}_{reg}^{(b)} = \begin{bmatrix} \hat{\mathbf{x}_{\mathbf{i+2}}}^{(b)} & \cdots & \hat{\mathbf{x}_{\mathbf{i+j^{(b)}}}}^{(b)} \\ \mathbf{y_{i+1}}^{(b)} & \cdots & \mathbf{y_{i+j^{(b)}-1}}^{(b)} \end{bmatrix} \tag{3.8}$$

$$\mathbf{X}_{reg}^{(b)} = \begin{bmatrix} \hat{\mathbf{x}_{\mathbf{i+1}}}^{(b)} & \cdots & \hat{\mathbf{x}_{\mathbf{i+j^{(b)}-1}}}^{(b)} \\ \mathbf{u_{i+1}}^{(b)} & \cdots & \mathbf{u_{i+j^{(b)}-1}}^{(b)} \end{bmatrix} \tag{3.9}$$

$$\begin{bmatrix} \mathbf{Y}_{reg}^{(1)} & \cdots & \mathbf{Y}_{reg}^{(nb)} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{X}_{reg}^{(1)} & \cdots & \mathbf{X}_{reg}^{(nb)} \end{bmatrix} \tag{3.10}$$

The system matrices **A**,**B**,**C**, and **D** make up the traditional "unconstrained" subspace model. The key consideration in this approach is that using subspace identification techniques allows for the model to be identified with minimal complexity, albeit without necessarily respecting physical constraints. In the case of the CSTR example, the subspace identification procedure results in incorrect process gains. In particular, the unconstrained subspace model predicts new incorrect steady states for both temperature and concentration variables in CSTR 1 (see Figure 3.2) when introducing a step change in CSTR 2 (see Figure 3.3 for the validation inputs).

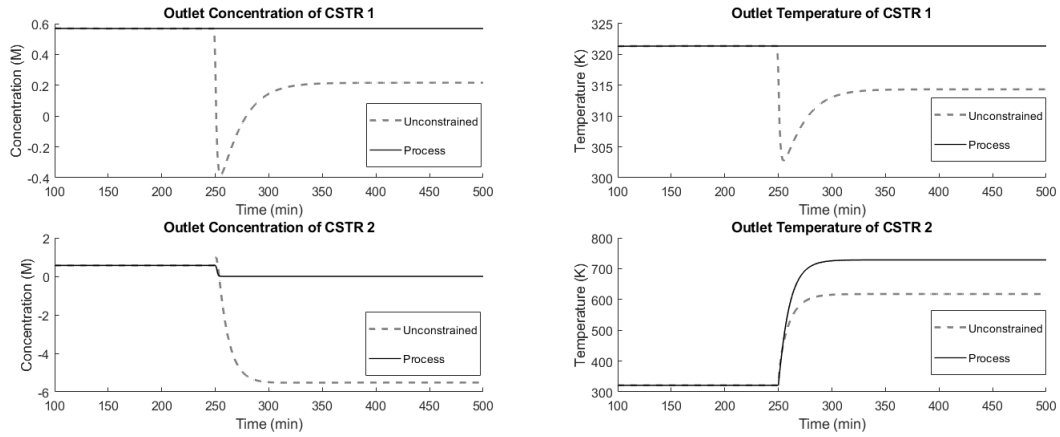


Figure 3.2: Outputs from both CSTRs after applying the targeted input sequence. The solid black line represents the process and the grey dashed line represents the predictions by the subspace model.

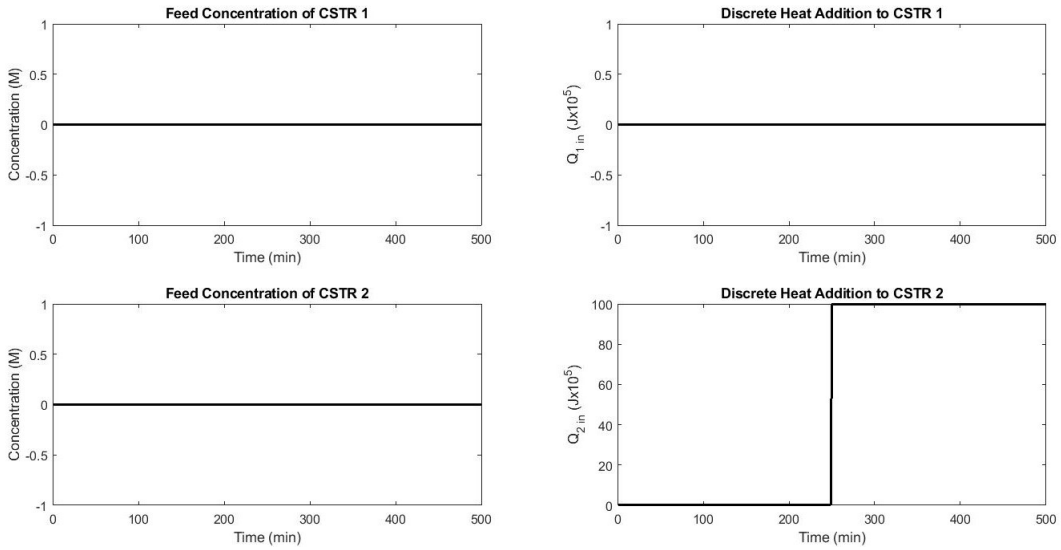


Figure 3.3: The input profiles used in targeted validation

3.3.3 Constrained Subspace Identification

This subsection highlights the approach used to impose first principles based constraints on the model.[22] The introduction of constraints and an iterative optimization scheme results in a new set of constrained system matrices consistent with the

states.

A generalized formulation of the optimization problem used to identify the constrained model is shown below:

$$\begin{aligned} \min_{\theta} \quad & (\mathbf{Y} - \theta\mathbf{X})^{\mathbf{T}}(\mathbf{Y} - \theta\mathbf{X}) \\ s.t. \quad & \textit{First principles knowledge based constraints} \end{aligned} \tag{3.11}$$

$$\theta = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \tag{3.12}$$

where Y and X are described in Equation 3.10, θ is comprised of the system matrices, and the constraints are derived from appropriate first principles knowledge. The first principles knowledge referred to in this work is the gain constraints between the inputs of the 2nd CSTR and the outputs of the 1st CSTR. This optimization problem is solved in a previous paper and the state space model has been included in this work as a starting point for the reader. The problem is solved using gain constraints as shown in Eqn 13 that are placed on the the least squares solution in Eqn 10. Using a nonlinear optimization solver to solve this problem allows for a model to be identified with the necessary first principles knowledge included. The resulting system matrices is called the constrained model. For a detailed explanation of how the constrained subspace model is identified see Patel et al. [22].

In the context of the CSTR example, the first principles knowledge based constraints

in Eqn 11 result in the optimization problem taking the form shown in Eqn 3.13:

$$\begin{aligned} \min_{\theta} \quad & (\mathbf{Y}-\theta\mathbf{X})^{\mathbf{T}}(\mathbf{Y}-\theta\mathbf{X}) \\ s.t. \quad & \mathbf{G}(\mathbf{1},\mathbf{3}) = 0 \\ & \mathbf{G}(\mathbf{1},\mathbf{4}) = 0 \\ & \mathbf{G}(\mathbf{2},\mathbf{3}) = 0 \\ & \mathbf{G}(\mathbf{2},\mathbf{4}) = 0 \\ where \quad & \mathbf{G}_{\theta} = tf(\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D}) \end{aligned} \tag{3.13}$$

where $tf(\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D})$ denotes the transfer function resulting from the state space matrices, and thus for instance, $\mathbf{G}(\mathbf{1},\mathbf{3})$ being zero implies that the transfer function between the third input (which is one of the inlets to the CSTR 2) and the first output (which is one of the outlets from CSTR 1) should be zero. Now when the same targeted input sequence is applied to the new constrained model, which has the appropriate process gains, the CSTR outputs are correctly predicted. In Figure 2.9 the predictions from the constrained model are compared against the unconstrained model. In the top two plots the constrained model clearly remains at the steady state instead of shifting like the unconstrained model does when the input shifts. Note that all the models demonstrate some level of plant-model mismatch resulting from process nonlinearity. However, the key point to note here is that the constrained model, with first principles based constraints, is able to more accurately predict the process online in comparison to the unconstrained model, and more importantly, respect the physical knowledge that the inlet to CSTR 2 should not effect CSTR 1.

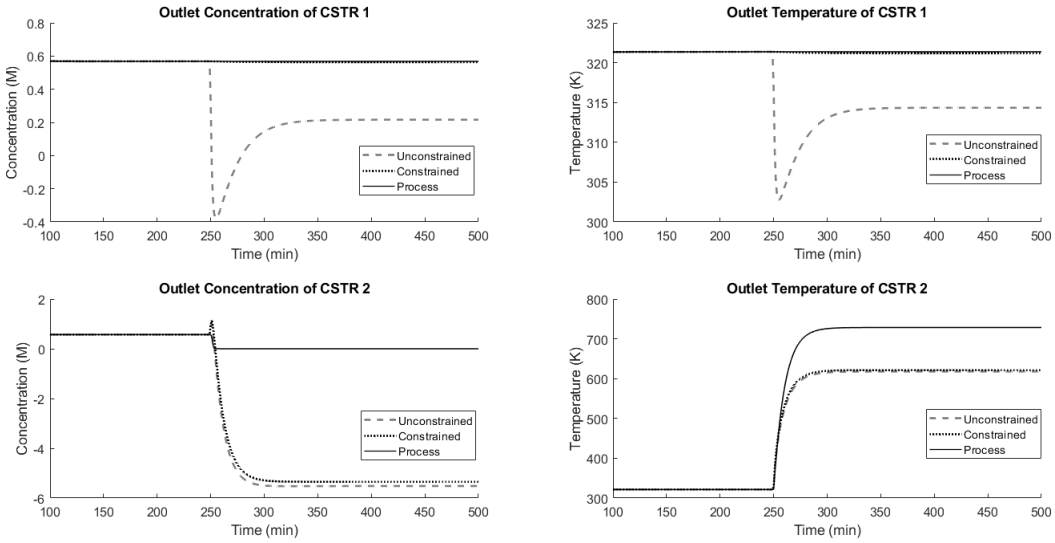


Figure 3.4: The figures show the outputs from each CSTR. In each figure the grey dashed line shows the unconstrained model predictions the black dotted line shows the constrained model predictions and the black line is the process outputs.

Remark 11. *The advantage to incorporating the constraint in the manner done in the present manuscript in comparison to prediction error minimization methods (PEM) is that PEM requires the entire trajectory and system matrices to be computed at once. The resulting optimization problem becomes quite large and includes highly nonlinear transfer function constraints. The constraints in this approach are proposed in a way to leverage the linear regression structure of subspace methods that can be computed quickly. Thus, this approach offers a trade-off between a complete solution (PEM based) and ignoring first principles knowledge completely as is the case with traditional subspace identification.*

Remark 12. *Note that in terms of the fit (which is the objective function when determining the models), the traditional unconstrained method has a better fit than the constrained model. However, when using the model for extrapolation (Figure 3.4) the constrained model predicts better. This is especially true for regions of operation where the physical constraints captured by the constrained model are 'active', thus in regions where the gain between*

certain inputs and outputs being zero is more pronounced. Having this information available to the model in the MPC in turn is expected to improve the closed-loop performance.

3.3.4 Traditional MPC

This subsection provides an overview of a traditional linear MPC Muske and Rawlings [20]. In this formulation, the MPC uses an infinite horizon open-loop objective function (see Eqn 3.14) which is minimized to obtain the optimal input trajectory $\mathbf{u^N}$, where $\mathbf{u^N} = \begin{bmatrix} \mathbf{u_0} & \mathbf{u_1} & \dots & \mathbf{u_{N-1}} \end{bmatrix}^T$.

$$\underset{u^N}{\text{minimize}} \quad \sum_{k=0}^{\infty} (\mathbf{y_k})^{\mathbf{T}} \mathbf{Q} (\mathbf{y_k}) + (\mathbf{u_k})^{\mathbf{T}} \mathbf{R} (\mathbf{u_k}) + \mathbf{\Delta u_k^T S \Delta u_k} \tag{3.14}$$

$\mathbf{y_k}$ and $\mathbf{u_k}$ represent the model output and model input k time steps from the initialization time. $\Delta\mathbf{u_k}$ is the change in inputs between time step k and time step k-1. Q, R, and S are all symmetric positive (Q) and positive semi-definite (R and S) matrices used to penalize the outputs, inputs and change in inputs between time steps respectively. N represents the prediction horizon. Eqn 3.14 can be expressed using a terminal cost function with a penalty matrix $\overline{\mathbf{Q}}$ to capture the infinite horizon cost. Thus, the optimization problem can be rewritten for a finite prediction horizon N as follows:

$$
\begin{aligned}
\underset{u^N}{\text{minimize}} \quad & (\mathbf{x_N})^{\mathbf{T}}\overline{\mathbf{Q}}(\mathbf{x_N}) + \mathbf{\Delta u_N^T S \Delta u_N} + \\
& \sum_{k=0}^{N-1} (\mathbf{x_k})^{\mathbf{T}}\mathbf{C}^{\mathbf{T}}\mathbf{Q}\mathbf{C}(\mathbf{x_k}) + (\mathbf{u_k})^{\mathbf{T}}\mathbf{R}(\mathbf{u_k}) + \mathbf{\Delta u_k^T S \Delta u_k} \\
\text{s.t.} \quad & \mathbf{x_{k+1}} = \mathbf{A x_k} + \mathbf{B u_k}\ ,\ k = 0, \ldots, N-1 \\
& \mathbf{\Delta u_k} = \mathbf{u_k} - \mathbf{u_{k-1}}\ ,\ k = 0, \ldots, N \\
& \mathbf{u_{min}} \leq \mathbf{u_k} \leq \mathbf{u_{max}}\ ,\ k = 0, \ldots, N-1 \\
& \mathbf{u_k} = 0\ ,\ k \geq N
\end{aligned}
\tag{3.15}
$$

For an open-loop stable process, $\overline{Q}$ is determined from the solution of the following discrete Lyapunov equation:

$$
\overline{\mathbf{Q}} = \mathbf{C}^{\mathbf{T}}\mathbf{Q}\mathbf{C} + \mathbf{A}^{\mathbf{T}}\overline{\mathbf{Q}}\mathbf{A} \tag{3.16}
$$

The optimization problem shown in Eqn 3.15 can be converted to a standard quadratic program (QP) by redefining the matrices as shown in Eqn 3.17.

$$
\begin{aligned}
\underset{u^N}{\text{minimize}} \quad & \frac{1}{2}\left(\mathbf{u^N}\right)^{\mathbf{T}}\mathbf{H}\mathbf{u^N} + \left(\mathbf{G}\left(\mathbf{x_0}\right) - \mathbf{F}\left(\mathbf{u_{-1}}\right)\right)^{\mathbf{T}}\mathbf{u^N} \\
\text{s.t.} \quad & \mathbf{u_{min}} \leq \mathbf{u^N} \leq \mathbf{u_{max}}
\end{aligned}
\tag{3.17}
$$

$u_{min},\ u_{max} \in \mathbb{R}^{N*p}$ are vectors that represent the lower and upper bounds on the constraints (in deviation form) respectively. u_{-1} represents the input at the previous time step in deviation form. Finally, the matrices H, G, F are given in Eqns 3.18 and 3.19 from continuous substitution using Eqn 3.2 and Eqn 3.3. [20]

$$\mathbf{H} = \begin{bmatrix} \mathbf{B^T\overline{Q}B + R + 2S} & \mathbf{B^TA^T\overline{Q}B - S} & \dots & \mathbf{B^TA^{T^{N-1}}\overline{Q}B} \\ \mathbf{B^T\overline{Q}AB - S} & \mathbf{B^T\overline{Q}B + R + 2S} & \dots & \mathbf{B^TA^{T^{N-2}}\overline{Q}B} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{B^T\overline{Q}A^{N-1}B} & \mathbf{B^T\overline{Q}A^{N-2}B} & \dots & \mathbf{B^T\overline{Q}B + R + 2S} \end{bmatrix} \tag{3.18}$$

$$\mathbf{G} = \begin{bmatrix} \mathbf{B^T\overline{Q}A} \\ \mathbf{B^T\overline{Q}A^2} \\ \vdots \\ \mathbf{B^T\overline{Q}A^N} \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \mathbf{S} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} \tag{3.19}$$

3.4 Proposed MPC Formulation

3.4.1 MPC Design

In this section, the MPC formulation is adapted to allow using the feed-through matrix in the model. To this end, the state space MPC approach described in Section 3.3.4 is adapted to utilize the subspace model proposed in Eqn 3.10 where the D matrix is nonzero to result in the following formulation:

$$
\begin{aligned}
\underset{u^N}{\text{minimize}} \quad & (\mathbf{x_N})^{\mathbf{T}}\overline{\mathbf{Q}}(\mathbf{x_N}) + \mathbf{\Delta u_N^T S \Delta u_N} + \\
& \sum_{k=0}^{N-1} (\mathbf{x_k})^{\mathbf{T}}\mathbf{C}^{\mathbf{T}}\mathbf{Q}\mathbf{C}(\mathbf{x_k}) + (\mathbf{u_k})^{\mathbf{T}}\mathbf{R}(\mathbf{u_k}) + \mathbf{\Delta u_k^T S \Delta u_k} \\
& \qquad +\mathbf{2}(\mathbf{u_k})^{\mathbf{T}}\mathbf{D}^{\mathbf{T}}\mathbf{Q}\mathbf{C}(\mathbf{x_k}) + (\mathbf{u_k})^{\mathbf{T}}\mathbf{D}^{\mathbf{T}}\mathbf{Q}\mathbf{D}(\mathbf{u_k}) \\
\text{s.t.} \quad & \mathbf{x_{k+1}} = \mathbf{A x_k} + \mathbf{B u_k}\ ,\ k = 0, \ldots, N-1 \\
& \mathbf{\Delta u_k} = \mathbf{u_k} - \mathbf{u_{k-1}}\ ,\ k = 0, \ldots, N \\
& \mathbf{u_{min}} \leq \mathbf{u_k} \leq \mathbf{u_{max}}\ ,\ k = 0, \ldots, N-1 \\
& \mathbf{u_k} = 0\ ,\ k \geq N
\end{aligned}
\tag{3.20}
$$

The key differences in the formulation result from the process outputs being a product of the states and inputs. This gives rise to the additional terms containing the D matrix. Note that the terminal costs function does not change since at the end of the prediction horizon the inputs are zero.

This leads to a new QP formulation using the following matrices:

$$
\begin{aligned}
\underset{u^N}{\text{minimize}} \quad & \frac{1}{2}\left(\mathbf{u^N}\right)^{\mathbf{T}}\mathbf{H_1}\mathbf{u^N} + \left(\mathbf{G_1}\left(\mathbf{x_0}\right) - \mathbf{F_1}\left(\mathbf{u_{-1}}\right)\right)^{\mathbf{T}}\mathbf{u^N} \\
\text{s.t.} \quad & \mathbf{u_{min}} \leq \mathbf{u^N} \leq \mathbf{u_{max}}
\end{aligned}
\tag{3.21}
$$

Finally, the matrices H_1, G_1, F_1 are given in Eqns 3.21 and 3.23 from continuous substitution using Eqn 3.2 and Eqn 3.3. [20]

$$
\mathbf{H_1} = \begin{bmatrix}
\mathbf{B^T\overline{Q}B + R + 2S + D^TQD} & \mathbf{B^TA^T\overline{Q}B - S + 2D^TCB} & \ldots & \mathbf{B^TA^{T^{N-1}}\overline{Q}B + 2D^TA^{N-1}CB} \\
\mathbf{B^T\overline{Q}AB - S} & \mathbf{B^T\overline{Q}B + R + 2S + D^TQD} & \ldots & \mathbf{B^TA^{T^{N-2}}\overline{Q}B + 2D^TA^{N-2}CB} \\
\vdots & \vdots & \ddots & \vdots \\
\mathbf{B^T\overline{Q}A^{N-1}B} & \mathbf{B^T\overline{Q}A^{N-2}B} & \ldots & \mathbf{B^T\overline{Q}B + R + 2S + D^TQD}
\end{bmatrix}
\tag{3.22}
$$

$$\mathbf{G_1} = \begin{bmatrix} \mathbf{B^T\overline{Q}A + D^TQCA} \\ \mathbf{B^T\overline{Q}A^2 + D^TQCA^2} \\ \vdots \\ \mathbf{B^T\overline{Q}A^N + D^TQCA^N} \end{bmatrix} \quad , \quad \mathbf{F_1} = \begin{bmatrix} \mathbf{S} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} \tag{3.23}$$

Remark 13. *The key benefit of the above formulation is in retaining the nature of the problem as a quadratic program. Thus in principle, the feedthrough matrix or feedthrough effect can readily be handled by nonlinear MPC formulations, with the optimization problem turning into a non-convex problem (or at least resulting in the optimizer not exploiting the essentially quadratic program structure).*

Remark 14. *Note that if one were to use a nonlinear model (assuming that an accurate enough model was available), a feedthrough term would not be necessary. However, when identifying a linear model from process data, it is evident that retaining a feedthrough term is better than dropping it, purely from a fit and prediction perspective. Note that there could very well be systems wherein the D matrix (or parts of it) should be zero and the present modeling and control formulation allows for these as special cases. More importantly it allows for cases where the dynamics between some of the inputs and outputs are very quick, resulting in a situation where it is best to capture the relationship as a feed-through term in the time scale at which the model is developed. In essence, the proposed approach allows for handling multi-rate behaviour present in many physical systems. In bioreactors for example, changes in temperature happen much faster in comparison to changes in cell titer. From a modeling perspective being able to account for the impact of these fast modes in the process using feed-through terms could be beneficial in overcoming the modeling challenge of modeling such multi-rate dynamics.*

Remark 15. *It is important to note that utilizing a constrained subspace model to improve MPC performance is at its core different from, and complimentary to, an offset free MPC algorithm.[14, 19, 21, 2] This approach focuses on improving the model at the modeling stage itself, and can very well be complemented with an offset free mechanism to further improve closed-loop performance. One of the directions of future work would be to incorporate this model within an offset-free formulation to demonstrate further improvement.*

3.5 Application to the Motivating Example

To show the effectiveness of the constrained subspace model identification approach the two stage CSTR process is controlled using an MPC algorithm. The constrained model is compared against the traditional subspace model and the traditional subspace model where the feed-through matrix is set to zero.

3.5.1 MPC Implementation

In this scenario, the two stage CSTR process is initialized at a range of initial conditions within 5%, 25% and 50% away from the steady state. 10 sets of random initial conditions are created within each threshold and used to initialize the process. An MPC is then implemented, using a Luenberger state observer to estimate the subspace states. Until the state observer converges the system is given the nominal input. For each MPC implementation, a prediction horizon of 5 time steps is chosen. The MPC

parameters Q,R and S are shown below:

$$\mathbf{Q} = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix} \mathbf{R} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{S} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

Remark 16. *The choice of tuning parameters was specific to this process model where the goal was to penalize the changes in the inputs to avoid extreme input changes that are not practical in an industrial setting. Additionally, the penalty on the states was designed to drive the system to the steady state. These tuning parameters are purely based on the outcome the user hopes to achieve and can take any values as long as Q is positive definite and R and S are positive semi-definite.*

Remark 17. *Note that while the present implementation utilizes a Luenberger observer, in practice any state observer can be utilized for this approach depending on the nature of the system. Future work will consider the effects of the state observer on the MPC algorithm when attempting to stabilize the system. Note that the state observer is an important part of the MPC implementation since the model states need to converge before it can be used in computing the control action. Therefore, instead of computing control action that can be potentially disruptive to the system, only the nominal input is applied to the system till the state observer converges.*

3.5.2 Closed-loop Results

In order to determine the effectiveness of the model, the control sequence calculated by the MPC can be used. The criteria used in this approach is to evaluate the integral

of the objective function in Eqn 3.15 by taking the sum of the objective function at each time step until convergence. The highest summation of the objective function is thus the MPC implementation (and the model) that performs the worst. The values reported in Table 3.2 were taken as an average over ten different initial conditions within each of the various percent deviation thresholds.

Each of the ten simulations has the states initialized at a random value at 5%, 25% and 50% away from the steady state. The error is then calculated using the input and state trajectory to evaluate the value of the objective function as shown in Eqn 3.21. The average error for the constrained model is lower in comparison to both the traditional and the traditional model with the D matrix set to 0. The constrained model without a D matrix also has a lower error in comparison to the traditional subspace model. This is a result of the constrained models having the correct process gains which allows the MPC to take better control action and ultimately reach the steady state faster.

Figure 3.5 shows the input trajectories calculated by the MPC using the four different subspace models for one initial condition. As expected the two constrained models, with and without the D matrix, compute similar input trajectories which are significantly different from the two unconstrained models. The difference in the input trajectories results in Figure 3.6 where both the constrained models based MPC stabilize much faster in comparison to the unconstrained model based MPCs. The true steady state values of the system were $0.385 mol/L$ and $332.3K$ for the outlet concentration and temperature respectively of both CSTRs. When comparing the performance of the MPC with and without a feed-through matrix the MPC with the D matrix is capable of stabilizing faster. Figure 3.6 shows how the additional oscillations in the unconstrained model and the slow response of the others results in the constrained model with a D matrix having the best model fit in Table 3.2. In regards to the predictive capabilities of the model Table 3.3 shows the validation results as

demonstrated by Figure 3.7. This figure shows how the unconstrained models diverge from the steady state due to incorrect process gains. Additionally, the constrained model without a D matrix deviates from the setpoint slightly as shown by the higher error value in Table 3.3. While the prediction error for a fixed input change is relatively little, the mismatch manifests in poorer control performance as demonstrated by Figure 3.7 and Table 3.3.

Table 3.2: The average value of the MPC objective function from each of the subspace models starting from initial conditions at a percent deviation away from the steady state values

Model	*5%*	*25%*	*50%*
Traditional	3.0907e3	5.7666e4	2.0385e5
Traditional with no D Matrix	2.0581e3	4.2171e4	1.4870e5
Constrained with no D matrix	1.2982e3	2.7312e4	9.5921e4
Constrained	624.3626	1.3396e4	4.7102e4

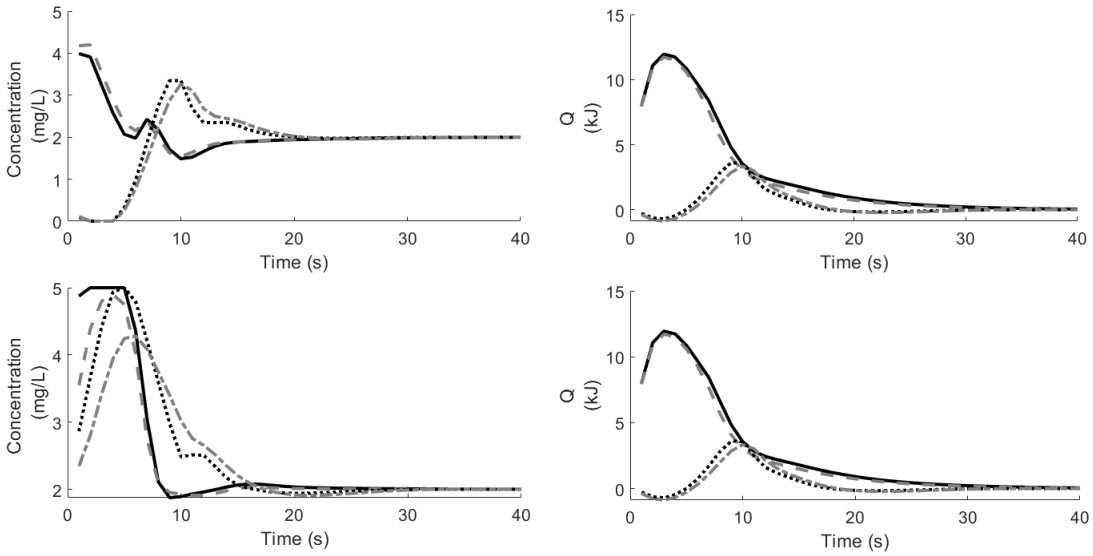


Figure 3.5: The input profiles for both CSTRs are shown with CSTR 1 inputs in the top two figures. The constrained model (-), constrained model without a D matrix (–), unconstrained model (:) and unconstrained model without a D matrix (-.) are all plotted against each other.

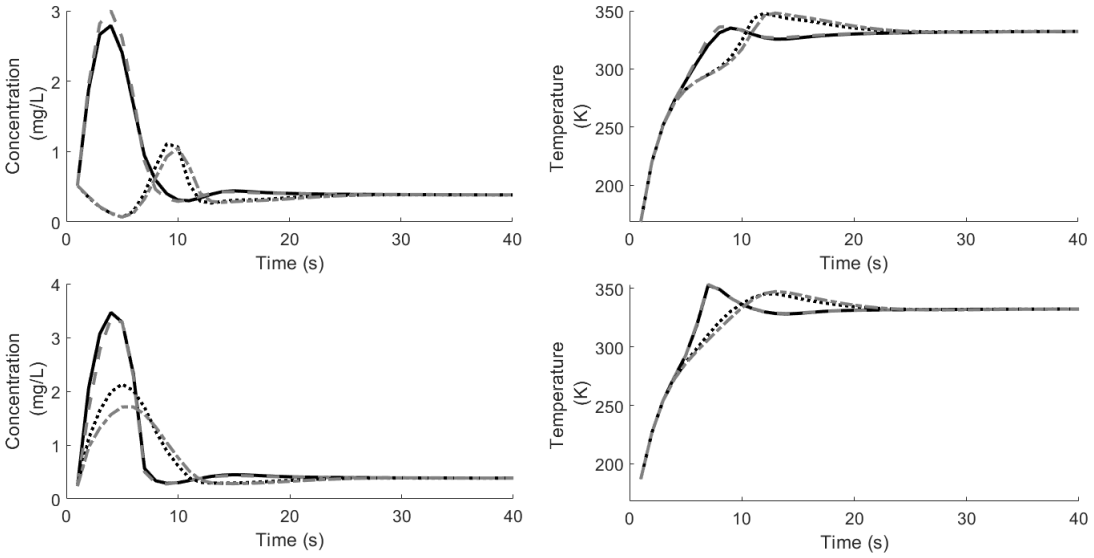


Figure 3.6: The output profiles for both CSTRs are shown with CSTR 1 outputs in the top two figures. The constrained model (-), constrained model without a D matrix (–), unconstrained model (:) and unconstrained model without a D matrix (-.) are all plotted against each other.

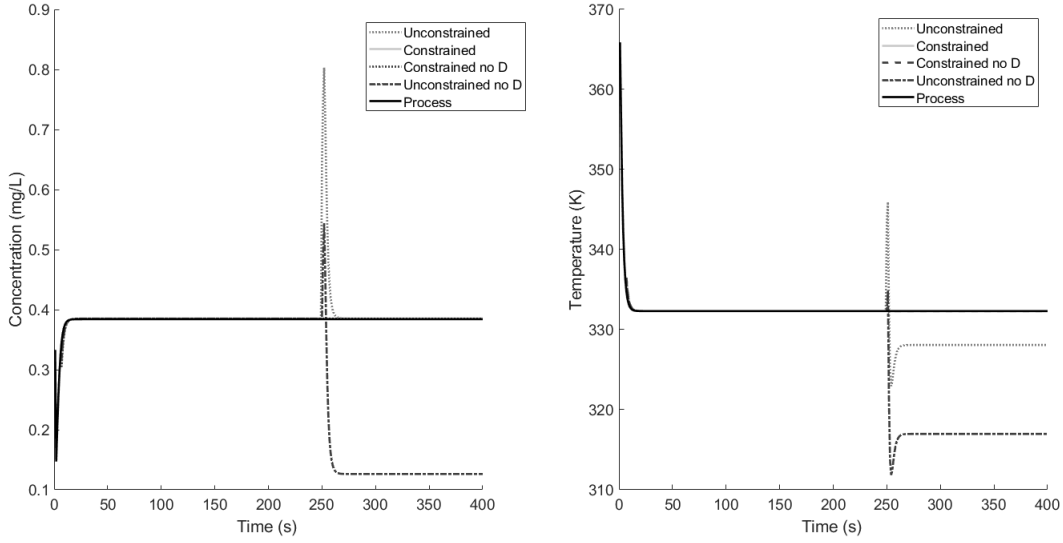


Figure 3.7: The output profiles for temperature and concentration in the first CSTR . The constrained model (-), constrained model without a D matrix (–), unconstrained model (:) and unconstrained model without a D matrix (-.) are all plotted against each other.

Table 3.3: The targeted validation prediction errors for each of the models.

Model	Error
Traditional	31.7719
Traditional with no D Matrix	124.6866
Constrained with no D matrix	9.5276e-4
Constrained	9.5073e-4

3.6 Conclusions

In this work, a new MPC formulation is proposed to handle a feed-through matrix. The MPC is then tested using an improved constrained subspace model in comparison to traditional subspace models. The constrained model is identified in a way that it has the correct process gains based on first principles allowing it to respond better to the controller. Additionally, the constrained model identified a feed-through matrix meaning the state space MPC must be appropriately modified to include the additional information. The use of the constrained subspace model in the modified state space MPC has shown improved performance in comparison to traditional approaches.

3.7 Acknowledgment

Financial support from the McMaster Advanced Control Consortium is gratefully acknowledged.

Bibliography

[1] Alenany, A., Shang, H., Soliman, M., and Ziedan, I. (2011). Improved subspace identification with prior information using constrained least squares. *IET Control Theory and Applications*, 5(13):1568–1576(8).

[2] Borrelli, F. and Morari, M. (2007). Offset free model predictive control. In *2007 46th IEEE Conference on Decision and Control*, pages 1245–1250. IEEE.

[3] Chen, H. and Allgöwer, F. (1998). Nonlinear model predictive control schemes with guaranteed stability. In *Nonlinear model based process control*, pages 465–494. Springer.

[4] Corbett, B. and Mhaskar, P. (2016). Subspace identification for data-driven modeling and quality control of batch processes. *AIChE Journal*, 62(5):1581–1601.

[5] Corbett, B. and Mhaskar, P. (2017). Data-driven modeling and quality control of variable duration batch processes with discrete inputs. *Industrial & Engineering Chemistry Research*, 56(24):6962–6980.

[6] Garg, A. and Mhaskar, P. (2017). Subspace Identification Based Modeling and Control of Batch Particulate Processes . *Industrial & Engineering Chemistry Research.* , 2017, Submitted.

[7] Ghosh, D., Hermonat, E., Mhaskar, P., Snowling, S., and Goel, R. (2019). Hybrid modeling approach integrating first-principles models with subspace identification. *Industrial & Engineering Chemistry Research*, 58(30):13533–13543.

[8] Hu, B., Zhao, Z., and Liang, J. (2012). Multi-loop nonlinear internal model controller design under nonlinear dynamic pls framework using arx-neural network model. *Journal of Process Control*, 22(1):207–217.

[9] Huang, B., Ding, S. X., and Qin, S. J. (2005). Closed-loop subspace identification: an orthogonal projection approach. *Journal of process control*, 15(1):53–66.

[10] Inoue, M., Matsubayashi, A., and Adachi, S. (2015). Moment-constrained subspace identification using a priori knowledge. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 2731–2736. IEEE.

[11] Larimore, W. E. (1996). Statistical optimality and canonical variate analysis system identification. *Signal Processing*, 52(2):131 – 144. Subspace Methods, Part II: System Identification.

[12] Ljung, L. (2002). Prediction error estimation methods. *Circuits, Systems and Signal Processing*, 21(1):11–21.

[MacGregor et al.] MacGregor, J. F., Jaeckle, C., Kiparissides, C., and Koutoudi, M. Process monitoring and diagnosis by multiblock pls methods. *AIChE Journal*, 40(5):826–838.

[14] Maeder, U., Borrelli, F., and Morari, M. (2009). Linear offset-free model predictive control. *Automatica*, 45(10):2214–2222.

[15] Mayne, D. Q., Rawlings, J. B., Rao, C. V., and Scokaert, P. O. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814.

[16] Mhaskar, P., El-Farra, N. H., and Christofides, P. D. (2005). Predictive control of switched nonlinear systems with scheduled mode transitions. *IEEE Transactions on Automatic Control*, 50(11):1670–1680.

[17] Mhaskar, P., El-Farra, N. H., and Christofides, P. D. (2006). Stabilization of nonlinear systems with state and control constraints using lyapunov-based predictive control. *Systems & Control Letters*, 55(8):650–659.

[18] Moonen, M., De Moor, B., Vandenberghe, L., and Vandewalle, J. (1989). On-and off-line identification of linear state-space models. *International Journal of Control*, 49(1):219–232.

[19] Muske, K. R. and Badgwell, T. A. (2002). Disturbance modeling for offset-free linear model predictive control. *Journal of Process Control*, 12(5):617–632.

[20] Muske, K. R. and Rawlings, J. B. (1993). Model predictive control with linear models. *AIChE Journal*, 39(2):262–287.

[21] Pannocchia, G. and Rawlings, J. B. (2003). Disturbance models for offset-free model-predictive control. *AIChE journal*, 49(2):426–437.

[22] Patel, N., Nease, J., Aumi, S., Ewaschuk, C., Luo, J., and Mhaskar, P. (2020). Integrating data-driven modeling with first-principles knowledge. *Industrial & Engineering Chemistry Research*, 59(11):5103–5113.

[23] Söderström, T., Stoica, P., and Friedlander, B. (1991). An indirect prediction error method for system identification. *Automatica*, 27(1):183–188.

[24] Trnka, P. and Havlena, V. (2009). Subspace like identification incorporating prior information. *Automatica*, 45(4):1086–1091.

[25] Van Overschee, P. and De Moor, B. (1994). N4sid: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30(1):75–93.

[26] Van Overschee, P. and De Moor, B. (1995). A unifying theorem for three subspace system identification algorithms. *Automatica*, 31(12):1853–1864.

[27] Venkat, A. N., Rawlings, J. B., and Wright, S. J. (2005). Stability and optimality of distributed model predictive control. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 6680–6685. IEEE.

[28] Verhaegen, M. and Dewilde, P. (1992). Subspace model identification part 2. analysis of the elementary output-error state-space model identification algorithm. *International journal of control*, 56(5):1211–1241.

[29] Zhao, Y., Huang, B., Su, H., and Chu, J. (2012). Prediction error method for identification of lpv models. *Journal of process control*, 22(1):180–193.

Chapter 4

Subspace Based Model Identification for Missing Data

This chapter presents a novel subspace identification approach for dealing with the problem of missing data in data-driven modeling approaches. The main problem is that certain modeling steps like singular value decomposition are reliant on a complete set of data to work. While this problem is typically solved through interpolation for small amounts of missing values, interpolation is not suited when dealing with multi-rate sampling or quality modeling scenarios where the amount of missing data is much higher. Recognizing that the non-iterative partial least squares algorithm is designed to handle regression with missing values, this work utilizes partial least squares and principal component analysis to generate a subspace model. The model generated from missing data was then compared against a traditional subspace model with both mean and linear interpolation for varying amounts of missing data. This work was completed in collaboration with Dr. Corbett as part of his graduate course where he provided guidance and technical expertise in creating the algorithm.

Patel, N., Mhaskar, P., & Corbett, B. (2020). Subspace based model identification for missing data. *AIChE Journal*, 66(10), e16538.

4.1 Abstract

This paper addresses the problem of missing process data in data-driven dynamic modeling approaches. The key motivation is to avoid using imputation methods or deletion of key process information when identifying the model, and utilizing the rest of the information appropriately at the model building stage. To this end, a novel approach is developed that adapts nonlinear iterative partial least squares (NIPALS) algorithms from both partial least squares (PLS) and principle component analysis (PCA) for use in subspace identification. Note that the existing subspace identification approaches often utilize singular value decomposition (SVD) as part of the identification algorithm which is generally not robust to missing data. In contrast, the NIPALS algorithms used in this work leverage the inherent correlation structure of the identification matrices to minimize the impact of missing data values while generating an accurate system model. Furthermore, in computing the system matrices, the calculated scores from the latent variable methods are utilized as the states of the system. The efficacy of the proposed approach is shown via simulation of a nonlinear batch process example.

4.2 Introduction

Batch processes are important for a wide range of manufacturing industries such as chemicals, polymers, specialty glass, ceramics, and steel production. Batch processes carry out a sequence or recipe, which can entail the addition of ingredients and executing processing steps. The recipe can be adjusted based on results from previous batches to maintain and promote quality control. [18] Additionally, the use of batch, instead of continuous processes, allows for batches that fail to meet the quality standards to be discarded without influencing other on-spec products. However, the high

value of these products means that discarding batches result in significant lost revenue. This motivates the need for advanced batch process control strategies which, in turn, often necessitates a good process model. Recent advances in data storage technology have resulted in increased availability of accessible historical process data, making data-driven modeling more viable than before. Data driven modeling techniques however, have to often deal with several challenges ranging from process nonlinearity to incomplete data.

One problem that is particularly common when dealing with historical data is that of missing data where for certain time instances, process data is not recorded. In some cases there are periods where an entire set of data is missing due to sensor failure or maintenance on the system. In other cases the measurements contain large errors and need to be removed from the data set. A common occurrence of missing data in chemical engineering processes is when different sensors have different sampling periods. Thus while there is continuous data from each sensor the measurements cannot be readily aligned with the other recorded variables.

Partial least squares (PLS) is a data-driven techniques that is often applied to industrial data, especially in cases for which missing data must be accounted for.[14, 23, 33] In this method, process data from each run is collected and projected into two lower dimensional subspaces (latent variable space). This ensures correct handling of correlated input and output space variables. Furthermore, the subspace are characterized by independent latent variables which can be used to understand relationships in the process.[MacGregor et al.] PLS accounts for missing data by exploiting covariance structures between related variables in the original variable space. This inherent ability to handle missing data is one of the attributes that makes PLS an attractive method for modeling batch processes.

One particularly successful approach to utilizing PLS techniques for modeling batch processes has been using the batchwise unfolding approach [12, MacGregor et al.] In

this approach, trajectory data from each batch is rearranged to form a single PLS observation and is subsequently related to quality outcomes. For on-line use, the approach uses missing data imputation methods to make predictions of the remaining batch trajectories. The efficacy of this approach has been well demonstrated in a variety of industrial cases. However, the approach necessitates so-called alignment of the batches to account for varying batch duration which can be challenging in practice. Furthermore, the method identifies a time-varying model, and at its core, does not distinguish between output and input variables, which limits its natural applicability for traditional model predictive control, and particularly applications where the process duration itself might be a decision variable.

To address these issues, new approaches adapting existing subspace identification techniques [22, 26, 15, 31] for batch processes have been proposed [5, 8]. The present contribution focuses on adapting a recently proposed batch subspace identification approach [5]. Recall that subspace identification is different to PLS in that it inherently distinguishes between process input and output variables. [22, 26, 15, 31] Subspace identification is carried out in two distinct steps in the first historical batch data is used to identify the state trajectory and the second step is to compute the system matrices using a least squares solution. Subspace identification algorithms utilize different techniques from canonical variate analysis[17, 29], numerical algorithms for subspace state space identification [30]and multivariate output error state space algorithms [32]. Subspace identification algorithms require singular value decomposition (SVD) of the matrices [22, 15]. In recent results, traditional subspace algorithm has been modified to use the same SVD method but for batch data with varying batch lengths. [6] Subspace identification methods, however, are unable to directly handle the problem of missing data since SVD requires matrices to be full rank. A method introduced by [21] utilizes sub-matrices that contain full rank data by rearranging the data. However, this isn't appropriate for handling process data since subspace identification relies on the time-dependent relationships in the data to identify a state

trajectory.

To handle missing data there are several techniques that can be used; the most popular being listwise deletion for regression based modeling.[7, 25] This method involves removing the entire observation from the data set if one variable is missing before carrying out any model identification procedure. Listwise deletion avoids the problem of having mismatched lengths in the process data however, it removes the remaining information which is accurately recorded. While this method is fairly popular due to low computational cost, for large amounts of missing data significant trends may be lost when the analysis is conducted.

Another common method to prepare data for model identification is to replace the missing data values by using mean substitution for regression based models. [24, 13, 27, 2] The purpose of mean substitution is to attempt to prevent the missing data point from affecting the analysis of that variable while keeping the information from the other variables which were accurately recorded. However, this approach does not consider the data trends at the time when the measurement went missing. In cases of sensor failure resulting from the process being out of the range utilizing a mean value at this time period will result in a value that is significantly different from the true process. This can lead to errors in the model identification procedure since the states of the system using the mean replacement value will be different from the true states.

Motivated by these considerations, this manuscript presents a different approach to subspace identification that readily enables handing of missing data. Existing subspace identification techniques handle missing data in a number of different ways. In addition to data imputation listed above other approaches carry out the prediction minimization algorithms using only available measurements.[3, 19] Other approaches such as subspace clustering are more computationally complex than the proposed approach and also do not readily allow for online applications.[1, 10] Specifically, the novel use of PCA then PLS and finally a PCA step in the subspace identification

approach allows for missing data values in the training data to be handled. While the use of PCA techniques in subspace identification approaches isn't a novel introduction (see [34, 16]) it's use in subspace identification with missing data has not been fully explored. The key advantage of using PCA and PLS techniques is that any missing data in the higher dimension has a minimal effect in the reduced dimensional space. The first step is to use latent variable methods (PCA followed by PLS) to identify a reduced dimensional space for the variables which accounts for missing data values. The second step replaces singular value decomposition with PCA to identify the states of the system. The rest of the paper is organized as follows: Section Preliminaries presents a Polymethyl Methacrylate (PMMA) process as the motivating example, followed by an overview of subspace identification methods. Section Model Identification presents the subspace identification approach using latent variable methods. Section Model Validation presents the validation approach, and clarifies the necessity of an appropriate state estimator. In Section Application to Motivating Example, an application of the proposed approach to the PMMA example is presented with missing data values. Finally, concluding remarks are made in Section Conclusions.

4.3 Preliminaries

4.3.1 Motivating Example: Polymethyl Methacrylate (PMMA) Process

PMMA is an important part of the polymer industry with applications ranging from glass substitutes to furniture. PMMA like most quality specific products is typically produced as a batch process. The key parameter in determining the quality of each batch is the molecular weight distribution of PMMA. However, measuring the entire molecular weight distribution is often not practical requiring an alternate set of

measurable outputs to be used instead. To that end the number and weight average molecular weights at the end of the batch can be used to quantify batch quality and are therefore, the targeted variables for control.

Batch, free-radical polymerization of PMMA is carried out in a jacketed reactor with a continuous stirrer. The reactor is charged with methyl methacrylate (monomer), AIBN (initiator), and toluene (solvent). The heat input to the reactor is adjusted based on the coolant/heating fluid flow rate. Table 4.1 presents a representative set of initial batch conditions for the PMMA process.

Table 4.1: Initial batch conditions for PMMA reaction

Variable	*Value*	σ^2	*Units*
Initiator Concentration	2.06×10^{-2}	1.0×10^{-3}	kg/m^3
Monomer Concentration	4.57	0.5	kg/m^3
Temperature	334.15	2.5	K

The PMMA process in this work is described using a first principles dynamic model consisting of nine states governed by a set of nonlinear algebraic and differential equations. The nine states of the system are as follows: reactor temperature, monomer and initiator concentrations and six molecular weight distribution moments. The dynamic PMMA model is taken from [9] while making suitable adaptations [11, 28] with additional explanations of these adaptations provided in [4].

To reflect feed variations, the initial conditions for each batch are selected from a normal distribution around the nominal initial condition. Table 4.1 shows both the nominal initial condition value and the standard deviation in the initial conditions. In this example the measured variables are reactor temperature, reactant volume and stirrer torque. The volume measurements are used to determine the density which can be considered to represent the extent of the reaction. Therefore, the batch can be terminated based on a desired density set point instead of a set batch length. In the present manuscript, the first principles model for the PMMA process will be utilized

as a test bed to demonstrate the missing data handling capability of the proposed subspace identification algorithm.

4.3.2 Subspace Identification

This subsection provides a brief description of existing subspace identification methods. These techniques identify a state space model (typically) around steady-state operating conditions using only input and output data. Specifically, given a series of s measurements of the input $u_k \in \mathbb{R}^p$ and the output $y_k \in \mathbb{R}^q$, the objective is to identify a deterministic system of order n of the following form:

$$\mathbf{x}_{k+1}^d = \mathbf{A}\mathbf{x}_k^d + \mathbf{B}\mathbf{u}_k, \tag{4.1}$$

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k^d + \mathbf{D}\mathbf{u}_k, \tag{4.2}$$

where the identification approach should reveal the order n of this unknown system and the system matrices $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times p}$, $\mathbf{C} \in \mathbb{R}^{q\times n}$, $\mathbf{D} \in \mathbb{R}^{q\times p}$ (up to within a similarity transformation). Consider now data from a batch process where k is the sampling instant since the batch initiation and b denotes the batch index. Then the output Hankel submatrix for a batch b is constructed in the same fashion as a 'standard' Hankel matrix, given by:

$$\mathbf{Y}_{1|i}^{(b)} = \begin{bmatrix} \mathbf{y}^{(b)}[1] & \mathbf{y}^{(b)}[2] & \cdots & \mathbf{y}^{(b)}\left[j^{(b)}\right] \\ \vdots & \vdots & & \vdots \\ \mathbf{y}^{(b)}[i] & \mathbf{y}^{(b)}[i+1] & \cdots & \mathbf{y}^{(b)}\left[i+j^{(b)}-1\right] \end{bmatrix} \quad \forall\, b = 1, \ldots, nb \tag{4.3}$$

where nb is the number of batches used for identification.

Utilizing the Hankel matrix as presented above does not factor data taken from multiple batches. Additionally, a simple concatenation of data from multiple batches

would suggest that the states across multiple batches are linked which is clearly not the case. Thus, the key to batch subspace identification is to build a pseudo-Hankel matrix which enables data across batches to be used without the assumption that the end of one batch is the beginning of another. The pseudo-Hankel matrix is generated by horizontally concatenating the individual Hankel sub-matrices for each batch to form a matrix for both input and output data. The pseudo-Hankel matrix takes the form:

$$\mathbf{Y}_{1|i} = \begin{bmatrix} \mathbf{Y}_{1|i}^{(1)} & \mathbf{Y}_{1|i}^{(2)} & \cdots & \mathbf{Y}_{1|i}^{(nb)} \end{bmatrix} \tag{4.4}$$

Similarly, pseudo-Hankel matrices for input data are formed. This approach for handling multiple batches satisfies the requirements of subspace identification in addition to not requiring the batches to have identical lengths.

Construction of appropriate pseudo-Hankel matrices for input and output enables determination of state trajectories using any of the preexisting subspace identification algorithms available in the literature (such as the deterministic algorithm [22] utilized in this work). A consequence of concatenation of the Hankel sub-matrices is that the identified state trajectories will also be comprised of similarly concatenated state estimates for each training batch. Mathematically the identified state trajectory matrix can be represented as:

$$\hat{\mathbf{X}}_{i+1}^{(b)} = \begin{bmatrix} \hat{\mathbf{x}}^{(b)}[i+1] & \cdots & \hat{\mathbf{x}}^{(b)}\left[i+j^{(b)}\right] \end{bmatrix} \quad \forall\, b = 1, \ldots, nb \tag{4.5}$$

$$\hat{\mathbf{X}}_{i+1} = \begin{bmatrix} \hat{\mathbf{X}}_{i+1}^{(1)} & \hat{\mathbf{X}}_{i+1}^{(2)} & \cdots & \hat{\mathbf{X}}_{i+1}^{(nb)} \end{bmatrix} \tag{4.6}$$

where nb is the total number of batches used for identification.

With the state trajectory matrix, the system matrices can be estimated easily using

methods such as ordinary least squares which is shown below:

$$\mathbf{Y}_{reg}^{(b)} = \begin{bmatrix} \hat{\mathbf{x}}^{(b)}[i+2] & \cdots & \hat{\mathbf{x}}^{(b)}\left[i+j^{(b)}\right] \\ \mathbf{y}^{(b)}[i+1] & \cdots & \mathbf{y}^{(b)}\left[i+j^{(b)}-1\right] \end{bmatrix} \tag{4.7}$$

$$\mathbf{X}_{reg}^{(b)} = \begin{bmatrix} \hat{\mathbf{x}}^{(b)}[i+1] & \cdots & \hat{\mathbf{x}}^{(b)}\left[i+j^{(b)}-1\right] \\ \mathbf{u}^{(b)}[i+1] & \cdots & \mathbf{u}^{(b)}\left[i+j^{(b)}-1\right] \end{bmatrix} \tag{4.8}$$

$$\begin{bmatrix} \mathbf{Y}_{reg}^{(1)} & \cdots & \mathbf{Y}_{reg}^{(nb)} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{X}_{reg}^{(1)} & \cdots & \mathbf{X}_{reg}^{(nb)} \end{bmatrix} \tag{4.9}$$

The system matrices A,B,C, and D make up the subspace model of the system and can now be used for control.

4.3.3 PLS

This subsection provides a brief overview of the nonlinear iterative partial least squares (NIPALS) regression techniques which are utilized in the proposed approach. The purpose of PLS regression is to achieve the best explanation of both the X-space and the Y-space while maximizing the relationship between the two spaces. Inherently PLS techniques do not require a distinction between the output "Y" block and the input "X" block and thus can be arranged in any sequence. However, in order to get meaningful results it is important to mean-center the observations. Thus, given an $X \in R^{n \times k}$ and $Y \in R^{n \times m}$ block (where n is the number of observations, k is the number of variables in the X block, and m is the number of variables in the Y block), at the training stage (or determination of the PLS model), after mean centering and scaling the data, for a choice of the number of latent variables A, the loadings, scores and weighting are computed. The NIPALS algorithm to determine the model parameters is presented in the form of a pseudo code below:

First, set $X_0 = X$, $Y_0 = Y$

for a = 1,2..A

1. Start with arbitrary initial column $u_a \in R^{n\times 1}$
2. *while the scores t_a continues to change*
   (a) Regress columns from X_{a-1} onto u_a to get weightings $w_a \in R^{k\times 1}$
   (b) Normalize the weightings w_a to have unit length
   (c) Regress the rows from X_{a-1} onto w_a to get scores $t_a \in R^{n\times 1}$
   (d) Regress columns from Y_{a-1} onto t_a to get weightings $c_a \in R^{m\times 1}$
   (e) Regress the rows from Y_{a-1} onto c_a to get scores u_a
   (f) Go back to step (a) using the new u_a
3. Deflate component from X_{a-1} and Y_{a-1}, i.e.,
   (a) Regress columns from X_{a-1} onto t_a to get loadings p_a
   (b) $X_a = X_{a-1} - t_a p_a^T$
   (c) $Y_a = Y_{a-1} - t_a c_a^T$

end for

The resultant vectors t_a, w_a c_a and u_a can be stacked together and denoted by $T \in R^{n\times A}$, $W \in R^{k\times A}$, $C \in R^{m\times A}$, and $U \in R^{n\times A}$. After a PLS model has been computed, for a new $X_{new} \in R^{l\times k}$, where l is the new number of observations of the K variables, the X block is first mean centered and scaled, and then the variables in the $\hat{Y}_{new} \in R^{l\times m}$ are predicted as follows: define $W^* \equiv W(P^TW)^{-1}$ and then compute $T_{new} = X_{new}^T W^*$, and $\hat{Y}_{new}^T = T_{new}^T C'$, and finally $\hat{Y}_{new}^T$ is uncentered and unscaled to predict the observations.

4.4 Proposed Modeling Approach

4.4.1 Model Identification

The proposed adaptation of the subspace model identification approach also utilizes the Hankel matrices presented in Eq. 4.4 to compute the state sequence, and then uses the ordinary least squares approach as with the traditional subspace identification technique (see Eq. 4.9). The key adaptation is in how the state trajectories are computed using the Hankel matrices, in a way that readily enables handling missing data.

Having defined these matrix arrangements of the available observations recall that we can rewrite Eq. 4.1 and Eq. 4.2 as:

$$\mathbf{X}_{1|i+1} = \mathbf{A}^i\mathbf{X_1} + \mathbf{\Delta_i}\mathbf{U}_{1|i}, \tag{4.10}$$

$$\mathbf{Y}_{1|i} = \mathbf{\Gamma_i}\mathbf{X_1} + \mathbf{H_i}\mathbf{U}_{1|i}, \tag{4.11}$$

where $\mathbf{A}^i$ is the dynamic matrix A to the ith power. $\mathbf{\Delta_i}$, $\mathbf{\Gamma_i}$ and $\mathbf{H_i}$ can all be calculated by using iterative substitution of Eq. 4.1 and Eq. 4.2 (see [22] for a detailed explanation). Using the pseudo-inverse (denoted $\bullet^*$) Eq. 4.11 to isolate for X_1 and then substituting in Eq. 4.10 gives:

$$\mathbf{X}_{i+1} = \begin{bmatrix} \Delta_i - \mathbf{A}^i\mathbf{\Gamma_i^*}\mathbf{H_i} & \mathbf{A}^i\mathbf{\Gamma_i^*} \end{bmatrix} \begin{bmatrix} \mathbf{U}_{1|i} \\ \mathbf{Y}_{1|i} \end{bmatrix} \tag{4.12}$$

Next the following matrices can be defined where $\mathbf{W}_{1|i}$ represents everything that is

known and $\mathbf{L}_i$ represents everything that must be solved for.

$$\mathbf{L_i} = \begin{bmatrix} \mathbf{\Delta_i} - \mathbf{A}^i\mathbf{\Gamma}^*_\mathbf{i}\mathbf{H_i} & \mathbf{A}^i\mathbf{\Gamma}^*_\mathbf{i} \end{bmatrix} \tag{4.13}$$

$$\mathbf{W}_{1|i} = \begin{bmatrix} \mathbf{U}_{1|i} \\ \mathbf{Y}_{1|i} \end{bmatrix} \tag{4.14}$$

Eq. 4.12 can now be expressed as:

$$\mathbf{X}_{i+1} = \mathbf{L_i}\mathbf{W}_{1|i} \tag{4.15}$$

Rewriting Eq. 4.11 with respect to future outputs $\mathbf{Y}_{i+1|2i}$ and substituting in Eq. 4.15.

$$\mathbf{Y}_{i+1|2i} = \mathbf{\Gamma_i}\mathbf{L_i}\mathbf{W}_{1|i} + \mathbf{H_i}\mathbf{U}_{i+1|2i} \tag{4.16}$$

In order to implement the next step (which forms the basis of the novel contribution), we recall the definition of projecting a matrix $B \in R^{n\times A}$ onto another matrix $A \in R^{n\times A}$:

$$\mathbf{A}/\mathbf{B}^\perp = \mathbf{A}(\mathbf{1} - \mathbf{B^T}(\mathbf{B}\mathbf{B^T})^{-\mathbf{1}}\mathbf{B}) \tag{4.17}$$

The next step in the proposed approach is to project the future inputs onto the future outputs in order to remove the known correlations from the data.

$$\mathbf{Y}_{i+1|2i}/\mathbf{U}^{\perp}_{i+1|2i} = \mathbf{Y}_{i+1|2i}(1 - \mathbf{U}^{T}_{i+1|2i}(\mathbf{U}_{i+1|2i}\mathbf{U}^{T}_{i+1|2i})^{-1}\mathbf{U}_{i+1|2i}) \tag{4.18}$$

$$\mathbf{Y}_{i+1|2i}/\mathbf{U}^{\perp}_{i+1|2i} = (\mathbf{\Gamma_i}\mathbf{L_i}\mathbf{W}_{1|i} + \mathbf{H_i}\mathbf{U}_{i+1|2i})(1 - \mathbf{U}^{T}_{i+1|2i}(\mathbf{U}_{i+1|2i}\mathbf{U}^{T}_{i+1|2i})^{-1}\mathbf{U}_{i+1|2i}) \tag{4.19}$$

$$\begin{aligned}\mathbf{Y}_{i+1|2i}/\mathbf{U}^{\perp}_{i+1|2i} = (\mathbf{\Gamma_i}\mathbf{L_i}\mathbf{W}_{1|i})(1 - \mathbf{U}^{T}_{i+1|2i}(\mathbf{U}_{i+1|2i}\mathbf{U}^{T}_{i+1|2i})^{-1}\mathbf{U}_{i+1|2i}) + \\ (\mathbf{H_i}\mathbf{U}_{i+1|2i})(1 - \mathbf{U}^{T}_{i+1|2i}(\mathbf{U}_{i+1|2i}\mathbf{U}^{T}_{i+1|2i})^{-1}\mathbf{U}_{i+1|2i})\end{aligned} \tag{4.20}$$

We recognize that the second term in Eq. 4.20 is a projection of the future inputs, $\mathbf{U}^{\perp}_{i+1|2i}$, onto a space perpendicular to its row space which yields the zero matrix by definition. Thus, this equation simplifies as follows:

$$\mathbf{Y}_{i+1|2i}/\mathbf{U}^{\perp}_{i+1|2i} = \mathbf{\Gamma_i}\mathbf{L_i}\mathbf{W}_{1|i}\mathbf{U}^{\perp}_{i+1|2i} \tag{4.21}$$

The following projection of future outputs and past inputs and outputs onto the orthogonal component of the future inputs is the first step of the proposed approach in the subsection on the NIPALS algorithm for Subspace Identification, where NIPALS algorithm is used to get the orthogonal projection of the future inputs instead of existing techniques.

$$(\mathbf{Y}_{i+1|2i}/\mathbf{U}^{\perp}_{i+1|2i})^{T} = (\mathbf{W}_{1|i}\mathbf{U}^{\perp}_{i+1|2i})^{T}\mathbf{L_i^T}\mathbf{\Gamma_i^T} \tag{4.22}$$

For ease of notation, we define $\tilde{\mathbf{W}}_{1|i} = \mathbf{W}_{1|i}\mathbf{U}^{\perp}_{i+1|2i}$ and $\tilde{\mathbf{Y}}_{i+1|2i} = \mathbf{Y}_{i+1|2i}/\mathbf{U}^{\perp}_{i+1|2i}$. This is used to describe the deflated matrices from the PLS technique described below. Then Eq.4.22 can be re-written as:

$$\tilde{\mathbf{Y}}_{i+1|2i} = \tilde{\mathbf{W}}^{T}_{1|i}\mathbf{L}^{\mathbf{T}}_{\mathbf{i}}\mathbf{\Gamma}^{\mathbf{T}}_{\mathbf{i}} \tag{4.23}$$

In order to identify the state trajectory from Eq. 4.23 it first needs to be rearranged.

$$\beta = \mathbf{L}^{\mathbf{T}}_{\mathbf{i}}\mathbf{\Gamma}^{\mathbf{T}}_{\mathbf{i}} \tag{4.24}$$

$$\beta^{T} = \mathbf{\Gamma}_{\mathbf{i}}\mathbf{L}_{\mathbf{i}} \tag{4.25}$$

$$\beta^{T}\mathbf{W}_{1|i} = \mathbf{\Gamma}_{\mathbf{i}}\mathbf{L}_{\mathbf{i}}\mathbf{W}_{1|i} \tag{4.26}$$

$$\tag{4.27}$$

Then using Eq. 4.15

$$\beta^{T}\mathbf{W}_{1|i} = \mathbf{\Gamma}_{\mathbf{i}}\mathbf{X}_{i+1} \tag{4.28}$$

The above sequence of equations represents the basis for the proposed approach and the NIPALS algorithm to carry out these steps is presented as follows.

NIPALS algorithm for Subspace Identification

In this section we will describe how the equations presented above can be translated into a series of three NIPALS algorithms steps to determine the state trajectory followed by a linear regression step to identify the system matrices.

The procedure begins by projecting data perpendicular to future inputs. Specifically,

both the future inputs and past inputs and outputs are projected perpendicular to future inputs. For future outputs, this is motivated by the understanding that future outputs depend only on the state and future inputs. Therefore, by projecting the future outputs perpendicular to the future inputs, the projected values should be strongly related to the state. For past inputs and outputs, under open-loop control, if the system has been perfectly excited, there should theoretically be no covariance between the past inputs and outputs and future inputs. As such, projecting perpendicular to the future inputs should not substantially change these blocks of data. However, in cases where the system has not been fully excited (ie closed-loop identification data) this step removes covariance and exposes the independent variation that exists, permitting identification.

Remark 18. *This paper does not focus on the substantial issues involved in closed loop identification. As noted in previous work [6], batch processes under trajectory tracking control may provide sufficient excitation without directly introducing identification inputs such as pseudo random binary sequences. This observation is the justification for the application example presented in this paper. However, as noted above, the authors recognize the inherent potential of the proposed methods for closed-loop identification. In future work, this possibility will be explored explicitly. One area of particular interest is how squared prediction error statistics from the latent spaces identified in this method can be used to enforce model validity on when the models are applied in model predictive control.*

The projection of future outputs and past inputs and outputs perpendicular to future inputs is represented in Eq. 4.22 in the factor $(\mathbf{W}_{1|i}\mathbf{U}^{\perp}_{i+1|2i})^T$. This can be achieved using a NIPALS algorithm adapted from the NIPALS PCA algorithm as follows:

1. Initialize $t_{uf} \in R^{n\times 1}$ with the first column of $\mathbf{U}^T_{i+1|2i} \in R^{n\times j}$

2. Repeat until convergence in t_{uf}:

   (a) Regress columns of $\mathbf{U}_{i+1|2i}^T$ onto scores t_{uf} to get loadings $\mathbf{P}_{uf} \in R^{n\times 1}$

   (b) Normalize the loadings $\mathbf{P}_{uf}$ to have unit length

   (c) Regress rows of $\mathbf{U}_{i+1|2i}^T$ onto loadings to get updated scores t_{uf}

3. Deflate future inputs by subtracting what has been explained in t_{uf}:

$$\mathbf{U}_{i+1|2i}^T = \mathbf{U}_{i+1|2i}^T - t_{uf}P_{uf}^T \tag{4.29}$$

4. Regress columns of $\mathbf{W}_{1|i} \in R^{n\times p}$ onto t_{uf} to get loadings $P_W \in R^{p\times 1}$

5. Normalize loadings P_W to have unit length

6. Regress columns of $\mathbf{Y}_{i+1|2i} \in R^{n\times q}$ onto t_{uf} to get loadings $C_{yf} \in R^{q\times 1}$

7. Deflate past inputs, past outputs, and future outputs using the scores calculated in steps 1 to 3 as follows:

$$\tilde{\mathbf{W}}_{1|i} = \mathbf{W}_{1|i} - t_{uf}P_w^T \tag{4.30}$$

$$\tilde{\mathbf{Y}}_{i+1|2i} = \mathbf{Y}_{i+1|2i} - t_{uf}C_{yf}^T \tag{4.31}$$

8. Repeat steps 1 to 7 using the deflated matrices from the previous iteration until all variance in the the future inputs, $\mathbf{U}_{i+1|2i}^T$, is explained (ie. the deflated matrix doesn't change with each iteration).

where j is the number of inputs, p is the number of inputs times the number of outputs and q is the number of outputs.

The steps described above remove all variance from $\mathbf{W}_{1|i}$ and $\mathbf{Y}_{i+1|2i}$ that can be explained (is correlated) with future inputs therefore, the remaining relationship is the

effect of the current states on the future outputs. This is mathematically equivalent to projecting data perpendicular to the future inputs as described in Eq. 4.22. Note that steps1 to 3 are the standard NIPALS algorithm for calculating components of PCA models. The efficient convergence of this algorithm is know to be guaranteed as long as the initial guess for t_{uf} is non-zero. Steps 4 to 7 are similar to the standard deflation step of the $\mathbf{X}$ matrix in NIPALS PLS (see section PLS for details) and are non-iterative.

Remark 19. *One of the most beneficial aspects of the NIPALS algorithm is the ability to elegantly handle missing data. Specifically, in every regression step (ie steps 2a, 2c, 4, and 6) any row (observation) that has a missing value may be excluded. Because of the relatively low leverage of each observation in the overall model fit, excluding the missing observations does not negatively impact the calculated regression coefficients. The stability of NIPALS for PCA on datasets with missing data has been well studied.[23] . This ability uses the covariance of columns (variables) in the* $\mathbf{X}$ *space to effectively impute missing values. Note that the format of the data-blocks used in the proposed method guarantees a high degree of covariance because of the selected Hankel unfolding and the time-series nature of the data. This result is further used in each of the following NIPALS steps in the proposed methodology.*

Having projected past inputs, past outputs and future outputs perpendicular to future inputs, the next step in the proposed approach is to identify the relationship between past inputs and outputs and future outputs. Identifying this relationship is the fundamental objective of subspace identification as, given the form of the state-space model, this relationship is directly related to the underlying states. To identify this relationship using NIPALS, the standard NIPALS PLS algorithm is applied (see

subsection on PLS) on the following X_{PLS} and Y_{PLS} matrices:

$$X_{PLS} = \tilde{\mathbf{W}}_{1|i}$$
$$Y_{PLS} = \tilde{\mathbf{Y}}_{i+1|2i}$$

Note that the final model form of a PLS model (for prediction) is the linear form:

$$\hat{\mathbf{Y}}_{PLS} = \mathbf{X}_{PLS}\beta \tag{4.32}$$

Consequently, the PLS model described above is sufficient to describe the relationship laid out in Eq. 4.23 where it follows that $\beta = \mathbf{L}_i^T \mathbf{\Gamma}_i^T$. By taking a transpose and multiplying by $\mathbf{W}_{1|i}$ we arrive at:

$$\beta^T \mathbf{W}_{1|i} = \mathbf{\Gamma_i L_i W}_{1|i} \tag{4.33}$$
$$\tag{4.34}$$

By further substituting the from Eq. 4.15 we get:

$$\beta^T \mathbf{W}_{1|i} = \mathbf{\Gamma_i X}_{i+1} \tag{4.35}$$

In Eq. 4.35 the left hand side is entirely comprised of known values (coefficients from PLS and past inputs and outputs). Therefore, if the left hand side can be appropriately decomposed into matrices of the correct dimensions to match the right hand side, the result is a valid state-space representation. One way of achieving this, as described by [22] is to use SVD. However, most SVD algorithms are not suited for use with missing data. An alternative in this case is to use the NIPALS algorithm for PCA. (The mathematical relationship between PCA and SVD is well known.)

Conducting PCA on the transpose of the left hand side of Eq. 4.35, ie $\mathbf{X_{PCA}} = \mathbf{W}_{1:i}^T \beta$

we get:

$$\mathbf{W}_{1:i}^{T}\beta = \mathbf{T}_{pca}\mathbf{P}_{pca}^{T} + \mathbf{E}_{pca} \tag{4.36}$$

Therefore it follows that the scores calculated in this PCA step are a valid realization of the states of the system, $\mathbf{X}_{i+1} = \mathbf{T}_{pca}^{T}$. Furthermore, the loadings are related to the matrix $\mathbf{\Gamma_i}$. Notice that number of states (system order) is equivalent to the number of components used in the PCA model. The resulting matrix from the proposed algorithm results in a state trajectory matrix $\mathbf{X}_{i+1}$ being determined which can be used in Eq. 4.9 to compute the system matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ using standard linear regression.

4.4.2 Model Validation

Model validation is an integral part of the model identification procedure to ensure that the identified model is able to accurately predict on-line. In this step batches that were not part of the training data and do not contain any missing observations are used to test the model's ability to predict process outputs. In order to conduct a more rigorous validation test missing data was not considered in the validation batch. While it is possible for missing observations to occur on-line the missing observation would only result in one fewer data point when comparing validation error. If there are missing observations in the data before the Luenberger observer has converged then the estimation technique will have to be appropriately modified to handle missing data and it is expected the observer will take longer to converge. In the present work, we focus on the ability of the proposed approach to identify a good model for the process, and these specific illustrations remain the subject of future work.

During the initial part of the batch a Luenberger observer is used to estimate the states of the system until the model output predictions are within a predefined tolerance of

current process outputs. This is used as an indication that the current state of the model is similar to the current state of the system. Once the observer has determined the current state of the system the identified subspace model can be used to predict the remainder of the batch to determine the validation error. The observer has the following form:

$$\hat{\mathbf{x}}[k+1] = \mathbf{A}\hat{\mathbf{x}}[k] + \mathbf{B}\mathbf{u}[k] + \mathbf{L}(\mathbf{y}[k] - \hat{\mathbf{y}}[k]) \tag{4.37}$$

where $\mathbf{L}$ is the observer gain and is chosen to ensure that $(\mathbf{A} - \mathbf{LC})$ is stable. Additionally, the poles of the observer are placed in the positive quadrant of the unit circle to ensure stability. The initial state estimate, which can be chosen to be any value, is given as the initial state identified through traditional subspace techniques. Note that any state estimation technique, such as the Kalman filter, can be used and the Luenberger observer is just one example.

4.5 Application to the Motivating Example

To show the effectiveness of the proposed approach at handling missing data three different cases will be considered for the PMMA simulation example. Case 1 will contain training data that is not missing any observations, Case 2 will contain training data with random missing data from 5% up to 30% and Case 3 will contain a block of missing data from 5% up to 30%.

4.5.1 Model Identification

In this scenario, 15 training batches, under proportional integral (PI) control, are generated from the PMMA process and used to identify three subspace models. The first is identified using traditional subspace identification with mean replacement to impute the missing values, the second is identified using traditional subspace identification with linear interpolation to impute the missing values and the third is the proposed approach which uses PCA/PLS techniques. Note that the key novelty in the present approach is in dealing with missing data in the context of building dynamic models (and not in the context of missing data when building PCA models). In particular, when filling out missing data to identify dynamic models, the order of data (because it is indexed with time) is relevant- and thus a direct implementation of existing techniques for filling data in the context of PCA alone is not the most suited. In contrast, replacing the missing values with mean values of variables over the entire batch, especially when variables change significantly over the duration of the batch- may not be the best strategy (and ideal benchmarking) either. Thus, a comparison with missing data being replaced by linear interpolation is considered. The models identified using mean replacement, linear interpolation based replacement, and the proposed approach are then validated against a new batch in order to determine the validation error. The error is calculated as the difference between the process and the model predictions for the duration that the models are used for prediction (i.e., the Luenberger observer has converged). The validation error is also normalized by the number of the predictions to account for the observer converging at different times between the two models. The validation error for the batch is calculated as follows:

$$ValidationError = \frac{\sum(\hat{y} - y_{process})}{\#ofpredictions} \tag{4.38}$$

Figure 4.1 shows a typical output profile with random missing data entries. While Figure 4.2 shows the output profile with blocks of missing data entries. The gaps in the solid lines indicate that an observation is missing; note that the missing data observations can occur at any point in the profile. This data set with all of the missing variables is used as a training set to build a subspace model using PCA/PLS techniques. This is the same process output profile that the subspace model is attempting to predict except it utilizes mean replacement to fill in the missing observations.

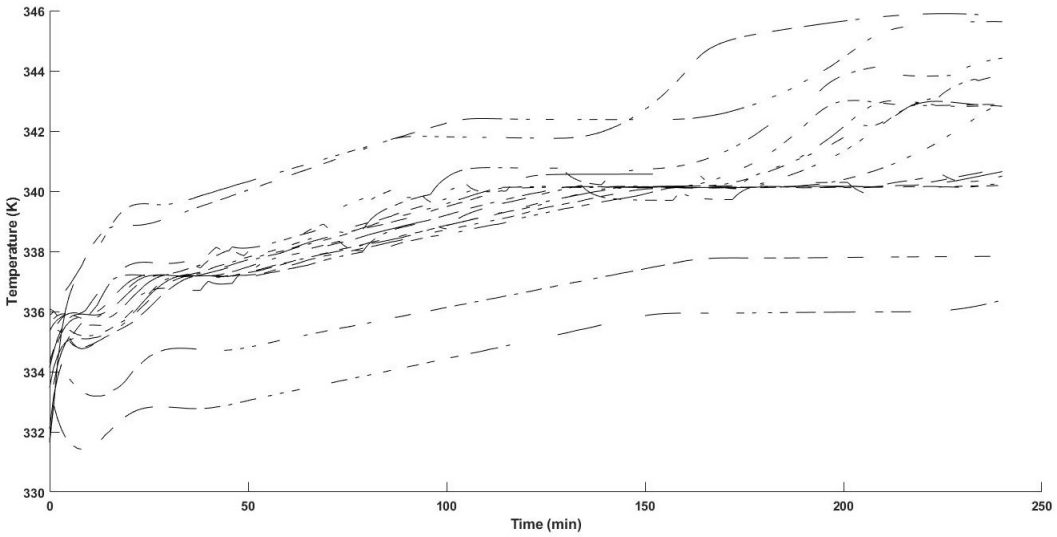


Figure 4.1: The output training data for the temperature output with 30% randomly missing data from all training batches.

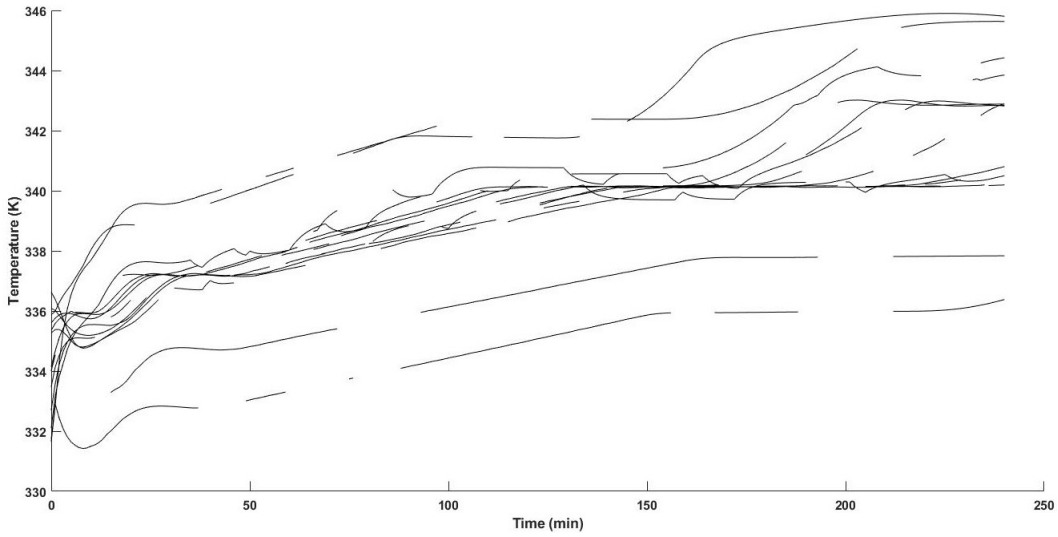


Figure 4.2: The output training data for the temperature output with 30% missing data in blocks from all training batches.

4.5.2 Case 1: No Missing Data

Case 1 represents the base case to demonstrate how the proposed approach compares against existing subspace identification techniques. Table 4.2 shows the fit error between the traditional subspace model and the model obtained from the proposed approach. The proposed technique has a lower fit error without missing data showing that the proposed approach generates a model that is comparable to the existing model identification approaches. The two models used identical training data, with the same number of states and Hankel matrices. Thus, the lower fit error is an important result highlighting the effectiveness of the proposed approach.

Figure 4.3 shows the accuracy of the model when compared against a typical validation batch. Without any missing data the proposed approach is still able to generate a model that is capable of accurately predicting the process and is a suitable starting point to handle the missing data problem.

Table 4.2: Validation error for the two models without missing data

Model	*Fit Error*
Traditional	2.1980
Proposed	1.2770

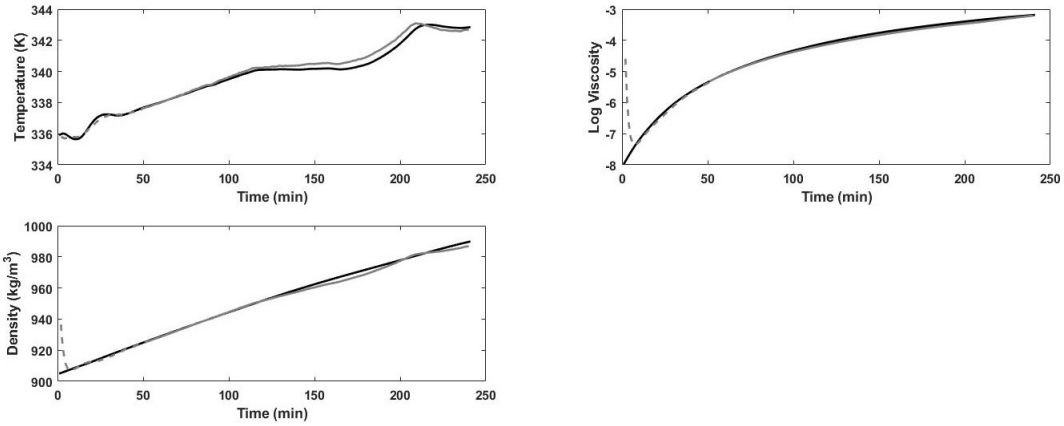


Figure 4.3: The output predictions from the new identified model for each output (temperature, viscosity and density). The dashed grey line shows the predictions until the state observer converged and then the grey line shows the predictions from the model compared against the process (black).

4.5.3 Case 2: Random Missing Data

In this case data from the training batches was randomly deleted in order to achieve missing data percentages between 5% and 30%. Data entries were removed from each batch in different variables and observations to ensure there was no bias in the training set. The proposed approach was then used to identify a subspace model. For comparison traditional subspace identification was carried out using mean replacement and linear interpolation for the missing values. The models were then validated based on validation batches which did not contain any missing observations.

Table 4.3 shows the fit error between the traditional subspace model and the model obtained from the proposed approach. The proposed approach is able to consistently predict more accurately compared to the traditional approach. As expected the tra-

ditional subspace method using mean replacement incorrectly identifies the process since it uses the dataset shown in Figure 4.1 with mean replacement used to impute the missing values (thus the dynamic model having to fit to large 'jumps' in values as a result of filling in the same mean value for all missing data points). Similarly even using linear interpolation to identify the missing values does not result in an accurate representation of the process since it relies on the gradient provided by the available data points, and forces an assumption that the variables evolve linearly with time over that duration. Figure 4.4 shows the predictions of the proposed approach along with the predictions using the Luenberger observer for the first 50 time-steps. When comparing the fit of the proposed approach against Figure 4.5 which is the traditional approach with mean replacement, the proposed approach predicts closer to the true process. It is important to note that while the linear interpolation validation shown in 4.6 is a better compared to mean replacement the proposed approach has the best validation results.

Table 4.3: Validation error for each of the models with random missing data

Model	*Error 5%*	*Error 10%*	*Error 15%*	*Error 20%*	*Error 25%*	*Error 30%*
Mean Replacement	3.1849	6.8222	6.5922	7.4813	8.7168	10.5306
Linear Interpolation	3.9284	4.01278	4.4872	4.5833	5.0355	5.0550
Proposed	1.4748	1.2755	1.2991	1.2027	1.3979	1.6336
Lags	7	7	7	7	7	7

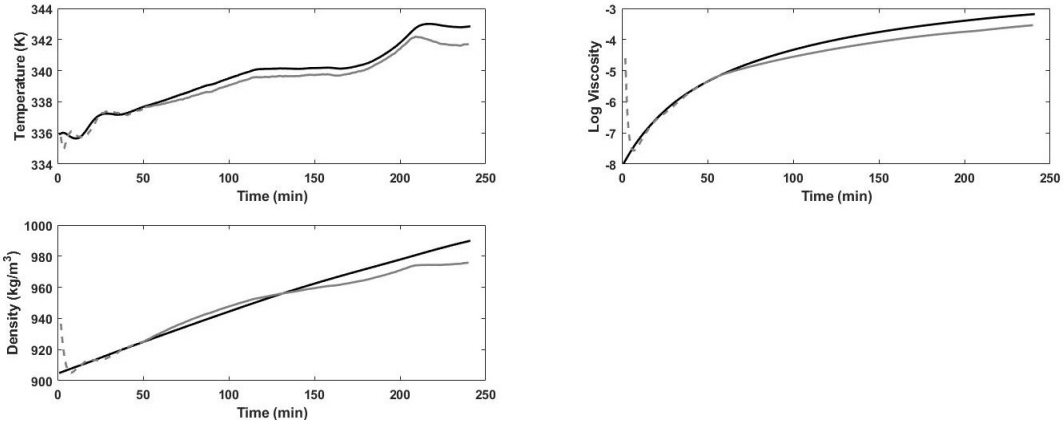


Figure 4.4: The output predictions from the new identified model with 30% missing data for each output (temperature, viscosity and density). The dashed grey line shows the predictions until the state observer converged and then the grey line shows the predictions from the model compared against the process (black).

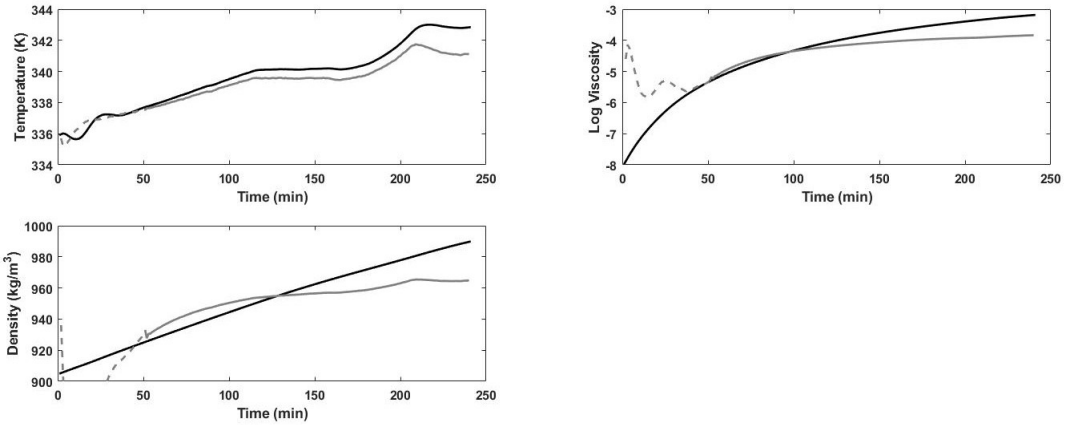


Figure 4.5: The output predictions from the traditional subspace model using mean replacement with 30% missing data for each output (temperature, viscosity and density). The dashed grey line shows the predictions until the state observer converged and then the grey line shows the predictions from the model compared against the process (black).

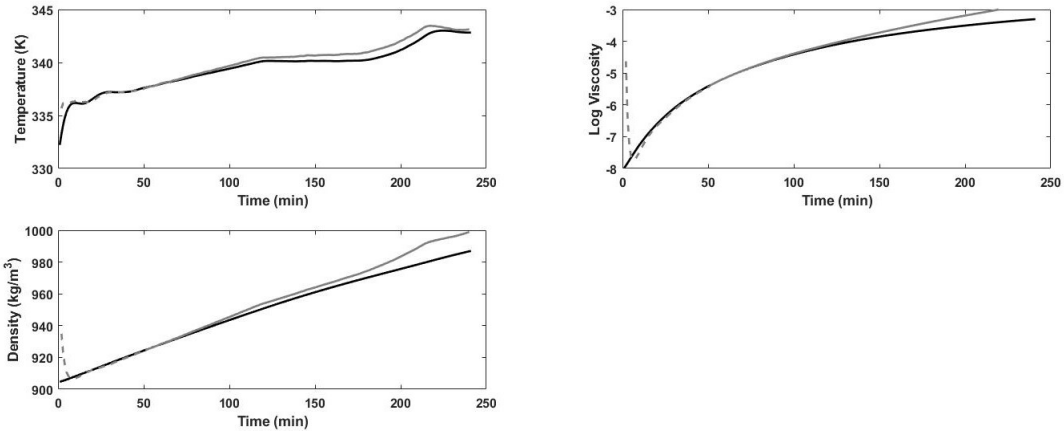


Figure 4.6: The output predictions from the traditional subspace model using linear interpolation with 30% missing data for each output (temperature, viscosity and density). The dashed grey line shows the predictions until the state observer converged and then the grey line shows the predictions from the model compared against the process (black).

4.5.4 Case 3: Block Missing Data

In this case data from the training batches was randomly deleted in blocks so that an entire sequence of observations from a variable was missing from the batch. The block size was uniformly distributed with a mean of 10 observations and a standard deviation of 3 observations and the block observations were randomly deleted throughout the entire data set. For comparison, traditional subspace identification was carried out using mean replacement and traditional subspace was carried out using linear interpolation for the missing values. The models were then validated based on separate validation batches which did not contain any missing observations.

Table 4.4 shows the error using the traditional subspace model and the model obtained from the proposed approach. The proposed approach is able to consistently predict more accurately compared to the traditional approach using both mean replacement and linear interpolation. As expected linear interpolation results provided a lower validation error compared to the mean replacement approach however, the proposed

approach still has the smallest error. Figure 4.7 shows the predictions of the proposed approach along with the predictions using the Luenberger observer for the first 50 time-steps of one validation batch. When comparing the fit of the proposed approach against Figure 4.8 which is the traditional approach with mean replacement, it can be clearly seen that the proposed approach predictions are closer to the true process outputs. The linear interpolation validation shown in 4.9 provides better predictions compared to the mean replacement approach however, the proposed approach results in the best model.

Table 4.4: Validation error for each of the models with block missing data

Model	*Error 5%*	*Error 10%*	*Error 15%*	*Error 20%*	*Error 25%*	*Error 30%*
Mean Replacement	4.3018	6.0013	6.667	7.5709	8.3631	9.2156
Linear Interpolation	4.4767	3.1831	3.0405	3.0843	3.1881	2.3737
Proposed	2.9856	2.4233	2.8942	3.0653	2.7573	1.8549
Lags	7	7	7	7	7	16

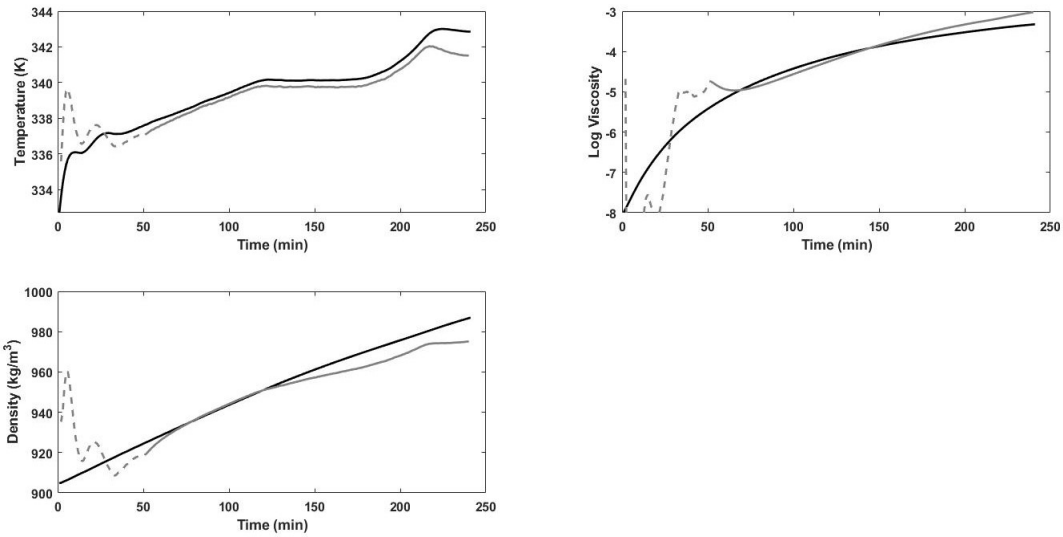


Figure 4.7: The output predictions from the new identified model with 30% missing data for each output (temperature, viscosity and density). The dashed grey line shows the predictions until the state observer converged and then the grey line shows the predictions from the model compared against the process (black).

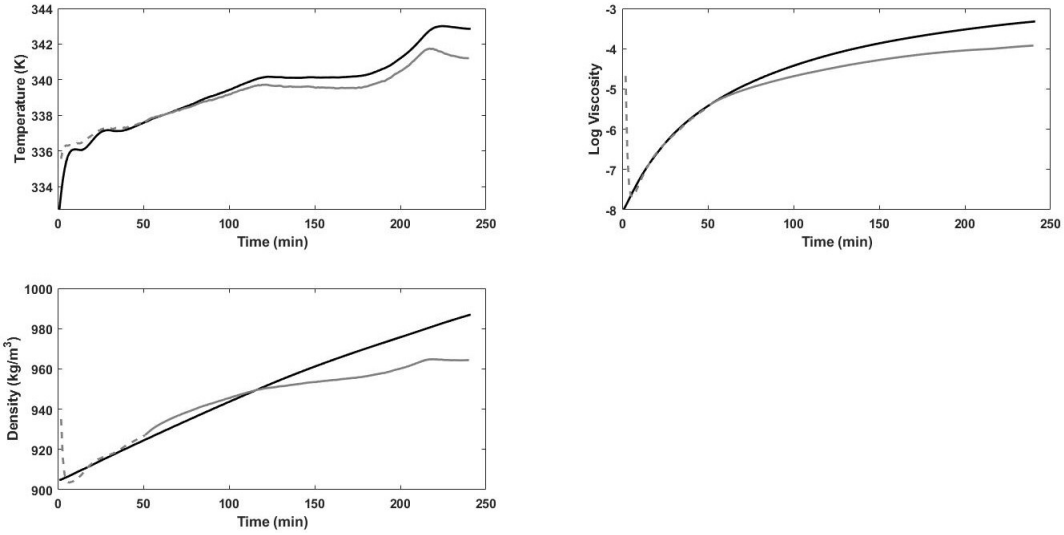


Figure 4.8: The output predictions from the traditional subspace model using mean replacement with 30% missing data for each output (temperature, viscosity and density). The dashed grey line shows the predictions until the state observer converged and then the grey line shows the predictions from the model compared against the process (black).

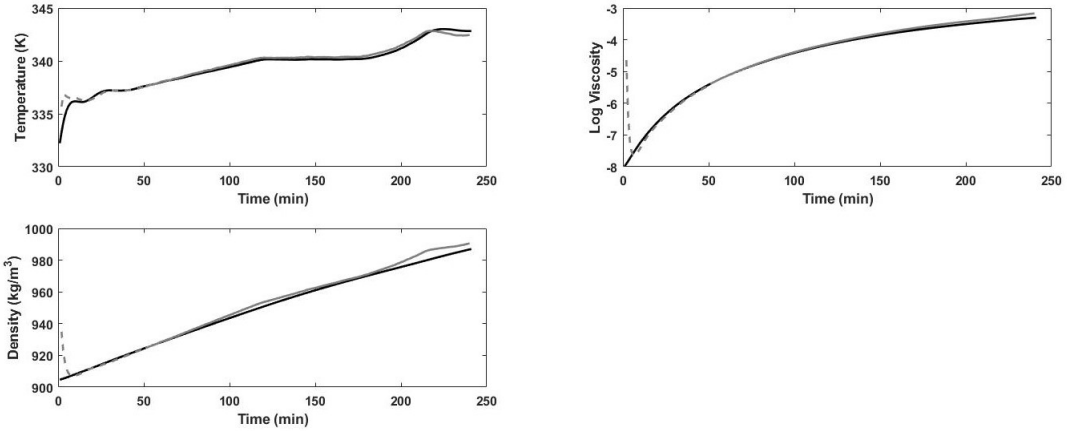


Figure 4.9: The output predictions from the traditional subspace model using linear interpolation with 30% missing data for each output (temperature, viscosity and density). The dashed grey line shows the predictions until the state observer converged and then the grey line shows the predictions from the model compared against the process (black).

4.6 Conclusions

In this work, a novel subspace identification procedure for batch processes was proposed with the goal of handling missing data. The identified model is compared to traditional subspace techniques using mean replacement and linear interpolation when handling both random and block missing data structures as demonstrated using simulation results for a batch PMMA reaction. We are able to demonstrate favorable results when comparing against existing mean replacement and linear interpolation techniques for all cases of missing data ranging from 5% to 30%. The results obtained by the proposed approach are explicable by the benefits of utilizing regression techniques to handle missing data.

4.7 Acknowledgment

Financial support from the Ontario Graduate Scholarship and the McMaster Advanced Control Consortium is gratefully acknowledged.

Bibliography

[1] Balzano, L., Szlam, A., Recht, B., and Nowak, R. (2012). K-subspaces with missing data. pages 612–615.

[2] Bennett, D. A. (2001). How can i deal with missing data in my study? *Australian and New Zealand journal of public health*, 25(5):464–469.

[3] Chen, J., Huang, B., Ding, F., and Gu, Y. (2018). Variational bayesian approach for arx systems with missing observations and varying time-delays. *Automatica*, 94:194–204.

[4] Corbett, B., Macdonald, B., and Mhaskar, P. (2013). Model Predictive Quality Control of Polymethyl Methacrylate. *IEEE Transactions on Control Systems Technology*, 23(2):3948–3953.

[5] Corbett, B. and Mhaskar, P. (2016). Subspace identification for data-driven modeling and quality control of batch processes. *AIChE Journal*, 62(5):1581–1601.

[6] Corbett, B. and Mhaskar, P. (2017). Data-driven modeling and quality control of variable duration batch processes with discrete inputs. *Industrial & Engineering Chemistry Research*, 56(24):6962–6980.

[7] Dodeen, H. M. (2003). Effectiveness of valid mean substitution in treating missing data in attitude assessment. *Assessment & Evaluation in Higher Education*, 28(5):505–513.

[8] Dorsey, A. W. and Lee, J. H. (2003). Building inferential prediction models of batch processes using subspace identification. *Journal of Process Control*, 13(5):397–406.

[9] Ekpo, E. and Mujtaba, I. M. (2008). Evaluation of neural networks-based controllers in batch polymerisation of methyl methacrylate. *Neurocomputing*, 71(7-9):1401–1412.

[10] Eriksson, B., Balzano, L., and Nowak, R. (2011). High-rank matrix completion and subspace clustering with missing data. *arXiv preprint arXiv:1112.5629*.

[11] Fan, S., Gretton-Watson, S., Steinke, J., and Alpay, E. (2003). Polymerisation of methyl methacrylate in a pilot-scale tubular reactor: modelling and experimental studies. *Chemical engineering science*, 58(12):2479–2490.

[12] Flores-Cerrillo, J. and MacGregor, J. F. (2004). Control of batch product quality by trajectory manipulation using latent variable models. *Journal of Process Control*, 14(5):539–553.

[13] Graham, J. W., Cumsille, P. E., and Shevock, A. E. (2012). Methods for handling missing data. *Handbook of Psychology, Second Edition*, 2.

[14] Hu, B., Zhao, Z., and Liang, J. (2012). Multi-loop nonlinear internal model controller design under nonlinear dynamic pls framework using arx-neural network model. *Journal of Process Control*, 22(1):207–217.

[15] Huang, B., Ding, S. X., and Qin, S. J. (2005). Closed-loop subspace identification: an orthogonal projection approach. *Journal of process control*, 15(1):53–66.

[16] Jiang, Q., Yan, X., and Huang, B. (2015). Performance-driven distributed pca process monitoring based on fault-relevant variable selection and bayesian inference. *IEEE Transactions on Industrial Electronics*, 63(1):377–386.

[17] Larimore, W. E. (1996). Statistical optimality and canonical variate analysis system identification. *Signal Processing*, 52(2):131 – 144. Subspace Methods, Part II: System Identification.

[18] Lee, K. S. and Lee, J. H. (2003). 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*, 13(7):607 – 621. Selected Papers from the sixth IFAC Symposium on Bridging Engineering with Science - DYCOPS - 6.

[19] Liu, Z., Hansson, A., and Vandenberghe, L. (2013). Nuclear norm system identification with missing inputs and outputs. *Systems & Control Letters*, 62(8):605–612.

[MacGregor et al.] MacGregor, J. F., Jaeckle, C., Kiparissides, C., and Koutoudi, M. Process monitoring and diagnosis by multiblock pls methods. *AIChE Journal*, 40(5):826–838.

[21] Markovsky, I. (2013). Exact system identification with missing data. In *52nd IEEE Conference on Decision and Control*, pages 151–155. IEEE.

[22] Moonen, M., De Moor, B., Vandenberghe, L., and Vandewalle, J. (1989). On-and off-line identification of linear state-space models. *International Journal of Control*, 49(1):219–232.

[23] Nelson, P. R., Taylor, P. A., and MacGregor, J. F. (1996). Missing data methods in pca and pls: Score calculations with incomplete observations. *Chemometrics and intelligent laboratory systems*, 35(1):45–65.

[24] Pallant, J. (2013). *SPSS survival manual*. McGraw-Hill Education (UK).

[25] Pigott, T. D. (2001). A review of methods for missing data. *Educational research and evaluation*, 7(4):353–383.

[26] Qin, S. J. (2006). An overview of subspace identification. *Computers and Chemical Engineering*, 30(10-12):1502–1513.

[27] Raaijmakers, Q. A. (1999). Effectiveness of different missing data treatments in surveys with likert-type data: Introducing the relative mean substitution approach. *Educational and Psychological Measurement*, 59(5):725–748.

[28] Rho, H.-J., Huh, Y.-J., and Rhee, H.-K. (1998). Application of adaptive model-predictive control to a batch mma polymerization reactor. *Chemical Engineering Science*, 53(21):3729–3739.

[29] Shang, L., Liu, J., Turksoy, K., Shao, Q. M., and Cinar, A. (2015). Stable recursive canonical variate state space modeling for time-varying processes. *Control Engineering Practice*, 36:113–119.

[30] Van Overschee, P. and De Moor, B. (1994). N4sid: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30(1):75–93.

[31] Van Overschee, P. and De Moor, B. (1995). A unifying theorem for three subspace system identification algorithms. *Automatica*, 31(12):1853–1864.

[32] Verhaegen, M. and Dewilde, P. (1992). Subspace model identification part 2. analysis of the elementary output-error state-space model identification algorithm. *International journal of control*, 56(5):1211–1241.

[33] Walczak, B. and Massart, D. (2001). Dealing with missing data: Part i. *Chemometrics and Intelligent Laboratory Systems*, 58(1):15–27.

[34] Wang, J. and Qin, S. J. (2002). A new subspace identification approach based on principal component analysis. *Journal of process control*, 12(8):841–855.

Chapter 5

Polymethyl Methacrylate Quality Modeling with Missing Data Using Subspace Based Model Identification

This chapter addresses the issue with batch quality modeling which is one of the key issues in batch control. Traditionally batch processes are controlled using trajectory tracking of an optimal recipe that has been tested numerous times with success. However, due to disturbances and different initial conditions this does not always lead to optimal batch quality or production. Another issue is that quality variables require additional testing and are available at very low frequencies making their incorporation into modeling approaches difficult. To overcome these issues this manuscript uses the missing data algorithm from the previous chapter to build a combined quality and output model. This manuscript demonstrates the improved performance in comparison to interpolation techniques. This manuscript was completed with the help of an undergraduate research student Kavita Sivanathan who helped run experiments to collect the data.

Patel, N., Sivanathan, K., & Mhaskar, P. (2021). Polymethyl Methacrylate Quality

Modeling with Missing Data Using Subspace Based Model Identification. *Processes, 9*(10), 1691.

5.1 Abstract

This paper addresses the problem of quality modeling in polymethyl methacrylate (PMMA) production. The key challenge is handling the large amounts of missing quality measurements in each batch due to the time and cost sensitive nature of the measurements. To this end, a missing data subspace algorithm that adapts nonlinear iterative partial least squares (NIPALS) algorithms from both partial least squares (PLS) and principal component analysis (PCA) is utilized to build a data driven dynamic model. The use of NIPALS algorithms allows for the correlation structure of the input-output data to minimize the impact of the large amounts of missing quality measurements. These techniques are utilized in a simulated case study to successfully model the PMMA process in particular, and demonstrate the efficacy of the algorithm to handle the quality prediction problem in general.

5.2 Introduction

Polymethyl methacrylate (PMMA) is an important industrial polymer with widespread applications ranging from plastics to electronics. PMMA production occurs via a specific recipe in a batch process in order to ensure the optimal product quality. The key parameter to determining product quality is the molecular weight distribution of PMMA. However, measuring the entire molecular weight is a long, complex procedure, therefore the number and weight average molecular weights are used as outcome measures instead. These quality variables are measured at the end of the batch and, as such, must be controlled tightly.

In order to achieve good quality control of the PMMA process, it is important to develop a process model capable of handling batch data. Batch processes are unique

in that they tend towards the low volume production of high value products, as this allows for poor quality batches to be discarded.[2] Each discarded batch represents a significant loss in revenue, motivating the need for advanced batch control approaches and, fundamentally, the development of accurate quality models. Recent advances in computing technology have led to increased amounts of historical data being available, making data-driven modeling a viable choice for model identification. [21, 12, 26, 15, 9, 23] However, these techniques still have challenges when dealing with nonlinear processes and scenarios where data is missing, as is the case with quality variables.

An important consideration when identifying data-driven models is the choice of input and output variables. While the distinction is typically based on variables that are controlled in comparison to those that are measured, the batch process also introduces a separate type of output variable: quality variables. Quality variables are still considered to be outputs from the process, however, they are not measured continuously, nor are they always measured online. Quality measurements are often calculated based on the regular measured outputs or are determined by separate analyses of the batch. This difference is an important consideration for data driven modeling techniques, as one assumption that is prevalent in process modeling scenarios is that the inputs and outputs are sampled at a single and uniform sampling rate. In practice, industrial processes often have different sampling rates for input and output variables. Additionally, processes record quality measurements at an even slower rate compared to traditional inputs and outputs. This leads to scenarios where inputs and outputs have some missing values due to differences in sampling rates, whereas quality measurements are available at extremely low frequencies. This presents a challenge to traditional data driven modeling approaches that require complete data sets to identify a model.

When attempting to model industrial batch processes using data containing missing observations, a common data-driven technique used is partial least squares (PLS). [8] In this technique, process data from multiple batches is taken and projected into two

lower dimensional subspaces (latent variable space). This ensures that the relationship between the correlated input and output space variables is maintained and is characterized by the independent latent variables.[13] PLS techniques are capable of handling missing data since they utilize the covariant structure between the input and output variables from the original variable space. This inherent ability is one of the key properties that makes PLS techniques suitable for modeling batch processes. The application of PLS techniques to batch process modeling has been previously explored and one successful approach is to utilize batchwise unfolding of the data. [6, 7, 13] Process data from each batch is unfolded into a single PLS observation that is subsequently related to quality variables. This approach can be applied to on-line process data by utilizing data imputation techniques to make predictions on the missing data observations. While this approach has been well-documented in handling industrial batch data, it requires batch alignment in order to account for varying batch duration. Furthermore, the approach identifies a time-varying model and, since PLS inherently doesn't distinguish between input and output variables, it has limited applicability in traditional model predictive control and scenarios where batch duration is a decision variable.

Another technique that is suitable for building process models is subspace identification [15, 18, 9, 23], which has been appropriately modified for handling batch data using Hankel matrices. [2] Note that subspace identification is different from PLS because it explicitly distinguishes between input and output variables. [15, 18, 9, 23] Subspace identification consists of two distinct steps: identifying a state trajectory from historical input and output batch data, and determining the system matrices using a least squares solution. To achieve these outcomes, subspace identification utilizes a range of techniques from canonical variate analysis [11, 20], numerical algorithms [22] and multivariate output error state space algorithms. [24] One common technique to subspace identification is singular value decomposition (SVD) of the matrices. [15, 9] However, SVD requires matrices to be full-rank, making it unsuitable

for handling batch data with missing observations. [14]

One way to handle the problem of missing data with subspace identification is through a lifting technique.[3] The prevalent assumption is that all the inputs are sampled at one frequency and all the outputs are sampled at another frequency. A model can then be derived using the grouped inputs and outputs wherever the sampling frequencies align. This process involves computed control moves over the larger sampling interval instead of every sampling instance. Lifting has many advantages, including the ability to convert a multi-rate system into a single rate system, but the key benefits lie in model predictive control. However, this process is unsuited for the quality control problem, as quality measurements are not frequent enough to provide sufficient excitation for modeling. These measurements are only available at percentages less than 10%, making a single rate system impractical. Thus, a modeling approach that can use the lower frequency sampling rate of the inputs and outputs is required. The previous work [16] developed a technique for handling missing data in subspace identification approach with a focus on solving the problem of random missing data. The present work recognizes that the missing data subspace approach can be applied to the broader quality measurement induced missing data problem. Thus, the present manuscript provides a solution for instances where the low frequency of quality measurements in relation to other online measurements results in a missing data problem (as in the simulated PMMA process example).

To overcome these challenges and model the PMMA process, the current paper focuses on a recent modification to subspace identification [17] that treats nonuniform sampling rate data as a missing data problem. Specifically, the addition of PCA and PLS steps to the subspace identification approach permits accounting for the missing quality measurements. While the use of PCA and PLS techniques is not a novel introduction to model identification (see [25, 10]), the reduced latent variable space is marginally affected by missing data, thus allowing for the treatment of nonuni-

form sampling problems as a case of randomly missing data. The first step is to use latent variable methods (PCA followed by PLS) to identify a reduced dimensional space for the variables which accounts for missing observations. The second step replaces SVD with PCA to identify the states of the system, whereupon traditional subspace approaches can be utilized. The rest of the paper is organized as follows: Section Preliminaries presents the Polymethyl Methacrylate (PMMA) process, followed by a review of traditional subspace identification. The next subsection presents the Subspace Identification approach that can readily handle missing data issues. In the results section, identification and validation results for the PMMA process are presented. Finally, concluding remarks are made.

5.3 Preliminaries

5.3.1 Polymethyl Methacrylate (PMMA) Process Description

PMMA is produced via free-radical polymerization in a jacketed reactor with a continuous stirrer. At the start of each batch, the reactor is charged with methyl methacrylate (monomer), AIBN (initiator), and toluene (solvent). The system is controlled by manipulating the heat input to the reactor, which is based on the coolant/heating fluid flow rate. Table 5.1 presents the nominal set of batch conditions used to initialize the PMMA process with standard deviation of σ^2. In order to provide good quality batches, a nominal temperature trajectory known to yield good quality results is tracked for each batch. Proportional-integral (PI) tracking is used to track the batch temperature by manipulating the jacket temperature. The batch is then terminated after the polymer density reaches the desired set-point.

Table 5.1: Initial batch conditions for PMMA reaction

Variable	*Value*	σ^2	*Units*
Initiator Concentration	2.06×10^{-2}	1.0×10^{-3}	kg/m^3
Monomer Concentration	4.57	0.5	kg/m^3
Temperature	61	2.5	°C

A first principles dynamic model is utilized as a test bed consisting of nine states governed by a set of nonlinear algebraic and differential equations. The nine system states are: reactor temperature, monomer and initiator concentrations and six molecular weight distribution moments. This dynamic PMMA model has been taken from [4] with suitable adaptations [5, 19] included [1]. To replicate natural process variability in each batch, the initial conditions of the PMMA process are varied using a normal distribution around the nominal conditions as shown in Table 5.1. Three variables are assumed to be measured continuously: reactor temperature, reactant volume and stirrer torque. The volume measurements are used to calculate the density, which is used to measure the extent of the reaction, allowing the batch to be terminated once the reaction has reached a certain threshold, as opposed to terminating at a fixed batch length. This leads to variable batch lengths depending on the time required to reach the target density.

5.3.2 Subspace Identification

This subsection provides a brief description of existing subspace identification methods that provide the basis for the PCA and PLS based modifications. These techniques utilize input/output data to identify a state space model (typically) around steady-state operating conditions. Specifically, given a batch of s measurements of the inputs $u_k \in \mathbb{R}^p$ and outputs $y_k \in \mathbb{R}^q$, a system with order n can be determined in the following

form:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k, \tag{5.1}$$

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{D}\mathbf{u}_k, \tag{5.2}$$

where the identification approach identified the system order n along with the system matrices $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times p}$, $\mathbf{C} \in \mathbb{R}^{q\times n}$, $\mathbf{D} \in \mathbb{R}^{q\times p}$ (up to within a similarity transformation). Considerable work has been done to show how this model can be developed for batch processes. Thus, consider k as the sampling instant from the start of the batch and b denoting the batch number [2]. The identified state trajectory matrix for each batch is represented as:

$$\mathbf{X}_{i+1}^{(b)} = \begin{bmatrix} \mathbf{x}^{(b)}[i+1] & \cdots & \mathbf{x}^{(b)}\left[i+j^{(b)}\right] \end{bmatrix} \quad \forall\, b = 1, \ldots, nb \tag{5.3}$$

$$\mathbf{X}_{i+1} = \begin{bmatrix} \mathbf{X}_{i+1}^{(1)} & \mathbf{X}_{i+1}^{(2)} & \cdots & \mathbf{X}_{i+1}^{(nb)} \end{bmatrix} \tag{5.4}$$

where nb is the total number of batches used for identification.

Using the state trajectory matrix, the system matrices can be estimated easily using the following matrix regression steps:

$$\mathbf{Y}_{reg}^{(b)} = \begin{bmatrix} \mathbf{x}^{(b)}[i+2] & \cdots & \mathbf{x}^{(b)}\left[i+j^{(b)}\right] \\ \mathbf{y}^{(b)}[i+1] & \cdots & \mathbf{y}^{(b)}\left[i+j^{(b)}-1\right] \end{bmatrix} \tag{5.5}$$

$$\mathbf{X}_{reg}^{(b)} = \begin{bmatrix} \mathbf{x}^{(b)}[i+1] & \cdots & \mathbf{x}^{(b)}\left[i+j^{(b)}-1\right] \\ \mathbf{u}^{(b)}[i+1] & \cdots & \mathbf{u}^{(b)}\left[i+j^{(b)}-1\right] \end{bmatrix} \tag{5.6}$$

$$\begin{bmatrix} \mathbf{Y}_{reg}^{(1)} & \cdots & \mathbf{Y}_{reg}^{(nb)} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{X}_{reg}^{(1)} & \cdots & \mathbf{X}_{reg}^{(nb)} \end{bmatrix} \tag{5.7}$$

5.3.3 Noniterative Partial Least Squares (NIPALS) Algorithm

This subsection provides a brief description of the existing NIPALS based PCA and PLS subspace identification approach. These steps are used to carry out regressions in the traditional subspace approach that rely on full rank matrices, since PCA and PLS can handle missing data. This approach is fully described in Patel et al. [16].

The first step in subspace identification is to project the future outputs and past inputs and outputs perpendicular to future inputs to remove the correlation between the past data and future inputs. This can be achieved using a NIPALS algorithm for PCA. The PCA step removes all of the variance associated with the future inputs, and the remaining correlations are the result of the effect of current states on future outputs. The next step in the NIPALS approach is to identify the relationship between the past inputs and outputs and the future outputs. This relationship provides the basis of subspace identification techniques, as it relates directly to the underlying states of the process. In order to identify this relationship, a NIPALS algorithm for PLS is applied. Here, the two blocks are chosen such that the first block contains the past inputs and outputs and the second block contains the future outputs. Finally, after carrying out PLS, the first block contains only known values (coefficients from the past inputs and outputs). This matrix can then be appropriately decomposed to matrices to match the second block, resulting in a valid state space representation. In traditional subspace, the relationship is then identified through singular value decomposition (SVD). However, SVD is not suitable for handling matrices with missing data, and PCA can be used instead. It then follows that the scores from PCA are a valid realization of the process states and can then be used in Eq. 5.7 to compute the system matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ via linear regression.

Remark 20. *This approach is particularly important when considering the quality control problem, and opens up the possibility of direct control of quality.*

Thus, since the model can be utilized to predict quality variables during the intervals which it is not measured, a classical control structure such as PID can be utilized to directly control the (predicted) quality to a desired trajectory, terminating at the desired final quality.

5.4 PMMA Model Identification

Two case studies are considered for the PMMA process, where the output data is lost in storage (thus, the PI controller is able to run in the closed-loop using the available temperature measurements. Note that the approach and implementation readily holds for the situation where the measurements would simply be unavailable at random times, and would need a change in how the PI controller is implemented). Case 1 contains training data with 30% random missing output data and quality data measurements retained at intervals of 1 in every 10, 20 and 30 time steps, respectively. Case 2 contains training data with quality measurement available every 10 time steps and random missing output data ranging from 5% up to 30%. Thus, the effects of both lower frequency of quality measurement availability and increasing missing data are independently investigated.

Remark 21. *The key difference between data imputation approaches and the missing data approach is that imputation can lead to inconsistency between the imputed data (in terms of the model structure implied) and the model structure being identified. Thus, in the comparison case study, where linear interpolation techniques are utilized, it is seen that linear interpolation is not consistent with the type of model ultimately being identified (linear time invariant dynamic model), resulting in the inability to capture the process trends sufficiently, leading to poor model performance. The success of the approach is based on the format of the data-blocks, which guarantees a high degree of covariance*

due to batch unfolding of the Hankel matrices and the temporal nature of the data.

5.4.1 Model Identification

For each case, 15 training batches under proportional integral (PI) control are generated from the PMMA process and are used to identify two subspace models. To compare the present approach against existing results, first a model is identified using traditional subspace identification with linear interpolation of the measured outputs to determine missing values in the training batches and, subsequently, the proposed approach using PCA/PLS techniques to handle infrequent quality sampling is implemented. The two models are then validated against a new set of 15 batches (under PI control) that are also modified to have missing data similar to the training batches to determine the validation error.

Remark 22. *Note that the number of states and lags are a user specified parameter. The choice is based on the best training fit defined as the lowest error between the predictions and the process outputs. The states and lags used are specific to the present implementation and future work will explore further the relationship between the number of states and model quality.*

5.4.2 State Observer

After identifying the state space model an important consideration before utilizing it for online predictions is to ensure that the model states converge with the process. This approach utilizes the Luenberger observer, which updates the states using feedback from the outputs weighted using a gain matrix L in the following method:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{L}(y_{process} - \hat{y}) \tag{5.8}$$

To handle missing outputs when the model is run with a new batch, this approach uses linear interpolation until the process outputs converge within a tolerance of 0.1% or until 40 minutes have passed (the sampling instance when this occurs is denoted by c). After that, the model is run online and is used to predict the process outputs. The model's effectiveness can then be determined based on the prediction error until the end of the batch. The error is determined by the sum of the difference between each available process output ($y_{process}$) and model prediction ($\hat{y}$) for the duration that the models are used for prediction (i.e., after the Luenberger observer has converged).

This value is then normalized by the number of process outputs available to account for differences in batch duration as well as missing observations. The validation error for each batch is calculated as:

$$ValidationError = \frac{\sum_{k=c}^{s} |\hat{y}(k) - y_{process}(k)|}{\#ofpredictions} \tag{5.9}$$

where k is the batch index, c is the index when the Luenberger observer converges and s is the batch length.

5.4.3 Case 1: Missing Quality Data

In this case, the amount of missing quality data was varied to retain quality measurements in intervals of 10, 20 and 30 sampling times, resulting in missing data percentages of 90%, 95% and 97%, respectively. The reason for doing this is to test

the utility of the approach for situations where the quality measurements may take longer times to be obtained. Additionally, output data from the training batches are randomly deleted to obtain a missing data percentage of 30%. Both the missing data approach and the traditional subspace approach were then used to develop a subspace model. To handle the missing measurements, the traditional subspace identification approach uses linear interpolation in order to have a full rank matrix. The two models are then validated against validation batches where both output and quality data entries were removed, similar to training batches.

Table 5.2 shows the average validation error of the traditional subspace model and missing data approach from modelling 15 validation batches. The missing data approach consistently has a lower error compared to the traditional subspace model. Figure 5.1 shows the predictions for reactant conversion from the proposed approach, traditional approach, as well as the output estimates using the Luenberger observer for the first 40 time-steps from both models compared to the true process for one batch. figures 5.2 and 5.3 show the predictions for the number average molecular weight and weight average molecular weight from both models compared to the true process. Looking at each of the model predictions, the three figures clearly show how the missing data approach is able to more accurately model the true process in comparison to the traditional subspace identification.

Table 5.2: Average validation error for two models with missing quality data and 30% random output data

Model	*Keep 10 90% Missing*	*Keep 20 95% Missing*	*Keep 30 97% Missing*
Traditional	56,106	73,996	61,190
Proposed	10,149	8,779	9,781
Lags	9	12	14

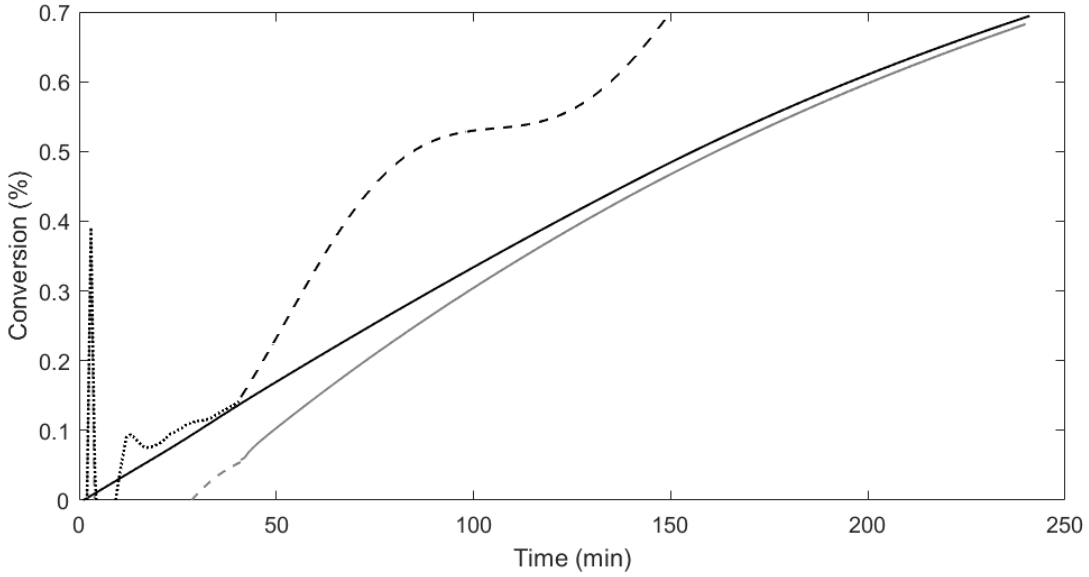


Figure 5.1: The conversion predictions from both models with every tenth quality measurement retained and 30% missing output data. The dashed grey lines represent Luenberger observer estimates until the 40th minute and the solid grey line shows the predictions from the proposed approach in comparison to the process, represented by the solid black line. The dotted and dashed black lines show the predictions made by the Luenberger observer and traditional subspace identification using linear interpolation, respectively.

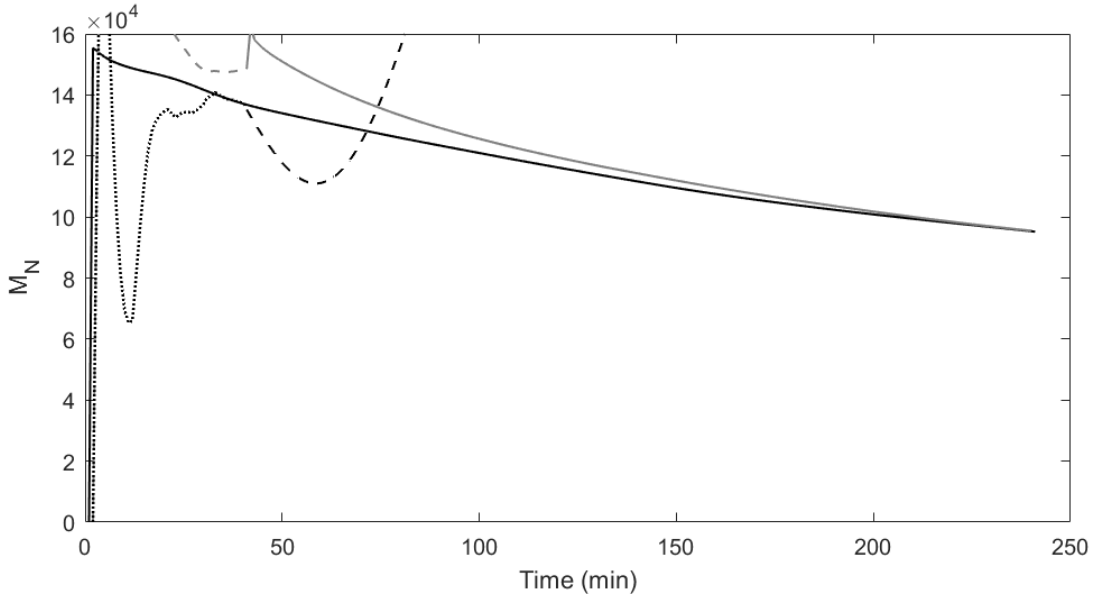


Figure 5.2: The number average molecular weight predictions from both models with every tenth quality measurement retained and 30% missing output data. The dashed grey lines represent estimates made by the Luenberger observer until the 40th minute and the solid grey line shows the predictions from the proposed approach in comparison to the process, represented by the solid black line. The dotted and dashed black lines show the estimates made by the Luenberger observer and traditional subspace identification using linear interpolation, respectively.

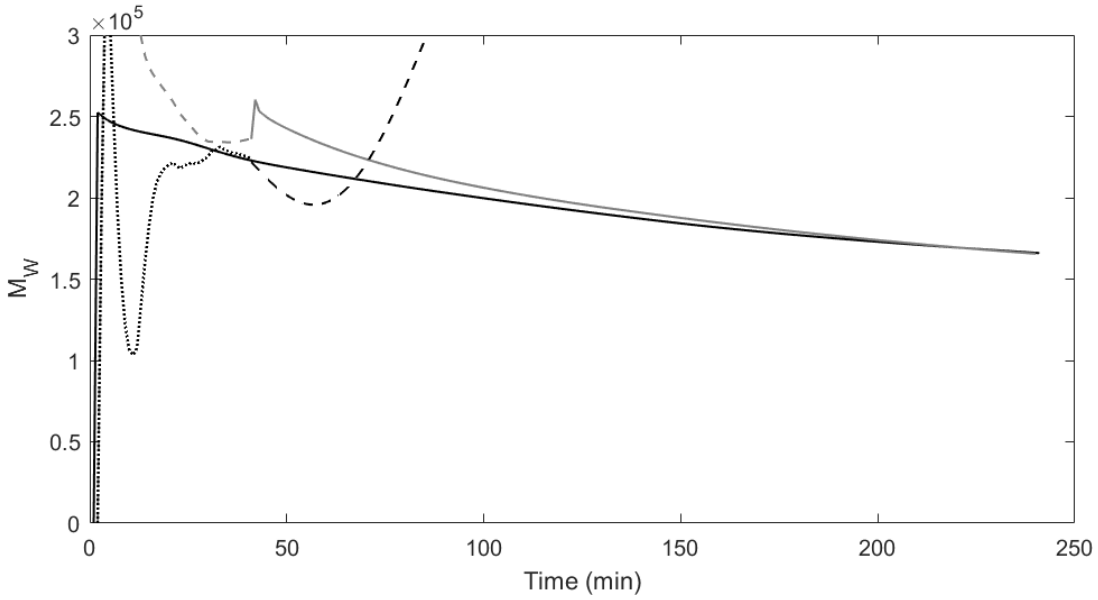


Figure 5.3: The weight average molecular weight predictions from both models with every tenth quality measurement retained and 30% missing output data. The dashed grey lines represent estimates made by the Luenberger observer until the 40th minute and the solid grey line shows the predictions from the proposed approach in comparison to the process, represented by the solid black line. The dotted and dashed black lines show the estimates made by the Luenberger observer and traditional subspace identification using linear interpolation, respectively.

5.4.4 Case 2: Random Missing Output Data

In this case, the output data from training batches is randomly deleted in order to achieve missing data percentages from 5% up to 30% while every tenth quality variable measurement is kept (90% missing outputs). The missing data approach is then compared against traditional subspace identification with linear interpolation. The models are then validated against validation batches with both output and quality data entries removed in a similar manner to training batches.

Table 5.3 shows the average validation error in predicting quality variable data for 15 batches from the traditional subspace identification and missing data approach. In comparison to the traditional subspace approach, the missing data approach is

able to consistently make more accurate predictions. Figure 5.4 shows the conversion predictions made by the proposed approach, traditional subspace approach, as well as the predictions from the Luenberger observer for the first 40 time-steps for both models compared to the true process for one validation batch. figures 5.5 and 5.6 show the predictions for number average molecular weight and weight average molecular weight from both models compared to the true process. The three graphs clearly show that the proposed approach is able to more accurately model and predict the process data compared to traditional subspace identification.

Table 5.3: Average validation error for two models with every tenth quality measurement retained and random missing output data

Model	*Error 5%*	*Error 10%*	*Error 15%*	*Error 20%*	*Error 25%*	*Error 30%*
Traditional	28,649	29,760	34,187	36,041	36,645	56,106
Proposed	12,674	8,016	9,342	10,450	10,242	10,149

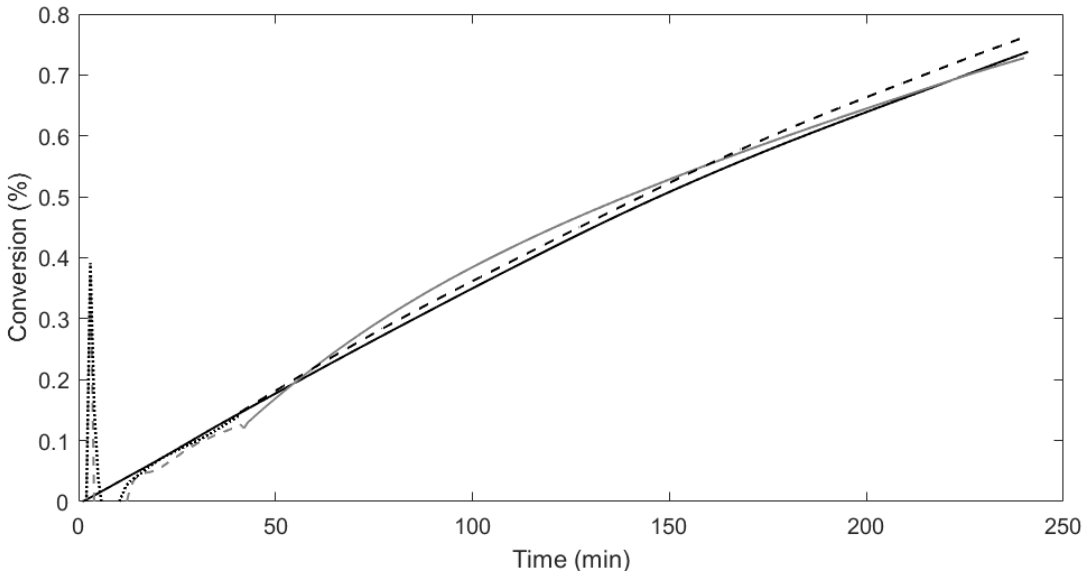


Figure 5.4: The conversion predictions from both models with every tenth quality measurement retained and 20% missing output data. The dashed grey lines represent estimates made by the Luenberger observer until the 40th minute and the solid grey line shows the predictions from the proposed approach in comparison to the process, represented by the solid black line. The dotted and dashed black lines show the estimates made by the Luenberger observer and traditional subspace identification using linear interpolation, respectively.

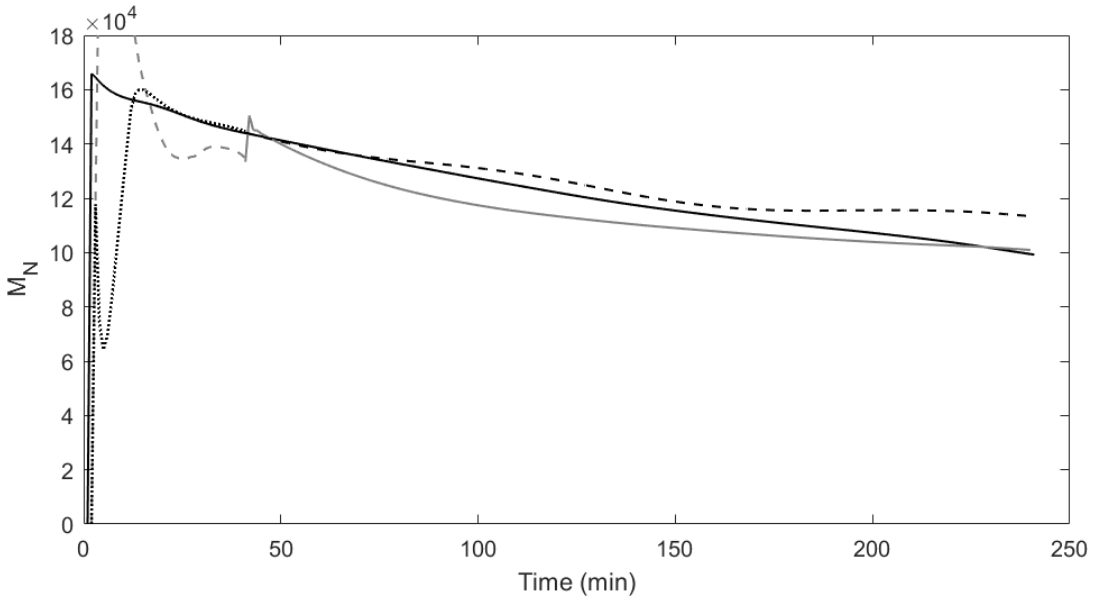


Figure 5.5: The number average molecular weight predictions from both models with every tenth quality measurement retained and 20% missing output data. The dashed grey lines represent estimates made by the Luenberger observer until the 40th minute and the solid grey line shows the predictions from the proposed approach in comparison to the process, represented by the solid black line. The dotted and dashed black lines show the estimates made by the Luenberger observer and traditional subspace identification using linear interpolation, respectively.

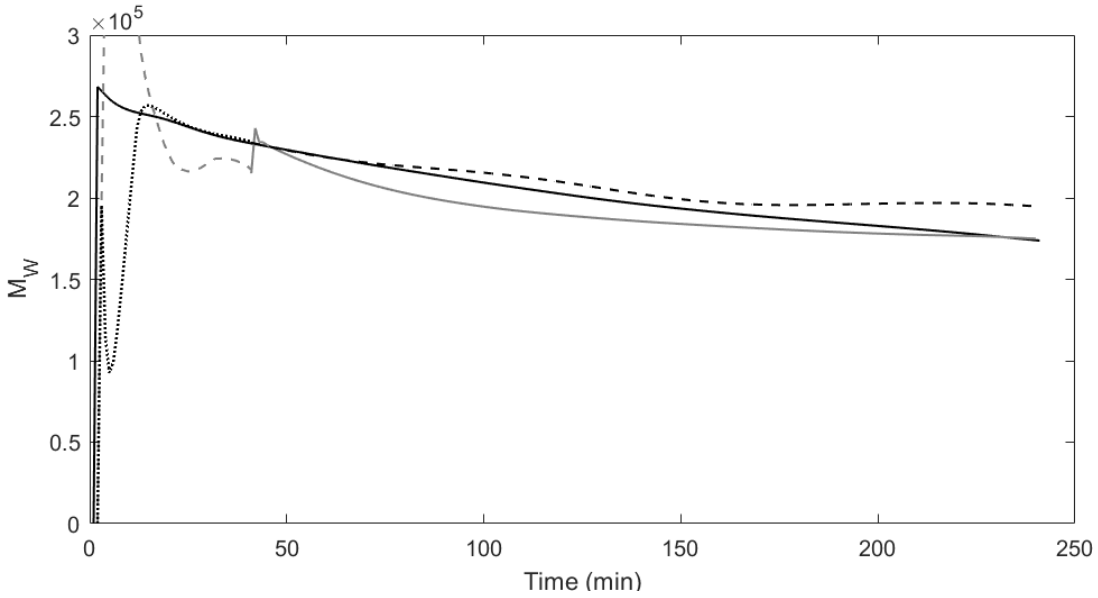


Figure 5.6: The weight average molecular weight predictions from both models with every tenth quality measurement retained and 20% missing output data. The dashed grey lines represent estimates made by the Luenberger observer until the 40th minute and the solid grey line shows the predictions from the proposed approach in comparison to the process, represented by the solid black line. The dotted and dashed black lines show the estimates made by the Luenberger observer and traditional subspace identification using linear interpolation, respectively.

Remark 23. *Note that while the subspace identification approach itself is not a novel introduction, its application to the batch quality problem is. Quality variables are different from missing data scenarios since they are available at a much lower frequency than traditional outputs and they contain more important process information. Modeling the intermittent quality measurements along with the process outputs is a difficult task, given the large differences in data frequency. However, a singular model is desirable for control. This paper demonstrates how missing data algorithms can be adjusted to include quality variables for accurate process modeling.*

5.5 Conclusions

In this work, a missing data subspace identification approach for batch processes with missing quality measurements and outputs is proposed. In this approach, missing quality measurements were treated as missing data, allowing the model to make predictions without inaccurately imputing the missing observations. The missing data subspace model is compared to traditional subspace identification using linear interpolation. The proposed model was able to perform better than the traditional approach for cases with quality measurements retained in intervals of 10, 20 and 30 as well as cases with random missing output data ranging from 5% to 30%. The results from the missing data approach show the benefits of using latent variable methods to handle missing data.

5.6 Acknowledgments

Financial support from the McMaster Advanced Control Consortium is gratefully acknowledged.

Bibliography

[1] Corbett, B., Macdonald, B., and Mhaskar, P. (2013). Model Predictive Quality Control of Polymethyl Methacrylate. *IEEE Transactions on Control Systems Technology*, 23(2):3948–3953.

[2] Corbett, B. and Mhaskar, P. (2016). Subspace identification for data-driven modeling and quality control of batch processes. *AIChE Journal*, 62(5):1581–1601.

[3] Ding, J. and Lin, J. (2014). Modified subspace identification for periodically non-uniformly sampled systems by using the lifting technique. *Circuits, Systems, and Signal Processing*, 33(5):1439–1449.

[4] Ekpo, E. and Mujtaba, I. M. (2008). Evaluation of neural networks-based controllers in batch polymerisation of methyl methacrylate. *Neurocomputing*, 71(7-9):1401–1412.

[5] Fan, S., Gretton-Watson, S., Steinke, J., and Alpay, E. (2003). Polymerisation of methyl methacrylate in a pilot-scale tubular reactor: modelling and experimental studies. *Chemical engineering science*, 58(12):2479–2490.

[6] Flores-Cerrillo, J. and MacGregor, J. F. (2004). Control of batch product quality by trajectory manipulation using latent variable models. *Journal of Process Control*, 14(5):539–553.

[7] Flores-Cerrillo, J. and MacGregor, J. F. (2005). Latent variable mpc for trajectory tracking in batch processes. *Journal of process control*, 15(6):651–663.

[8] Hu, B., Zhao, Z., and Liang, J. (2012). Multi-loop nonlinear internal model controller design under nonlinear dynamic pls framework using arx-neural network model. *Journal of Process Control*, 22(1):207–217.

[9] Huang, B., Ding, S. X., and Qin, S. J. (2005). Closed-loop subspace identification: an orthogonal projection approach. *Journal of process control*, 15(1):53–66.

[10] Jiang, Q., Yan, X., and Huang, B. (2015). Performance-driven distributed pca process monitoring based on fault-relevant variable selection and bayesian inference. *IEEE Transactions on Industrial Electronics*, 63(1):377–386.

[11] Larimore, W. E. (1996). Statistical optimality and canonical variate analysis system identification. *Signal Processing*, 52(2):131 – 144. Subspace Methods, Part II: System Identification.

[12] Ljung, L. (2002). Prediction error estimation methods. *Circuits, Systems and Signal Processing*, 21(1):11–21.

[13] MacGregor, J. F., Jaeckle, C., Kiparissides, C., and Koutoudi, M. (1994). Process monitoring and diagnosis by multiblock pls methods. *AIChE Journal*, 40(5):826–838.

[14] Markovsky, I. (2013). Exact system identification with missing data. In *52nd IEEE Conference on Decision and Control*, pages 151–155. IEEE.

[15] Moonen, M., De Moor, B., Vandenberghe, L., and Vandewalle, J. (1989). On-and off-line identification of linear state-space models. *International Journal of Control*, 49(1):219–232.

[16] Patel, N., Mhaskar, P., and Corbett, B. (2020a). Subspace based model identification for missing data. *AIChE Journal*, 66(10):e16538.

[17] Patel, N., Nease, J., Aumi, S., Ewaschuk, C., Luo, J., and Mhaskar, P. (2020b). Integrating data-driven modeling with first-principles knowledge. *Industrial & Engineering Chemistry Research*, 59(11):5103–5113.

[18] Qin, S. J. (2006). An overview of subspace identification. *Computers and Chemical Engineering*, 30(10-12):1502–1513.

[19] Rho, H.-J., Huh, Y.-J., and Rhee, H.-K. (1998). Application of adaptive model-predictive control to a batch mma polymerization reactor. *Chemical Engineering Science*, 53(21):3729–3739.

[20] Shang, L., Liu, J., Turksoy, K., Shao, Q. M., and Cinar, A. (2015). Stable recursive canonical variate state space modeling for time-varying processes. *Control Engineering Practice*, 36:113–119.

[21] Söderström, T., Stoica, P., and Friedlander, B. (1991). An indirect prediction error method for system identification. *Automatica*, 27(1):183–188.

[22] Van Overschee, P. and De Moor, B. (1994). N4sid: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30(1):75–93.

[23] Van Overschee, P. and De Moor, B. (1995). A unifying theorem for three subspace system identification algorithms. *Automatica*, 31(12):1853–1864.

[24] Verhaegen, M. and Dewilde, P. (1992). Subspace model identification part 2. analysis of the elementary output-error state-space model identification algorithm. *International journal of control*, 56(5):1211–1241.

[25] Wang, J. and Qin, S. J. (2002). A new subspace identification approach based on principal component analysis. *Journal of process control*, 12(8):841–855.

[26] Zhao, Y., Huang, B., Su, H., and Chu, J. (2012). Prediction error method for identification of lpv models. *Journal of process control*, 22(1):180–193.

Chapter 6

Subspace Based Model Identification for an Industrial Bioreactor: Handling Infrequent Sampling Using Missing Data Algorithms

This chapter provides a practical application of the missing data algorithm along with the additional benefit of handling discrete measurements to solve the infrequent sampling problem. The key problems were the large amounts of missing data due to the different sampling rates and a discrete input signal. This work leverages the missing data algorithm described in the previous chapter to use a higher frequency sampling rate that allows the discrete inputs to be treated as step inputs over one sampling period. This work was completed in collaboration with Sartorius who provided insight into the industrial bioreactor and provided data and modeling experience.

Patel, N., Corbett, B., Trygg, J., McCready, C., & Mhaskar, P. (2020). Subspace Based Model Identification for an Industrial Bioreactor: Handling Infrequent Sampling Using Missing Data Algorithms. *Processes*, 8(12), 1686.

6.1 Abstract

This manuscript addresses the problem of modeling an industrial bioreactor using process data. Bioreactor design is an integral part of modern industrial practice and is responsible for the production of high value products, and in particular, because of confidentiality issues, for what we will refer to as the Sartorius Bioreactor. Due to the complex mechanisms driving cell growth, determining the optimal operating conditions for the Sartorius Bioreactor is a challenging but important problem. One of the key impediments to this is the difficulty in developing good dynamic process models. While increased availability of sensors has made more data available, the appropriate use of this data for developing a data-driven dynamic model remains challenging. In particular, in the context of the Sartorius Bioreactor, it is important to appropriately address the problem of dealing with a large number of variables, which are not always measured or are measured at different sampling rates, without taking recourse to simpler interpolation or imputation based approaches. These approaches might be suitable for steady state modeling, but do not necessarily serve the objective of data driven dynamic modeling. This manuscripts addresses the problem of developing a dynamic model for the Sartorius Bioreactor via appropriately adapting a recently developed subspace model identification technique which in turn uses nonlinear iterative partial least squares (NIPALS) algorithms to gracefully handle the missing data. The other key contribution is evaluating the ability of the identification approach to provide insight into the process by computing interpretable variables such as metabolite rates.

6.2 Introduction

Bioreactors are an important part of many different industries ranging from environmental engineering to bio-pharmaceuticals. Several factors influence the productivity of a bioreactor including mass transfer, heat transfer and concentration of the biocatalyst used to produce the final product. [26] One such bio-pharmaceutical product is monoclonal antibodies, which is the focus of the present work. In this process, one of the key objectives is to maximize the volume specific production of the antibody. While the desired product can be characterized in many ways, the Sartorius Bioreactor is designed to produce the specific protein through a careful manipulation of bioreactor properties. The volumetric production of the monoclonal antibodies is influenced by several parameters such as feed concentrations and cell growth that must be appropriately modeled and controlled.

There are a host of parameters that determine the cell growth, death, and protein production dynamics. Glucose is a key nutrient for cell growth providing the necessary source of energy and biomass formation however, excess glucose can lead to production of lactate leading to increased cell death. Glutamine behaves similar to glucose acting as a nutrient promoting cell growth especially in cases of fast cell growth. Both lactate and ammonia have detrimental effects on cell growth with the latter being more potent.[12] External factors like temperature and pH play an important role in maximizing the effects of the various nutrients on cell growth. Increasing the temperature up to a certain threshold increases cell growth due to increased system dynamics. After which a temperature shift midway through the batch is used to increase antibody production. [2, 29] The effects of pH are more complicated since the effects of lactate and ammonia are dependent on the pH levels therefore a shift in the pH is also often necessary in later stages of the batch.[12] The complex effects of these metabolites and external factors in batch product leads to a challenging

modeling and control problem.

While detailed first principles equations for bioreactors exist in general, and for the Satorius Bioreactor in particular [12], the associated parameter estimation problem is quite a challenging one. There have been several implementations of parameter estimation techniques and other mathematical approaches for first principles modeling of industrial-scale growth, [23, 6, 19, 5, 1, 16]including one used by Sartorius [12], however, further contributions to these methods remains the subject of another work. The focus in the present manuscript is on leveraging data to build data driven dynamic models. There are several challenges with the measurements available from the bioreactor. While some of the process variables can be measured continuously using online sensors (ie. temperature, pH) other variables (ie. metabolites) require sampling and separate tests. This results in an infrequent sampling problem where only some observations are available frequently while the rest are not, leading to instances of 'missing data'. In order to account for the missing observations, existing modeling approaches must either interpolate the missing values or use a method to align the available measurements. Interpolating values is not a reliable approach when dealing with highly nonlinear dynamics. Additionally, using only the available measurements to build a model ignores the continuous measurements available between the sampling intervals. Note that each of th metabolite concentrations must be measured independently requiring several samples to be taken in order to get a full range of measurements. This isn't practical in a bioreactor therefore measurements are only taken a couple of times a day leading to large amounts of missing data. Additional factors that must be accounted for are that due to the negative effects of excess glucose, the Sartorius Bioreactor operation involves discrete additions after levels drop below a certain threshold. This discrete addition leads to a discontinuous glucose profile which must be appropriately accounted for. Furthermore, since cell growth relies on a host of parameters and peaks during batch operation, the length of each batch is a design choice and can be variable, and the data driven and modeling approach

must be able to handle batches of varying lengths.

Due to the reasons stated above, many of the existing approaches for data-driven modeling are not directly suitable to solve the current identification problem. When attempting to analyze industrial batch data containing missing observations, a common data-driven technique used is partial least squares (PLS) which works on the principle of projection to latent space. [9] In this technique, process data from multiple batches is taken and projected into a lower dimensional subspace (latent variable space). This ensures that the relationships between the correlated input and output space variables is maintained and is characterized by the independent latent variables.[MacGregor et al.] PLS techniques are capable of handling missing data since they utilize the covariance structure between the input and output variables from the original variable space. This inherent ability is one of the key properties that make PLS techniques suitable for modeling batch processes. The application of PLS techniques to batch process modeling has been previously explored and one successful approach is to utilize batchwise unfolding of the data. [7, MacGregor et al.] Process data from each batch is unfolded into a single PLS observation that is subsequently related to quality variables. This approach can be applied to on-line process data by utilizing data imputation techniques to make predictions on the missing data observations. While this approach has been well-documented in handling industrial batch data, it is not readily suitable for the current problem since it requires batch alignment in order to account for varying batch duration (although techniques such as dynamic time warping or using alignment variables exist). More importantly this approach inherently does not distinguish between inputs and outputs- thus all variables are treated in the same fashion, in turn requiring special modifications to recognize the distinctions between process inputs and outputs.

Another technique that is suitable for building dynamic process models is subspace identification [18, 21, 10, 25] which has been adapted for handling batch data. [3].

Note that subspace identification is different from PLS because it explicitly distinguishes between input and output variables. [18, 21, 10, 25] Subspace identification consists of two distinct steps: identifying a state trajectory from historical input and output batch data and using a least squares solution to determining the system matrices of a Linear Time Invariant (LTI) system. To achieve these outcomes subspace identification utilizes a range of techniques from canonical variate analysis [14, 22], numerical algorithms [24] and multivariate output error state space algorithms. [27] One common technique to subspace identification is singular value decomposition (SVD) of the matrices. [18, 10] However, SVD requires matrices to be full-rank making it unsuitable for handling batch data with missing observations.[17] Thus, subspace identification by itself is unable to handle the metabolite rates with missing measurements coming from the infrequent sampling rates.

As a result of these considerations, a missing data subspace modeling approach using PCA and PLS steps was recently developed [20]. Specifically, the addition of PCA and PLS steps to the subspace identification approach allows for the missing observations to be accounted for. While the use of PCA and PLS techniques is not a novel introduction to model identification (see [28, 11]) the reduced latent variable space is marginally affected by missing data. The first step in the approach is to use latent variable methods (PCA followed by PLS) to identify a reduced dimensional space for the variables which accounts for missing observations. The second step replaces SVD with PCA, to handle missing observations, to identify the states of the system whereupon traditional subspace approaches can be utilized. The approach in [20], however, does not directly handle discrete additions (of glucose), and thus is not directly applicable. Another recent result [4] that explicitly handles discrete additions is not applicable due to two reasons- the first is that the results in [4] do not handle missing data, and the second is that a direction application of the approach in [4] coupled with the missing data approach of [20] would lead to batches with almost no data, in turn making the approach inapplicable. Finally, while the results in [4, 20] provide a

modeling framework, the resultant subspace models do not necessarily provide insight into the process dynamics, and could be improved by augmenting with tools to enable easier access to the practitioner.

Motivated by the above considerations, the present manuscript adapts the missing data approach of [20] to specifically handle the discrete addition nature of the Sartorius Bioreactor along with the missing data in the metabolite measurements and develops a data driven dynamic model that also predicts variables that can be much better interpreted by the practitioner. The approach is designed to handle batch data with variable batch length without the need for batch alignment techniques. This approach is utilized to identify two LTI models of the system: one for the concentrations and the other, to provide more insight into the process dynamics, for the metabolite rates. The rest of the paper is organized as follows: Section 6.3 presents the bioreactor process and overview of traditional subspace identification. In Section 6.4, an application of the proposed approach to the Sartorius Bioreactor is presented. Finally, concluding remarks are made in Section 6.6.

6.3 Preliminaries

A brief overview of the bioreactor process is presented in this section followed by the missing data subspace identification approach for batch processes.

6.3.1 Bioreactor Process Description

The Sartorius bioreactor is operated as a fed-batch reactor with nominal or centre point conditions of a pH of 7.1, dissolved oxygen of 60% and a temperature of $36.8°C$. It has a discrete feed input utilized to maintain the glucose concentration in the reactor at 2.5 g/L. After being initialized with a starting cell culture the process runs

for 12 days before the reactor is stopped and the final cell titer is measured. The process has some continuous measurements available such as pH and temperature, but the rest of the measurements are only sampled up to three times a day. The bioreactor has the ability to control temperature, pH and (through discrete additions), the glucose concentrations. The measured outputs are titer, viable cell density(VCD), cell viability, glutamine concentration, lactate concentration, glutamate concentration and ammonia concentration.

Sartorius Bioreactor utilizes a discrete nutrient feed system. Thus, glucose is added to the system in a series of discrete additions in order to maintain the target glucose concentration. The glucose measurement is utilized to determine the glucose addition time, and at each addition interval a feed volume (200mL) with a high glucose concentration is added to the bioreactor resulting in a sharp (slight) increase in the volume and a larger increase in the glucose concentration. The eventual objective of the model is to be utilized for the purpose of a control strategy such as model predictive control. The utility of the model therefore is in its ability to predict the final protein titer, by using the measured inputs and outputs for up to a given time in the batch, and based on candidate input variables at the end of the batch. Another key objective is to utilize the model to monitor rates of metabolites consumption. These rates provide a useful view'into the process and enables making more sense of the model, in turn making the model much more accessible to the practitioner.

6.4 Dynamic Modeling of the Bioreactor

In this section, first a dynamic model is identified, with the model output being measured variables. In the next subsection, a dynamic model is identified that uses a combination of measured and calculated variables to directly estimate metabolite consumption rates.

6.4.1 Dynamic Model Identification and Validation Using Measured Outputs

The first model examines the daily metabolite concentrations and their impact on cell titer. In this process the following measurements are available: glucose concentration, temperature setpoint, pH setpoint, titer, viable cell density(VCD), cell viability, glutamine concentration, lactate concentration, glutamate concentration and ammonia concentration. One of the first decisions in developing subspace identification based models is determining the input and output variables that allows for model identification, and is also in line with process implementation. Thus, pH setpoint and the temperature setpoint are selected as two of the input variables. The controller on the process works reasonably well, thus the pH and temperature values pretty closely follow the setpoints. The objective in this work is to determine the effect of these variables on the metabolites and cell titer, not the effect of the pH and temperature setpoint on the pH and temperature. In essence, the temperature and pH directly influence cell growth dynamics and the shifts in the setpoints represents changes in the growth profiles. The other measured variables inside the bioreactor, however, do not cause significant changes in the temperature or pH values and so the measured output values have more noise than useful information and are consequently omitted. Thus only the titer, viable cell density (VCD), cell viability, glutamine concentration, lactate concentration, glutamate concentration and ammonia concentration are chosen as the seven outputs.

Glucose on the other hand poses its own challenge. There are two potential ways to include glucose in the model. The first is to include the glucose addition as an input, and model glucose as an output. In such a scenario the model would be trying to decipher the effect of glucose addition on the glucose concentration- which is a fairly straightforward mole balance, and the other is the effect of the rate of consumption

of glucose in its role as a metabolite. While this is possible in principle, every discrete addition of glucose would cause a jump in the glucose measurement, and would in turn cause the states to 'jump'. Such a discrete addition piece could be modeled using the subspace identification approach in [3], but would lead to having to split the batch into multiple batches- with each batch comprising the time period between discrete additions. While this would be possible in principle (and reasonable for the process considered in [3]), in the present instance this would lead to each of the batches having very sparse measurements- thereby comprising mostly missing data. In this case, the recently developed missing data approach [20] would not be directly applicable.

Glucose is therefore considered an input in the present manuscript. From a practical standpoint, it is reasonable because the glucose concentration can be readily measured and modified and thus be an input in a controller implementation. The dataset however poses an interesting challenge in this regard because the measurements of glucose are takn before the glucose addition, but not measured right after the glucose addition. The first measure to handle that includes the computation of the glucose concentration right after the glucose addition. This is the more intuitive part, and can be computed readily as follows: $V^{+}C_G^{+} = V^{-}C_G^{-} + V_{G_{Feed}}C_G^{Feed}$, where V is the volume in L, C_G is the concentration of glucose in mg/L and the + and - represent the after and before feed addition respectively

The other more important question is how to utilize the newly computed glucose concentration. Again, there are two alternatives and here one approach is clearly incorrect. The first alternative is to add an additional data point in the batch. Thus right after the data point before the addition, a new point is added where the value of the glucose measurement is changed, but the value of the other variables is kept the same. While this sounds intuitively right, such a choice would provide the model with false information. In particular, it would suggest to the model that the value of the glucose changed in one sampling time while the others stayed the same. This

is counter to what happens in the process in that the value of the glucose jumps instantaneously. The implementation of this approach is shown in Figure 6.1 and it clearly shows how the concentration in the reactor 'increases' between sampling intervals. For example after days 3,6, 7 8,9,11 the glucose concentration seems to increase slowly over time which is contrary to what we know happens (i.e., glucose gets consumed). The second and correct adaptation then, is to replace the value of the glucose measurement by the newly calculated measurement. As seen In Figure 6.2 the concentration increases instantly upon glucose addition and the next measured sample shows that the glucose concentration decreases between sampling instances.

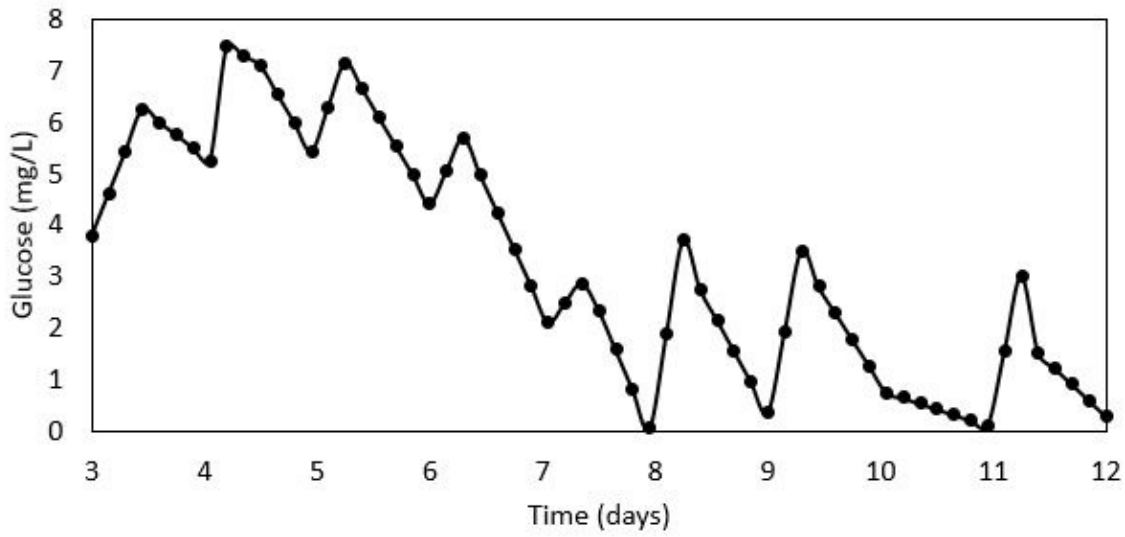


Figure 6.1: The glucose input profiles for a training batch using the incorrect assumption of taking measurements whenever they are sampled.

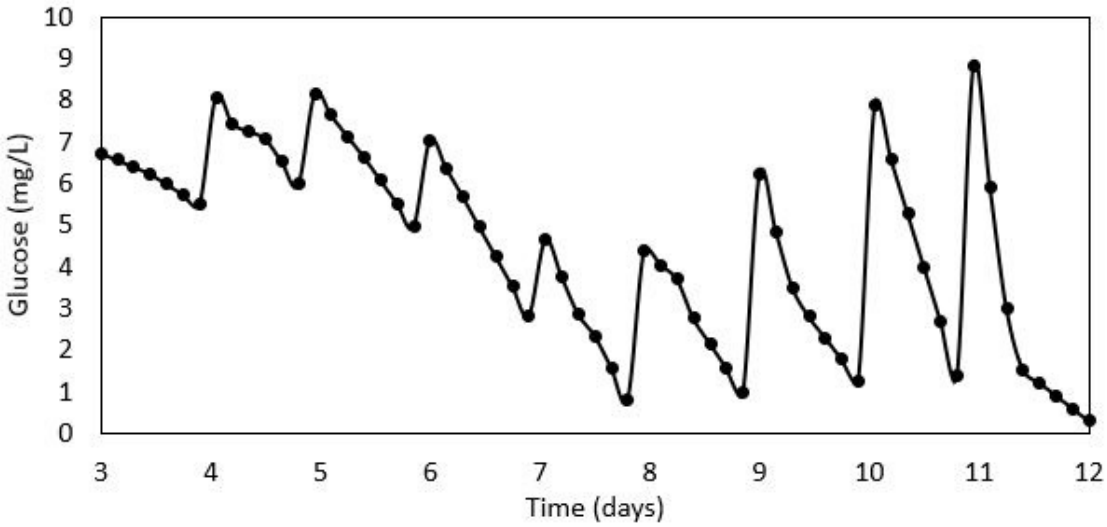


Figure 6.2: The glucose input profiles for a training batch using the correct approach of updating the glucose concentration instantaneously.

Having determined the right set of inputs and outputs, the training input sequence from one batch is shown below in Figure 6.3. Note that the temperature and pH setpoints are only moved from the center line values to induce variations in the dataset, reflective of the true process, and as such not all batches have these shifts. To identify the model, data from 11 different batch runs were used for training batches. The training batches were chosen to establish the daily operating conditions of the Sartorius bioreactor with sufficient variation provided by different temperature and pH setpoint changes providing a reasonably rich data set.

Remark 24. *We recognize that the use of 11 training batches does limit the ability to accurately validate the model. In future work, as more data becomes available, the identification can be redone to include more batches. What is perhaps more important to recognize is that the model is good for the data range it is used for in the training. Thus, in conjunction with existing model monitoring techniques [13], one can readily monitor if the model continues to be valid for the batch under consideration. If the monitoring technique reveals*

that the model predictions are diverging from the observations, the model can be retrained using the new on-line measurements in order to improve model accuracy.

Having handled the discrete nature of the input addition, the data driven modeling approach [20] was subsequently implemented to identify a system model. A state space model of order 3 was identified by ensuring the best model fit during the training stages.

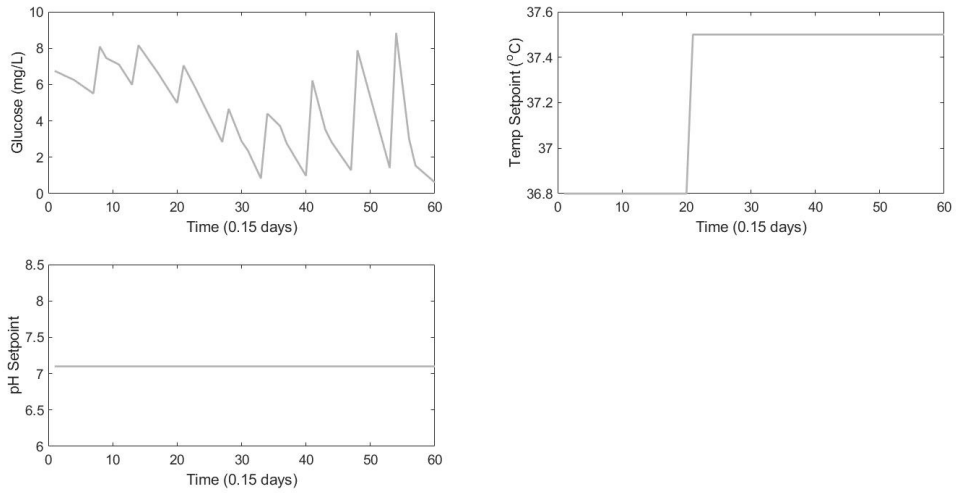


Figure 6.3: The input profiles for a training batch.

The modeling identification procedure [20] used is a combination of subspace identification with PCA and PLS techniques to handle variable batch length missing data problems. The identification approach used in this paper identifies an LTI model as follows: Given s measurements (where s represents the length of the data) of the input $u^{(b)}[k] \in \mathbb{R}^m$ and the output $y^{(b)}[k] \in \mathbb{R}^l$ variables from each batch a model with order n can be identified using the following equations:

$$\begin{aligned} \hat{\mathbf{x}}^{(b)}[k+1] &= \mathbf{A}\mathbf{x}^{(b)}[k] + \mathbf{B}\mathbf{u}^{(b)}[k], \\ \mathbf{y}^{(b)}[k] &= \mathbf{C}\hat{\mathbf{x}}^{(b)}[k] + \mathbf{D}\mathbf{u}^{(b)}[k], \end{aligned} \tag{6.1}$$

where the objective is to determine the order n, from cross validation, and the system matrices $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{B} \in \mathbb{R}^{n \times m}$, $\mathbf{C} \in \mathbb{R}^{l \times n}$, $\mathbf{D} \in \mathbb{R}^{l \times m}$.

The system matrices are identified in two stages where first a state sequence is identified and then subsequently the system matrices. The subspace identification approach is carried out using a series of PCA and PLS regressions with non-iterative partial least squares algorithms (NIPALS). The first part of subspace identification is to identify a state trajectory. This is done using PCA by projecting the past inputs and outputs perpendicular to the future inputs. We recognize that the future inputs should be completely independent of any past data however, this step insures we remove any potential correlations as a result of insufficient excitation. Additionally, the future outputs are projected perpendicular to the future inputs. Recognizing that the future outputs are a result of the states and future inputs by removing the correlation between the future inputs the remaining correlation depends on the states. In the next step, PLS is carried out between the newly deflated past inputs and outputs and future outputs. This is done in order to explain how the past data results in the current states for the future data and to expose the underlying state relationship. Finally in order to explicitly identify the state trajectory, where traditional methods [18] utilize singular value decomposition, this approach uses PCA. The end result is a state trajectory that can be used to identify the system matrices using regression techniques. The key use of NIPALS algorithms in these steps gives this approach the ability to handle missing data. For a more detailed explanation of the approach used in this paper see [20].

Remark 25. *Sartorius has developed a good first principles model of the bioreactor however, the parameter estimation problem continues to be a focus of future work. That said, the proposed data driven approach could readily be utilized with the first principles model. Thus, a data driven approach which leverages the process data can be utilized to develop a hybrid model for improved*

prediction power. [8]

Remark 26. *One of the considerations when modeling a dynamic process like cell growth is that there are different phases in cell growth that occur over the course of one batch. The metabolic response of cells to their environment is complex and therefore strongly nonlinear. This response is also widely believed by biologists to be non-markov (ie cells have memory of historical conditions). In these different phases the process may behave differently making a linear time invariant model an unsuitable choice. To handle this situation it is possible to treat each different growth phase as a separate smaller batch. This differs from the traditional batch problem since the beginning of each smaller batch represents the end of the previous smaller batch. These smaller batches would then be used as part of the model identification allowing the identified subspace model to appropriately capture the behavior in each phase. As can be seen in the application section, the present data driven modeling approach works reasonably well. In future work, as more batches need to be modeled (and the model will likely be utilized for feedback control purposes), such phase-based identification approaches will be pursued. Finally, another direction of generalization would be to determine good initial conditions for the states based on the measured observations. Presently, the states are initialized at a value which is the average of the value for the batches used in training. In future work, an approach can be followed where the subspace model is better initialized for quick convergence of the state observer and the resultant ability to predict starting from early on in the batch.*

Remark 27. *Note that one of the advantages of a first principles modeling approach is that it can be more easily extrapolated. Thus if a first principles modeling approach is used, the resultant rate expressions can be utilized to model the operation of the process in a continuous fashion. On the other*

hand, a data driven model identified using data from batch operation cannot be directly applied to continuous operation. It can serve as a 'starting point' and adapted using the monitoring based re-identification approach, and, even more quickly retrained if it is utilized as part of a hybrid modeling strategy via leveraging the extrapolation capability of the underlying first principles model.

6.4.2 Dynamic Model Validation

This section illustrates the validation procedure for a new batch. Recall that validation is the key step in model identification, by providing a means to evaluate the successes of a developed model. Note that one of the inherent features of any state space based model is the requirement of the knowledge of initial states. If it is a first principles model, where not all the state variables are measured (as is often the case) using the first principles model would require an initial state estimation process step. In the present instance, the model is a linear state space mode, with the states being a realization of the input output dynamics, and thus by construction, unmeasured. By the same construction though, the states are observable from the measured outputs, and thus enable the design of a state observer/estimator. For a new data set therefore, an initial state estimate is first computed before prediction is possible. In the present work, a Luenberger observer design is used at the beginning of the batch until the predicted outputs converge with the process outputs. The observer has the following form:

$$\hat{\mathbf{x}}[k+1] = \mathbf{A}\hat{\mathbf{x}}[k] + \mathbf{B}\mathbf{u}[k] + \mathbf{L}(\mathbf{y}[k] - \hat{\mathbf{y}}[k]) \tag{6.2}$$

where $\mathbf{L}$ is the observer gain and is chosen to ensure that $(\mathbf{A} - \mathbf{LC})$ is stable.

The missing data problem has specific implication in this regard, and need to be adequately accounted for. Thus, the above observer cannot be 'implemented' directly when parts of the output are missing. Specifically, when the output measurement is missing the term used to update the prediction, $\mathbf{L}(\mathbf{y}[k] - \hat{\mathbf{y}}[k])$, yields an undefined value. In order to operate the state observer with missing data this work uses the linearly interpolated value as the process measurement at time k in order to update the states.

Remark 28. *The use of linear interpolation for state estimation is only one of the possible approaches. In addition to multirate state estimation, which is a well documented problem, it possible to build a smaller model without missing data for state estimation. This approach involves building a separate subspace model using the continuous output observations. This model can then be used to estimate the states of the system until they converge and the full model can be used for validation. This approach is not considered in the present manuscript, primarily because of the observed success of the modeling approach, but with increased data availability and modeling challenges, could very well be included in future work.*

After the states have converged this is where the identified model's predictive capabilities are tested with the missing outputs. The remainder of the batch is predicted using the model however, as the process measurement comes in the model uses that estimate with the observer in order to update the state estimate at that specific sampling time.

Remark 29. *While the present illustration utilizes linear interpolation for the state estimator it does not assume any knowledge of the process instead taking measurements as they become available. Linear interpolation is only used to allow for a good state estimate to be obtained which is not a part*

of validating the identified model's predictive capabilities. The model is still identified from a dataset with missing values and can be used to predict when process measurements are not available. Note that the model's predictive capability is not limited to a 'next step prediction' the model predicts to the end of the batch and updates the trajectory with each available measurement to predict the final quality more accurately.

To show the effectiveness of the missing data approach on the Sartorius bioreactor case study, this section identifies a dynamic model used to predict the quality variables and a dynamic model to identify the metabolite rates. These models are built on training data from the Sartorius bioreactor and then validated on a separate batch. The error is calculated as the normalized prediction error between the predicted model and the true process outputs. Note that the error is only calculated at the points where process measurements are available. The error is calculated as follows:

$$PredictionError = \sum \frac{|\hat{y} - y_{process}|}{predictions} \tag{6.3}$$

where $y_{process}$ represents the process outputs, $\hat{y}$ represents the predicted outputs and predictions represent the number of available measurements. The prediction errors from both the training data and the validation batch are shown below in Table 6.1. As expected the validation error is slightly larger than the training error since the validation batch was not used in model identification. Figure 6.4 shows the training results and Figure 6.5 shows the validation results from the quality model. In Figure 6.4 there are model predictions despite the lack of a process measurement because the model keeps track of the states internally allowing it to make predictions at every time step. As shown in both sets of figures the model is able to accurately predict the trends in the metabolites and more importantly the viable cell density which shows

the cell concentration at the end of the batch. This is the key parameter Sartorius uses in downstream processes and despite the large amounts of missing data the trend was accurately predicted.

Table 6.1: The prediction error between the subspace based model and the process for both the training and validation batches.

Model	*Fit Error*
Training	0.7930
Validation	1.9696

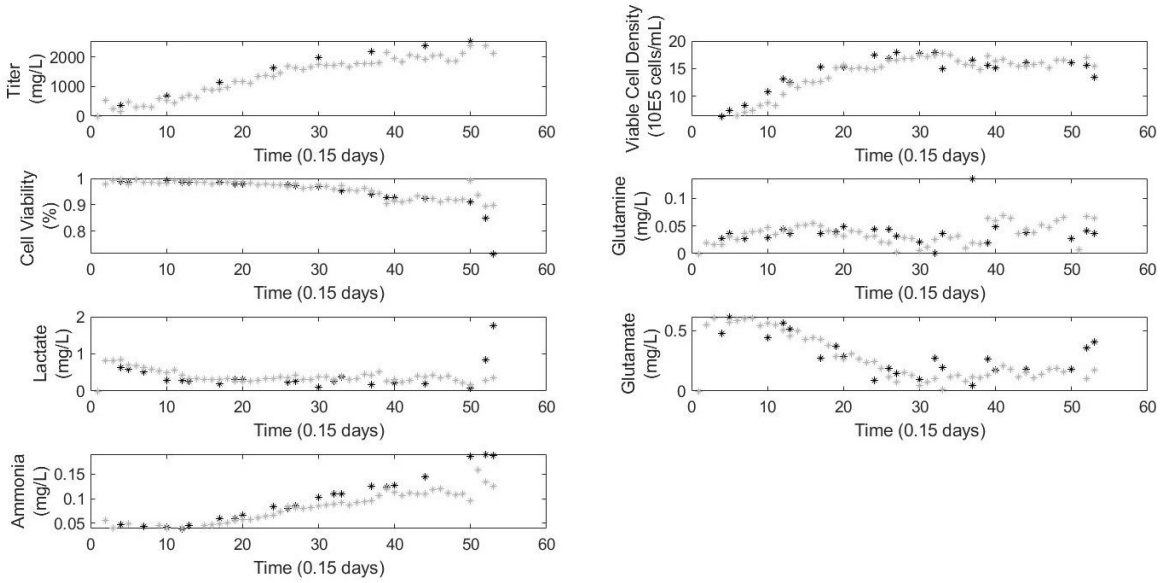


Figure 6.4: The training fit (grey) from the dynamic model for each output are compared against the process data (black) for a training batch.

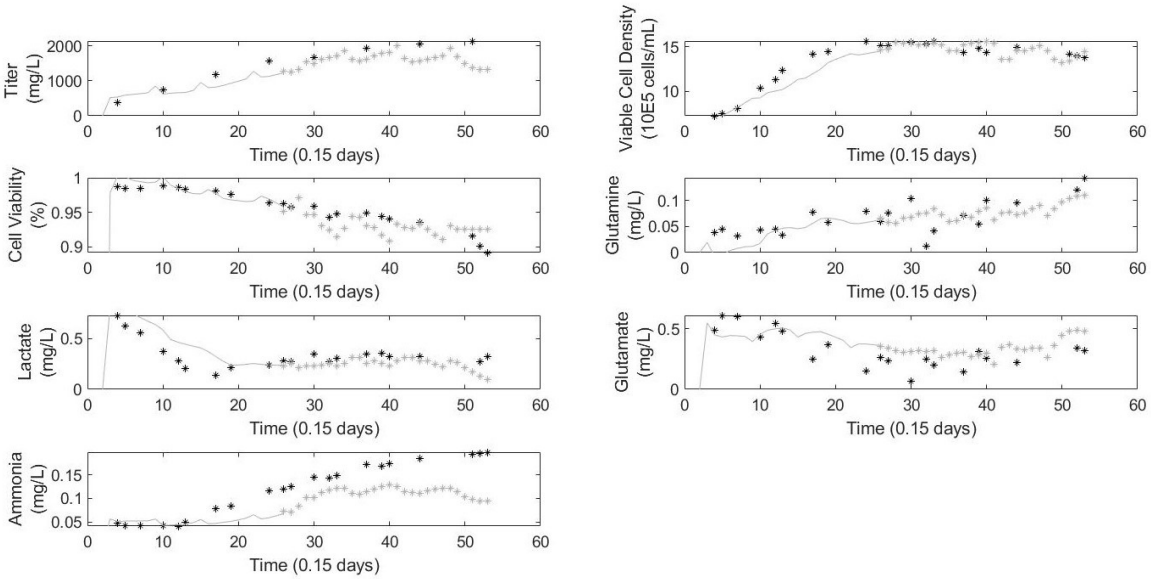


Figure 6.5: The process data (black) is compared with the dynamic model predictions using the state observer (grey solid) until the states converge and then the dynamic model predicts the remainder of the validation batch (grey starred).

6.5 Metabolite Rate Modeling of the Bioreactor

The metabolite rate model is important for Sartorius in order to see the daily trends in the bioreactor. The goal is to be able to control the reactor overnight based on the end of day predictions. Thus, knowing the trends in the metabolite rates is an important factor when considering what additions need to be before allowing the process to run. In addition to improving the model predictions, knowing the specific metabolite rates is important to ensuring the data driven model matches the physical properties of the system. As described in Section 6.3.1 the metabolite concentrations have certain effects on the process that must be represented in the data driven model.

6.5.1 Metabolite Rate Model Identification

The metabolite rates are an important part of the bioreactor process as they determine how the outputs from the dynamic mode change with respect to the viable cell density. Analyzing the metabolite rates is an important part of determining the ideal input conditions required for optimal growth in each stage. The specific metabolite rates are calculated as follows:

$$R_{m_t}(x_v) = \frac{m_t(t+h) - m_t(t)}{ix_v} \tag{6.4}$$

$$ix_v = \frac{0.6x_v(t) + 0.4x_v(t+h)}{h} \tag{6.5}$$

Where R_{m_t} denotes the metabolite rate for a metabolite m_t, x_v represents the viable cell density, ix_v represents the integrated viable cell density and h represents the sampling interval. The modeling approach calculates metabolite rates using three inputs (glucose concentration, temperature setpoint and pH setpoint) and 5 outputs (glucose rate, glutamine rate, lactate rate, glutamate rate and ammonia rate), and then builds a model to directly predict the metabolite rates. A metabolite rate model of order 3 was identified based on training fit results.

6.5.2 Metabolite Rate Model Validation

The training fit and validation error is shown in Table 6.2 and are similar in magnitude. As shown in Figure 6.6 for the training data and in Figure 6.7 for the validation batch, the metabolite rates have a large amount of daily fluctuation. These trends are key to understanding the overnight behavior of the process and the validation fit in Figure 6.7 shows how the metabolite rate model is able accurately model the rates.

Table 6.2: The prediction error between the subspace based metabolite rate model and the process for both the training and validation batches.

Model	*Fit Error*
Training	4.7848
Validation	4.9276

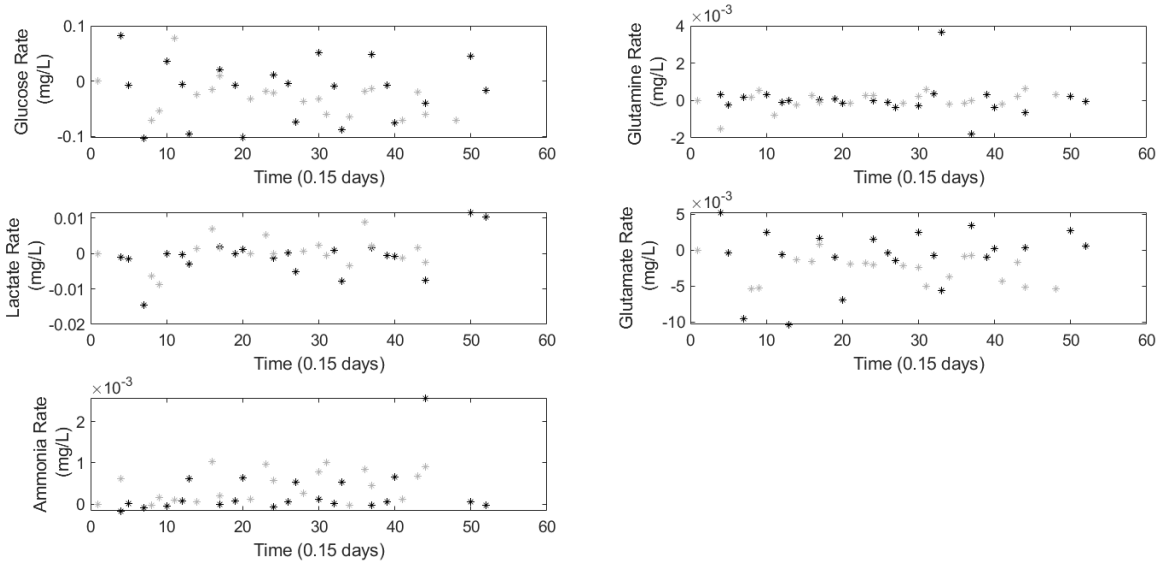


Figure 6.6: The output predictions (grey) from the metabolite model for each output are compared against the process data (black) for one training batch.

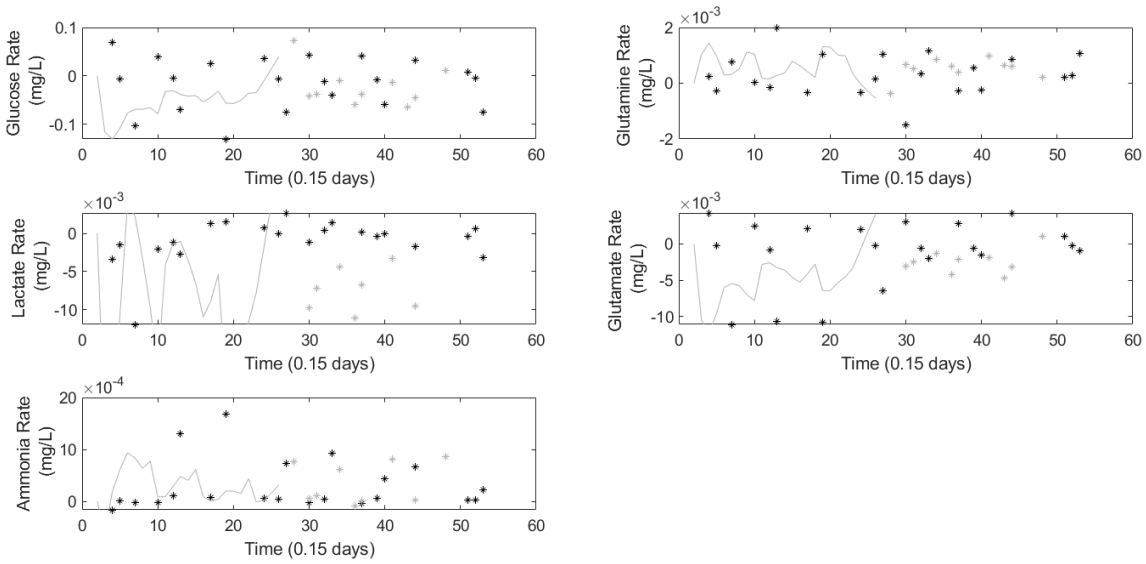


Figure 6.7: The process data (black) is compared with the metabolite rate model predictions using the state observer (grey solid) until the states converge and then the metabolite rate model predicts the remainder of the validation batch (grey starred).

For comparison the metabolite rates are calculated using the dynamic model and shown in Figure 6.8 below and compared to the metabolite rates calculated using the measurements. As seen in this figure the rate predictions calculated from the predicted measurements do not match very well with the rates calculated using the measurements themselves. In essence, the errors in the predictions of the variables get much more enlarged when using them in the calculations of the metabolite rate. The calculated rates differ by a magnitude of ten in comparison to the rates calculated based on the measurements. The advantage of directly modeling the metabolite rates is clearly demostrated as the calculated rate rely on the model predictions of the viable cell density and the metabolites. Thus, small errors in these variables compound resulting in a poor result in the prediction of the metabolite rates. Another drawback of calculating the model parameters is that in the glucose consumption rate. In the modeling approach, glucose is utilized as an input [12]. Therefore, using the calculated parameters to identify the glucose rate is not meaningful when attempting to use this model for control purposes. Given the limitations using calculated rates and the

inability to model glucose the use of a separate model to identify the metabolite rates is necessary.

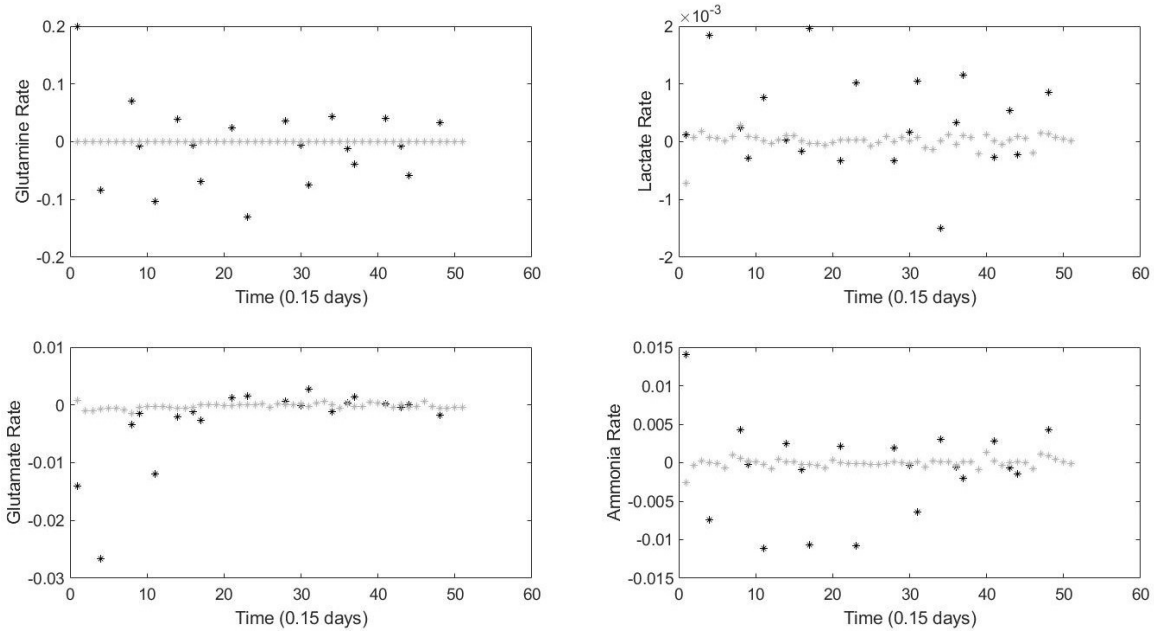


Figure 6.8: The process data (black) is compared with the model predictions using the state observer (grey solid) until the states converge and then the model predicts the remainder of the batch (grey starred).

6.6 Conclusions

In this work, the problem of identifying a dynamic batch model with large amounts of missing data was solved using a modified subspace identification procedure. The Sartorius bioreactor problem also had discrete inputs from the glucose feed additions which were modeled as instantenous additions to great effect. The dynamic modeling approach used the NIPALS algorithms to gracefully handle missing data allowing for accurate model predictions of the validation batches. When modeling interpretable variables like metabolite rate the modified approach is shown to be more accurate in comparison to calculating the rates from process measurements.

6.7 Acknowledgement

Financial support from Sartorius and the McMaster Advanced Control Consortium is gratefully acknowledged.

Bibliography

[1] Bernard, O., Mairet, F., and Chachuat, B. (2015). Modelling of microalgae culture systems with applications to control and optimization. In *Microalgae Biotechnology*, pages 59–87. Springer.

[2] Chusainow, J., Yang, Y. S., Yeo, J. H., Toh, P. C., Asvadi, P., Wong, N. S., and Yap, M. G. (2009). A study of monoclonal antibody-producing cho cell lines: What makes a stable high producer? *Biotechnology and bioengineering*, 102(4):1182–1196.

[3] Corbett, B. and Mhaskar, P. (2016). Subspace identification for data-driven modeling and quality control of batch processes. *AIChE Journal*, 62(5):1581–1601.

[4] Corbett, B. and Mhaskar, P. (2017). Data-driven modeling and quality control of variable duration batch processes with discrete inputs. *Industrial & Engineering Chemistry Research*, 56(24):6962–6980.

[5] Deschenes, J.-S., Desbiens, A., Perrier, M., and Kamen, A. (2006). Multivariable nonlinear control of biomass and metabolite concentrations in a high-cell-density perfusion bioreactor. *Industrial & engineering chemistry research*, 45(26):8985–8997.

[6] Dochain, D. and Perrier, M. (1997). Dynamical modelling, analysis, monitoring and control design for nonlinear bioprocesses. In *Biotreatment, Downstream Processing and Modelling*, pages 147–197. Springer.

[7] Flores-Cerrillo, J. and MacGregor, J. F. (2004). Control of batch product quality by trajectory manipulation using latent variable models. *Journal of Process Control*, 14(5):539–553.

[8] Ghosh, D., Hermonat, E., Mhaskar, P., Snowling, S., and Goel, R. (2019). Hybrid

modeling approach integrating first-principles models with subspace identification. *Industrial & Engineering Chemistry Research*, 58(30):13533–13543.

[9] Hu, B., Zhao, Z., and Liang, J. (2012). Multi-loop nonlinear internal model controller design under nonlinear dynamic pls framework using arx-neural network model. *Journal of Process Control*, 22(1):207–217.

[10] Huang, B., Ding, S. X., and Qin, S. J. (2005). Closed-loop subspace identification: an orthogonal projection approach. *Journal of process control*, 15(1):53–66.

[11] Jiang, Q., Yan, X., and Huang, B. (2015). Performance-driven distributed pca process monitoring based on fault-relevant variable selection and bayesian inference. *IEEE Transactions on Industrial Electronics*, 63(1):377–386.

[12] Karra, S., Sager, B., and Karim, M. N. (2010). Multi-scale modeling of heterogeneities in mammalian cell culture processes. *Industrial & Engineering Chemistry Research*, 49(17):7990–8006.

[13] Kheradmandi, M. and Mhaskar, P. (2018). Adaptive model predictive batch process monitoring and control. *Industrial & Engineering Chemistry Research*, 57(43):14628–14636.

[14] Larimore, W. E. (1996). Statistical optimality and canonical variate analysis system identification. *Signal Processing*, 52(2):131 – 144. Subspace Methods, Part II: System Identification.

[MacGregor et al.] MacGregor, J. F., Jaeckle, C., Kiparissides, C., and Koutoudi, M. Process monitoring and diagnosis by multiblock pls methods. *AIChE Journal*, 40(5):826–838.

[16] Mairet, F., Bernard, O., Cameron, E., Ras, M., Lardon, L., Steyer, J.-P., and Chachuat, B. (2012). Three-reaction model for the anaerobic digestion of microalgae. *Biotechnology and Bioengineering*, 109(2):415–425.

[17] Markovsky, I. (2013). Exact system identification with missing data. In *52nd IEEE Conference on Decision and Control*, pages 151–155. IEEE.

[18] Moonen, M., De Moor, B., Vandenberghe, L., and Vandewalle, J. (1989). On-and off-line identification of linear state-space models. *International Journal of Control*, 49(1):219–232.

[19] Morel, E., Tartakovsky, B., Guiot, S., and Perrier, M. (2006). Design of a multi-model observer-based estimator for anaerobic reactor monitoring. *Computers & chemical engineering*, 31(2):78–85.

[20] Patel, N., Nease, J., Aumi, S., Ewaschuk, C., Luo, J., and Mhaskar, P. (2020). Integrating data-driven modeling with first-principles knowledge. *Industrial & Engineering Chemistry Research*, 59(11):5103–5113.

[21] Qin, S. J. (2006). An overview of subspace identification. *Computers and Chemical Engineering*, 30(10-12):1502–1513.

[22] Shang, L., Liu, J., Turksoy, K., Shao, Q. M., and Cinar, A. (2015). Stable recursive canonical variate state space modeling for time-varying processes. *Control Engineering Practice*, 36:113–119.

[23] Sirois, J., Perrier, M., and Archambault, J. (2000). Development of a two-step segregated model for the optimization of plant cell growth. *Control Engineering Practice*, 8(7):813–820.

[24] Van Overschee, P. and De Moor, B. (1994). N4sid: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30(1):75–93.

[25] Van Overschee, P. and De Moor, B. (1995). A unifying theorem for three subspace system identification algorithms. *Automatica*, 31(12):1853–1864.

[26] Van't Riet, K. and Tramper, J. (1991). *Basic bioreactor design*. CRC press.

[27] Verhaegen, M. and Dewilde, P. (1992). Subspace model identification part 2. analysis of the elementary output-error state-space model identification algorithm. *International journal of control*, 56(5):1211–1241.

[28] Wang, J. and Qin, S. J. (2002). A new subspace identification approach based on principal component analysis. *Journal of process control*, 12(8):841–855.

[29] Xie, L. and Wang, D. I. (1996). High cell density and high monoclonal antibody production through medium design and rational control in a bioreactor. *Biotechnology and bioengineering*, 51(6):725–729.

Chapter 7

Process-Aware Data Driven Modeling and Model Predictive Control of Bioreactor for Production of Monoclonal Antibodies

This chapter presents a novel subspace modeling approach for the Sartorius Bioreactor. The key idea was that the bioreactor is operated as a continuous process meaning that existing batch trajectories were no longer applicable. The goal of this work was to design a comprehensive model utilizing all the methods described in the previous chapters of this thesis. The bioreactor has several first-principles based constraints required in the model, large amounts of missing data, and discrete inputs. This work was completed in collaboration with a master's student Samardeep Sarna who is under my supervision with Dr. Mhaskar. As such my principal contribution was to develop the methods used in this work and assist in writing this work. This paper is still included as it showcases a practical application of all the previous approaches developed in the thesis. This was also done with industrial partner Sartorius who provided

insight into the industrial bioreactor and provided data and modeling experience. The paper is currently in the review process.

Sarna, S., Patel, N., Corbett, B., McCready, C., & Mhaskar, P. Submitted Computers & Chemical Engineering, (2022).

7.1 Abstract

This manuscript addresses the problem of controlling a bio-reactor to maximize the production of a desired product while respecting the constraints imposed by the nature of the bio-process. The approach is demonstrated by first building a data driven model and then formulating a model predictive controller (MPC) with the results illustrated by implementing on a detailed monoclonal antibody production model (the test bed) created by Sartorius Inc. In particular, a recently developed data driven modeling approach using an adaptation of subspace identification techniques is utilized that enables incorporation of known physical relationships in the data driven model development (constrained subspace model identification) making the data-driven model process aware. The resultant controller implementation demonstrates significant improvement in product compared to the existing PI controller strategy used in the monoclonal antibody production. Simulation results also demonstrate the superiority of the process aware or constrained subspace model predictive controller compared to traditional subspace model predictive controller. Finally, the robustness of the controller design is illustrated via implementation of a model developed using data from a test bed with a different set of parameters, thus showing the ability of the controller design to maintain good performance in the event of changes such as a different cell line or feed characteristics.

7.2 Introduction

The need for bio-based and pharmaceutical products is on the rise with advancements in healthcare and the demand of an ever-increasing global population. Bioreactors form an important part of this industry by allowing for the mass production of these bio-pharmaceutical products. One such product is a monoclonal antibody which is

produced by Sartorius and is used to demonstrate the model based controller design approach. The Sartorius Bioreactor is designed to produce this protein in a perfusion processing setup.

There exist several challenges associated with control of bio reactors in general, and the monoclonal antibody production in consideration, in particular (herewith referred to as the Sartorius Bioreactor). First off, in contrast to batch or fed-batch processing [22, 6, 26] the Sartorius Bioreactor is operated in perfusion mode (thus is a continuous removal of bleed and harvest streams). In addition to the perfusion mode of operation, there are several other factors, such as hydrodynamics and transport phenomena [16], that affect the volumetric production of the monoclonal antibodies. Factors like cell growth rate, feed rate and feed concentration are all key variables in bioreactor operation and thus, these factors need to be accounted for in order to maximize the final product. The final product is a combination of the volumetric flow rate (referred to as harvest rate) and a high volume specific concentration of the antibody (referred to as titer). In order to maximize the final product the individual interactions between different inputs, outputs and other parameters must first be examined. The first and most important parameter to consider is glucose concentration, as glucose is the key energy source, however, excess glucose is also detrimental due to lactate production which increases cell death. Similarly, Glutamine plays an important role especially for promoting cell growth during periods of fast growth. Lactate and more so, ammonia, are inimical to cell growth[14]. These metabolites are not the only factors affecting cell growth. Both temperature and pH play a key role. Increasing temperature has been shown to increase cell growth rate. However, high temperatures can also lead to cell death. To handle this issue, increased antibody production is achieved with a midway temperature shift [4, 29]. A more complex variable is pH since pH levels also affect ammonia and lactate levels. Often, a shift in pH in later stages is necessary [14]. Further, due to operational considerations, it is preferable to decrease the pH rather than increase it since it can be decreased by sparging CO_2 but increasing pH would

require the addition of a base that could potentially disturb the cell environment negatively. In essence, since the production of a specific product such as a protein by these cells is heavily affected by the environment in the reactor, such as the pH and glucose levels [2, 24, 26] and with such a diversity of variables affecting the system dynamics with many of these having contradictory effects in different ranges, the modelling and control problem is a challenging one.

With the increasing recognition of the flexibility provided by process control in process operation, process control is being adopted within the bio-processing industry [25]. One popular and successful control strategy that has been used in large scale production is model predictive control (MPC). MPC relies on a process model to calculate the optimal input trajectory to meet desired objectives while respecting constraints or bounds. MPC has been implemented in chemical industries and the energy sector with favorable results. In recent years it has also been implemented for biochemical and fermentation processes [19, 15, 3]. However, MPC of bioreactors is not common in industry due to the sensitive nature of the cells and the set batch recipes available. Instead proportional integral (PI) control is used to follow a batch trajectory. The use of PI control however, potentially limits the productivity of the process (as illustrated by the results in this manuscript) motivating the need to explore the implementation of MPC.

In an MPC implementation, the process model forms the heart of the entire strategy therefore identifying a good model is critical to improved control. When modelling a system, first principles models are valuable since they provide a direct insight into the process. Although parameter estimation for first principles models is challenging, parameter estimation methods for first principles models exists in literature [8, 27, 7, 20, 17, 1], and this has been applied to bioreactors [14]. More recently, Sartorius Inc. has developed a high fidelity simulator for the monoclonal antibody process, and is used in the present manuscript to illustrate the control approach. The

detailed simulator, while being a good representation of the bioreactor, is not very suitable for direct incorporation in an MPC formulation due to model complexity. More importantly, it is of much more benefit to the practitioner to demonstrate the implementation of a control approach that can readily utilize process data directly for model development and control implementation.

Data driven and black box models are one choice for ease of implementation[6, 30]. Reduced order models can also achieve high performance control if it is possible to capture basic and fundamental dynamical features of the system. The performance of the controller is often the main objective for model building in these instances and thus such kind of models are valuable [12]. Within data-driven methods, there are several different approaches; however, not all such approaches are suitable for the Sartorius bioreactor problem. One particular concern is that the complex metabolite interactions require specific gains that must be adhered to in the data driven model. To that end any modeling approach must be capable of incorporating these constraints with minimal complexity. Techniques such as Partial Least Squares (PLS) do not explicitly differentiate between inputs and outputs or handle multiple batches without additional complexity [11, 10]. To that end, an approach involving Linear Time Invariant (LTI) models would be better suited to handle this problem. One such approach is subspace identification, which is a well established system identification method and has several advantages such as having only one decision variable (the order of the system) and its ability to handle large multi-input multi-output (MIMO) problems well. The model complexity of MIMO and the simpler single-input single-output (SISO) systems is similar when using subspace identification. This is in contrast to methods such as Auto-Regressive Moving Average with eXogenous inputs (ARMAX) models which have multiple 'tuning' parameters. In comparison to methods such as ARMAX, subspace identification is often easier to implement, faster and more accurate, including cases with white noise. [9] Additionally, recent results have allowed imposing constraints in subspace identification at the modeling stage with minimal

additional computational complexity [22], to enable the model to be more 'aware' of the process.

Motivated by the above considerations, the present work addresses the problem of maximizing the production in a Sartorius bioreactor using MPC with a process aware or constrained subspace model. Specifically, a process aware subspace MPC is implemented on the simulation test-bed and compared against existing PI control. Next the need to implement process aware MPC is demonstrated by comparing against a traditional subspace model based MPC. Finally, the robustness of the MPC approach is tested by comparing the MPC against a new process with different system dynamics. The rest of the paper is arranged as following: Section 7.3 described the bioreactor process, reviews subspace identification and constrained subspace identification. The model predictive control scheme which is developed and used is presented in section 7.4. Section 7.5 presents the application of the proposed method to the Sartorius Bioreactor. Concluding remarks are presented in Section 7.6.

7.3 Preliminaries

7.3.1 Bioreactor Process Description

The Sartorius Bioreactor grows live cells in an enclosed environment meaning that the growth and death rates affect the environment in the reactor and consequently the titer (final product). A simplified schematic of the bioreactor is shown in figure 7.1. The recycle stream shown in the figure recycles live cells as a cell retention filter does not allow live cells to leave in the harvest stream.

A detailed first principles model developed by Sartorius is used as a test bed in the present manuscript. The Sartorius simulator comprises a system of 10 ordinary

differential equations to describe the time evolution of variables including the cells and metabolites (characterized by viable cell density (VCD), dead cell density, lysed cell density, biomaterial, titer, glucose, glutamine, lactate, ammonia and glutamate). The parameters, and various function describing growth rates etc are determined by fitting the model to experimental data from twelve AMBR 250 fed batch runs to yield a biologically meaningful and fairly accurate description of the bioreactor. Transferrability of this model structure from fed-batch to perfusion operation has been established by Sartorius researchers, and as such, the present model is being utilized to demonstrate the data driven modeling and control approach.

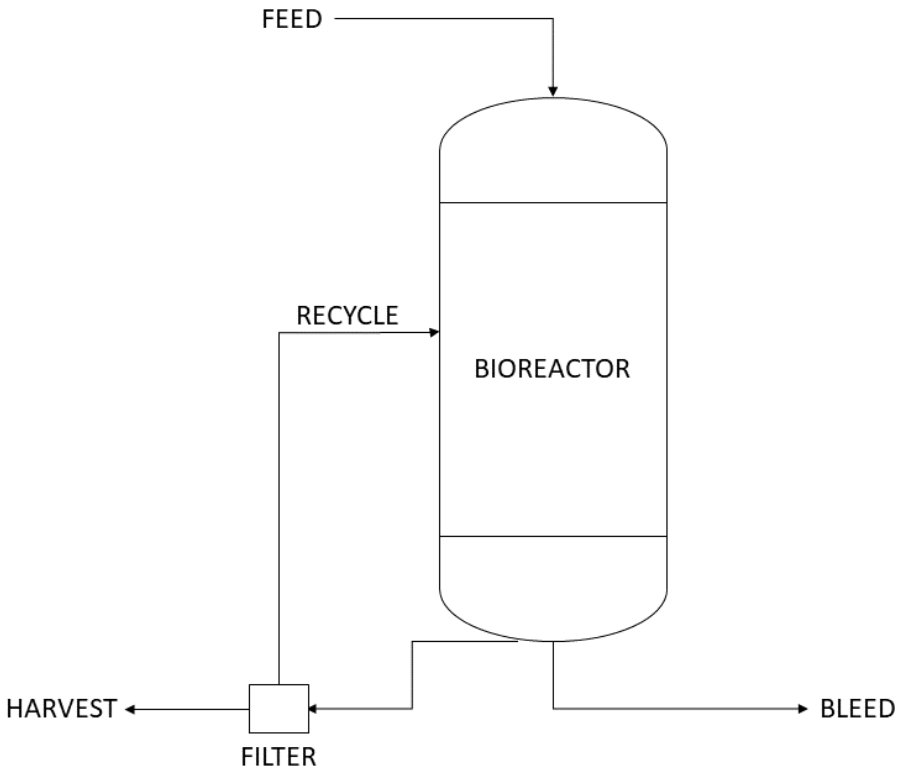


Figure 7.1: Schematic of Sartorius Bioreactor

The process initiates in growth phase for 3 days during which the system is operated in a fed batch fashion. This is followed by a perfusion phase for 30 days. In this work, based on the specific process used by Sartorius, the nominal values for the

temperature and pH are set at $36.1°C$ and 7.1 respectively. The reactor temperature (°C), pH, glucose feed concentration (g/L), feed rate (vols/day) and bleed rate(L/day) are available as potential inputs. The measured outputs are viability (%), viable cell density (VCD) (10^5 cells/mL)], titer (mg/L) and glucose concentration (g/L). The inputs and outputs are organized in the following vectors:

$$u = \begin{bmatrix} Reactor\ Temp \\ pH \\ Feed\ Conc \\ Feed\ Rate \\ Bleed\ Rate \end{bmatrix}$$

$$y = \begin{bmatrix} Viability \\ VCD \\ Titer \\ Glucose\ Conc \end{bmatrix}$$

The process objective is to maximize bioreactor production over the course of the perfusion phase which is currently done by putting VCD under PI control where with a fixed setpoint of 50. The PI controller that is currently employed adjusts the the bleed rate in order to control the VCD. With the feed rate kept constant at 0.25L/day or 1.25 volumes/day and under constant volume operation, the harvest rate can be computed as:

$$Harvest\ Rate = Feed\ Rate - Bleed\ Rate \tag{7.1}$$

As the bleed rate is the most significant contributor to cell growth, it is utilized as the control variable with additional shifts in temperature or pH being applied by the operators manually. The objective of the present work is to demonstrate the possibility

of using a data driven MPC to control and improve the bioprocess operation.

7.3.2 Subspace Identification Description

Subspace identification is one model identification technique that is used to identify a Linear Time Invariant (LTI) model of the form:

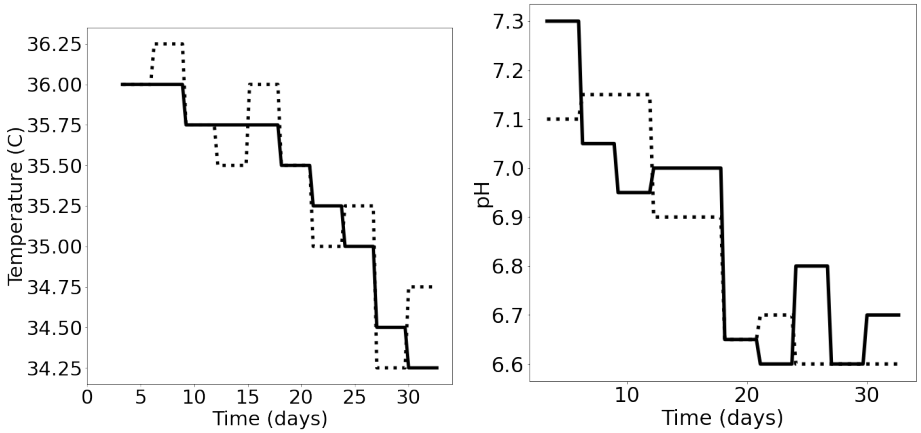
$$\begin{aligned} \hat{\mathbf{x}}[k+1] &= \mathbf{A}\mathbf{x}[k] + \mathbf{B}\mathbf{u}[k], \\ \mathbf{y}[k] &= \mathbf{C}\hat{\mathbf{x}}[k] + \mathbf{D}\mathbf{u}[k], \end{aligned} \tag{7.2}$$

where the objective is to identify the order n, and the system matrices $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times m}$, $\mathbf{C} \in \mathbb{R}^{l\times n}$, $\mathbf{D} \in \mathbb{R}^{l\times m}$. The particular subspace adaptation utilized in the present work is originally based off of [18], which was later adapted for batch processes[5].

An important consideration for the process under consideration is to ensure that the bioreactor is not disturbed too frequently otherwise the cell balance will be disrupted leading to inefficient protein production and additional costs. Thus an appropriate frequency of input changes is utilized in collecting data such that it has the fewest number of perturbations while meeting reasonable prediction accuracy. Based on a preliminary analysis, perturbation of inputs three times per day is utilized in the present work. An additional consideration is that the process runs in growth phase for 3 days followed by perfusion phase for 30 days thus it takes over a month of time (and significant costs) to generate data. To be able to demonstrate the approach, data is generated from a detailed simulation test bed provided by Sartorius. The data was generated by gradual shifts in the inputs over their appropriate constrained ranges along with small perturbations. Data was obtained from a single batch run over thirty days with measurements available thrice a day (for a total of 90 measurements). Note

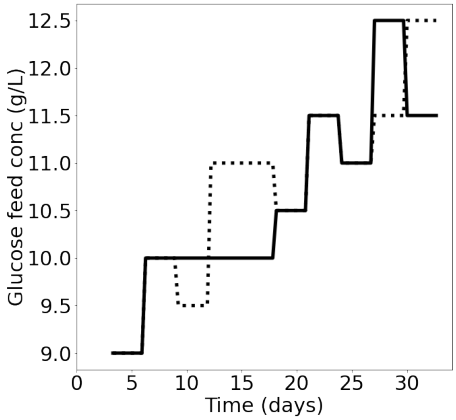
that the ability to use this relatively modest dataset is extremely important for the process under consideration where each run is prohibitively expensive.

The Sartorius simulator is used to generate input-output trajectory for one run, and this data is assumed to be available for building the data driven MPC, and shown in figures 7.2 and 7.3.

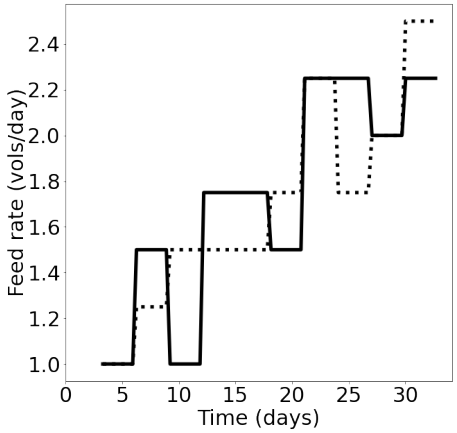


(a) Temperature

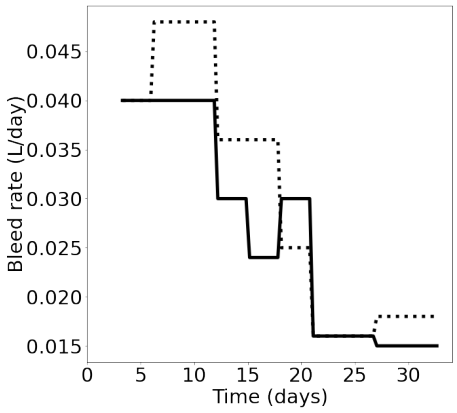
(b) pH



(c) Glucose feed concentration

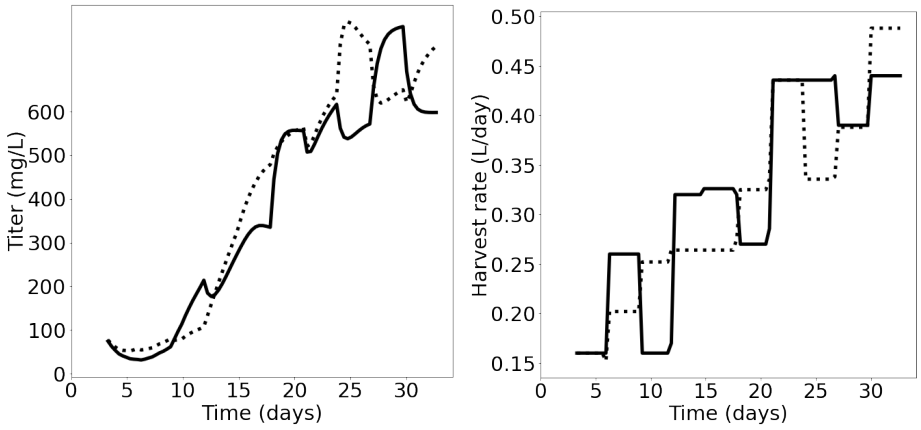


(d) Feed rate



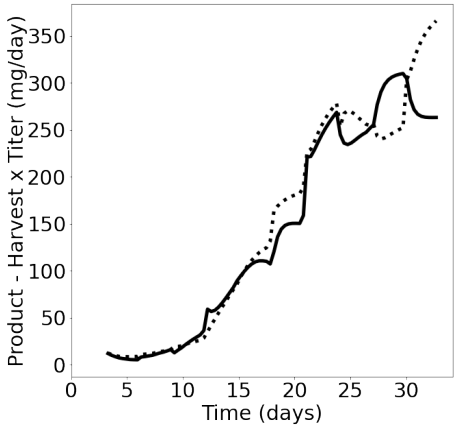
(e) Bleed rate

Figure 7.2: Input data for building model with training (dotted) and validation data (solid).

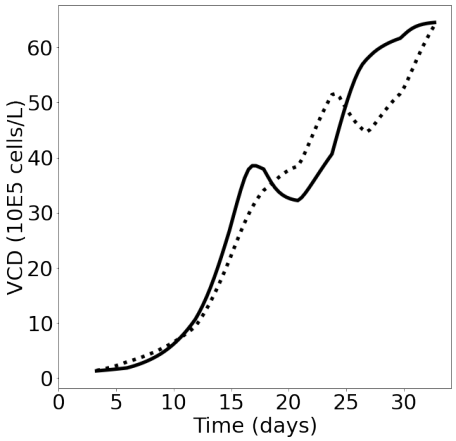


(a) Titer

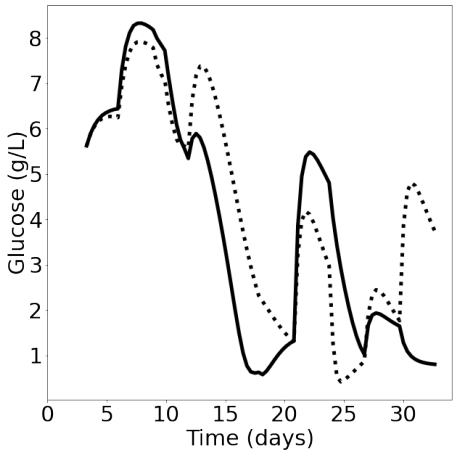
(b) Harvest



(c) Product (Titer × Harvest)



(d) Viable Cell Density (VCD)



(e) Glucose Concentration in Reactor

Figure 7.3: Output data for building model with training (dotted) and validation data (solid).

Identification of the system matrices is done in two stages, first stage involves identifying a state sequence and the second stage comprises of identifying the system matrices [18]. Using subspace identification, the state sequence can be identified using methods such as SVD before knowing the A,B,C,D system matrices. The system matrices are later identified by least squares regression.

In solving for the state sequence, block Hankel matrices are constructed for the inputs and outputs. The number of block rows (i) and columns (j) are chosen sufficiently large, typically i should be greater than or equal to $n+1$ and $j >> max(mi, li)$.

The output and input block Hankel matrices are:

$$Y_p = \begin{bmatrix} y[k] & y[k+1] & \dots & y[k+j-1] \\ y[k+1] & y[k+2] & \dots & y[k+j] \\ y[k+2] & y[k+3] & \dots & y[k+j+1] \\ \vdots & & & \vdots \\ y[k+i-1] & y[k+i] & \dots & y[k+i+j-2] \end{bmatrix}$$

$$U_p = \begin{bmatrix} u[k] & u[k+1] & \dots & u[k+j-1] \\ u[k+1] & u[k+2] & \dots & u[k+j] \\ u[k+2] & u[k+3] & \dots & u[k+j+1] \\ \vdots & & & \vdots \\ u[k+i-1] & u[k+i] & \dots & u[k+i+j-2] \end{bmatrix}$$

Y_f and U_f are defined similar to Y_p and U_p except the values are offset by i. These matrices are used to identify the state vector which can be organized in a Hankel matrix allowing the A,B,C,D matrices to be solved by least squares regression.

$$\begin{bmatrix} x[k+i+1]\dots & x[k+i+j] \\ y[k+i]\dots & y[k+i+j-1] \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x[k+i]\dots & x[k+i+j-1] \\ u[k+i]\dots & u[k+i+j-1] \end{bmatrix} \tag{7.3}$$

Note that the approach identifies a linear state space model where the states are unmeasured but observable from measured outputs. As subspace states are not measured, it is necessary to estimate the states before using the subspace model for prediction/validation. To this end, during model validation an initial state estimate is chosen (can be based on a state estimate from identification or a random initialization) and a state observer is utilized. The state observer is run until the error (Euclidean norm) between the predicted output and actual/observed output is below a chosen tolerance, from which point on the model can be utilized for prediction purposes. This same approach is utilized when the model is used for feedback control. Note that Sartorius Bioreactor operation has the unique advantage of an initial 3 day growth phase (without any feedback control) that can be used to converge the states allowing the controller to be used online immediately after the growth phase ends.

In this work a Luenberger observer was used which takes the following form:

$$\hat{\mathbf{x}}[k+1] = \mathbf{A}\hat{\mathbf{x}}[k] + \mathbf{B}\mathbf{u}[k] + \mathbf{L}(\mathbf{y}[k] - \hat{\mathbf{y}}[k]) \tag{7.4}$$

where $\mathbf{L}$ is the observer gain and is chosen such that $(\mathbf{A} - \mathbf{LC})$ is stable. $\hat{\mathbf{y}}[k]$ is the predicted value given by the state space equation $\mathbf{y}[k] = \mathbf{C}\hat{\mathbf{x}}[k] + \mathbf{D}\mathbf{u}[k]$.

7.3.3 Constrained Subspace Identification

In this subsection the approach used to impose the physical constraints on the subspace model is described [23]. The key idea is to include the first-principles knowledge

of the system at the model identification stage through the use of constraints to make the data-driven model process aware. In this approach, instead of using regression to determine the model parameters (7.3) an optimization problem with the first principles based constraints is posed and solved. Thus, while the initial state trajectory may have been determined without considering the physical constraints, the resultant matrices do account for the presence of constraints. To minimize the discord between the state trajectory and the constraints, the state trajectory is re-estimated using the newly computed system matrices, and this iterative process terminated when a pre-decided tolerance is achieved (see [23] for further details).

For the Bioreactor, it is understood that the steady state gain between the bleed rate and titer (product) should be negative. Similarly a positive relation holds for temperature, glucose feed concentration and feed rate and is incorporated into the constrained subspace model through constraints. The constraints are therefore mathematically formulated as:

$$\begin{aligned} dcgain(3,5) + 0.01 <= 0 \\ -dcgain(3,1) + 0.01 <= 0 \\ -dcgain(3,3) + 0.01 <= 0 \\ -dcgain(3,4) + 0.01 <= 0 \\ norm(eig(A)) - 0.99 <= 0 \end{aligned} \tag{7.5}$$

where $dcgain(i,j)$ refers to the $(i,j)^{th}$ index of the steady state gain matrix, which for a discrete linear time invariant (LTI) system is:

$$dcgain = D + C(I - A)^{-1}B \tag{7.6}$$

Thus the first constraint specifies $dcgain(3,5)$ to be negative. That is, the gain be-

tween the third output (titer) and the fifth input (bleed rate) should be negative. Similarly, the other gain constraints enforce the positive steady state gain relationship between the titer and the temperature, feed concentration and feed rate, respectively. The final constraint is for the eigenvalues of the identified A matrix to lie within the unit circle.

Remark 30. *Subspace identification was chosen as the model identification approach due to the following reasons: 1) The method results in a linear time invariant model which in turn makes the resultant control problem easy to solve and implement, 2) Compared to other approaches such as Projection to Latent Spaces (PLS), the method explicitly accounts for the presence of input and output variables, consistent with a control implementation and 3) Even though the model parameters are eventually determined using a regression, the state trajectory first invokes the key property of a state- that future outputs should be able to be completely determined using the current states and the future inputs, thus avoiding potential overfitting issues. The implementation does not require a first principles model- in the present manuscript, the first principles model is simply used as a test bed. Finally, the method can be readily adapted to incorporate first principles information either explicitly, as is done in the present manuscript, or through a hybrid model [13].*

Remark 31. *Yes another possibility for model identification would be to use the resurgent techniques of artificial neural networks. For developing a good neural network model, large amounts of data is needed. While it is possible in principle with the test bed, it would not be very feasible when this technique is implemented in practice on the Sartorius Bioreactor. Note that the data set size utilized for training in the present work was chosen while being cognizant of the cost and effort needed to generate data from bioreactor.*

7.4 Model Predictive Controller Formulation

In this section, the state space MPC formulation adapted to use the feedthrough matrix [21] utilizing the subspace model of Eqn 7.3 is described. A Python script utilizing scipy.optimize [28] is used to solve the optimization problem. At the l^{th} time step, with the observer determining $\hat{x}[l]$, the following optimization problem is solved to compute the control action:

$$
\begin{aligned}
\min_{\bar{u}} \sum_{i=1}^{P} (y_3[i])^T Q(u_4[i] - u_5[i]) \\
+(\Delta u^T) R(\Delta u) + S\Delta u \\
\text{s.t.} \\
u_{min} \leq \bar{u}[i] \leq u_{max} \ , \ i = 1, \ldots, P \\
\Delta u = \bar{u}[i] - \bar{u}[i-1] \ , \ i = 2, \ldots, P \\
\Delta u[1] = \bar{u}[1] - u[l-1] \\
x[1] = \hat{x}[l] \\
x[i+1] = Ax[i] + B\bar{u}[i] \ , \ i = 1, \ldots, P \\
y[i] = Cx[i] + D\bar{u}[i] \ , \ i = 1, \ldots, P
\end{aligned}
\tag{7.7}
$$

where y denotes the predicted output obtained from Eqn 7.3 and as described in section 6.3.1 y_3 corresponds to the titer, u is the input vector and $\bar{u}$ is the optimization variable i.e. the inputs MPC computes, and sets $u[l] = \bar{u}[1]$, and u_{min} and u_{max} are the vectors corresponding to the lower and upper bounds respectively for the inputs (see Table 7.1). The bounds are kept commensurate to the usual practice in industry and hence feed rate, albeit important and strongly related to maximizing product, has been given a smaller upper bound. An effort has been made to impute any increase in product to the other strongly related variables such as glucose by tuning

the weight appropriately. P denotes the prediction horizon, Q is a negative value picked to appropriately weigh the product maximization in the objective function. R is a diagonal matrix with appropriate penalties for input change. S is a scalar weight to additionally penalize a positive change in pH. This term has been chosen to not be a quadratic term specifically to penalize only positive changes. Note that the implementation of a positive pH change is done via using a buffer, which 'shocks' the cells and is preferably avoided.

Table 7.1: Input Constraints

u	units	u_{min}	u_{max}	u_{nom}
Temp	degC	35	36.8	36.1
pH		6.95	7.15	7.1
Glucose Feed Conc.	mg/L	6	12	9
Feed Rate	vols/day	1	1.6	1.25
Bleed Rate	L/day	0.01	0.05	0.025

Remark 32. *Note that in this formulation the predicted outputs are determined using a state space model with a feedthrough term. When identifying a data driven model it has been shown that retaining the feedthrough term provides more accurate control comparing to dropping it, motivating the use of the recent MPC formulation in the present work [21].*

The elements of the matrix R in the term $\Delta u^T R \Delta u$ are taken as the inverse of the nominal value of that input variable to compensate for the different scales of the manipulated inputs. The value S is utilized to specifically penalize positive changes in the pH. To accomplish this we adjust the elements in R and S such that the increase to the objective due to pH change from term R is higher than the decrease due to term S when the pH decreases. In case of a candidate positive pH change, since R and S have positive weights, the pH terms add up from $(\Delta u^T) R (\Delta u)$ which is always positive due to being quadratic and $S\Delta u$ which is positive since Δu is positive. In

the case of a negative pH change, the positive contribution from $(\Delta u^T)R(\Delta u)$ would outweigh the negative contribution from $S\Delta u$ for any reasonable change in pH when s is small. The value of s is taken as 0.05. The value of the R term corresponding to pH is $\frac{10}{7.1}$. For a Δu of -0.05, the $(\Delta u^T)R(\Delta u)$ term is +0.0035 and $S\Delta u$ term is -0.0025 resulting in a net +0.001 penalty. Thus, a change to the pH would be only made if it results in a net benefit to the product quality. On the other hand, for a candidate increase in pH, for a Δu of +0.05, the $(\Delta u^T)R(\Delta u)$ term is +0.0035 and $S\Delta u$ term is +0.0025 resulting in a net +0.006 penalty, a six times higher penalty than a corresponding decrease.

The net affect of such a choice of the tuning parameters and the formulation is that any significant pH changes are penalized but increases in pH are penalized more than decreases in pH.

The MPC is initialized when the error between predicted outputs and observed outputs (using the Luenberger observer) becomes smaller than a user specified tolerance. The tolerance is chosen such that there is minimal plant-model mismatch but also enough time left to implement an MPC strategy. Before the state observer converges, a constant nominal input is applied to the process.

$$Q = q_1 \tag{7.8}$$

$$R = \begin{bmatrix} \frac{r}{u_{nom,1}} & 0 & 0 & 0 & 0 \\ 0 & \frac{r}{u_{nom,2}} & 0 & 0 & 0 \\ 0 & 0 & \frac{r}{10\times u_{nom,3}} & 0 & 0 \\ 0 & 0 & 0 & \frac{r}{u_{nom,4}} & 0 \\ 0 & 0 & 0 & 0 & \frac{r}{u_{nom,5}} \end{bmatrix} \tag{7.9}$$

$$S = s \tag{7.10}$$

Table 7.2 reports q_1, r and s while u_{nom} values are reported in table 7.1. Since the inputs vary in orders of magnitude, the weights on input change penalty are also adjusted as such with their respective nominal values.

Table 7.2: Tuning parameters

q_1	r	s
-1	10	0.05

Remark 33. *The objective function in the MPC formulation focuses on maximizing the final product which depends on both the titer (similar to output concentration) and the harvest rate (similar to output flow rate). The specific objective function can readily be altered. The key contribution of the present manuscript are not the input profiles that the controller implements, but to demonstrate that a data driven model based MPC, with a meaningful objective function, can be implemented on the system (the test bed in this case) and a biologically acceptable control action and system behavior achieved. This objective function can be further fine tuned or changed based on the specific needs of the process operation.*

7.5 Results and Discussion

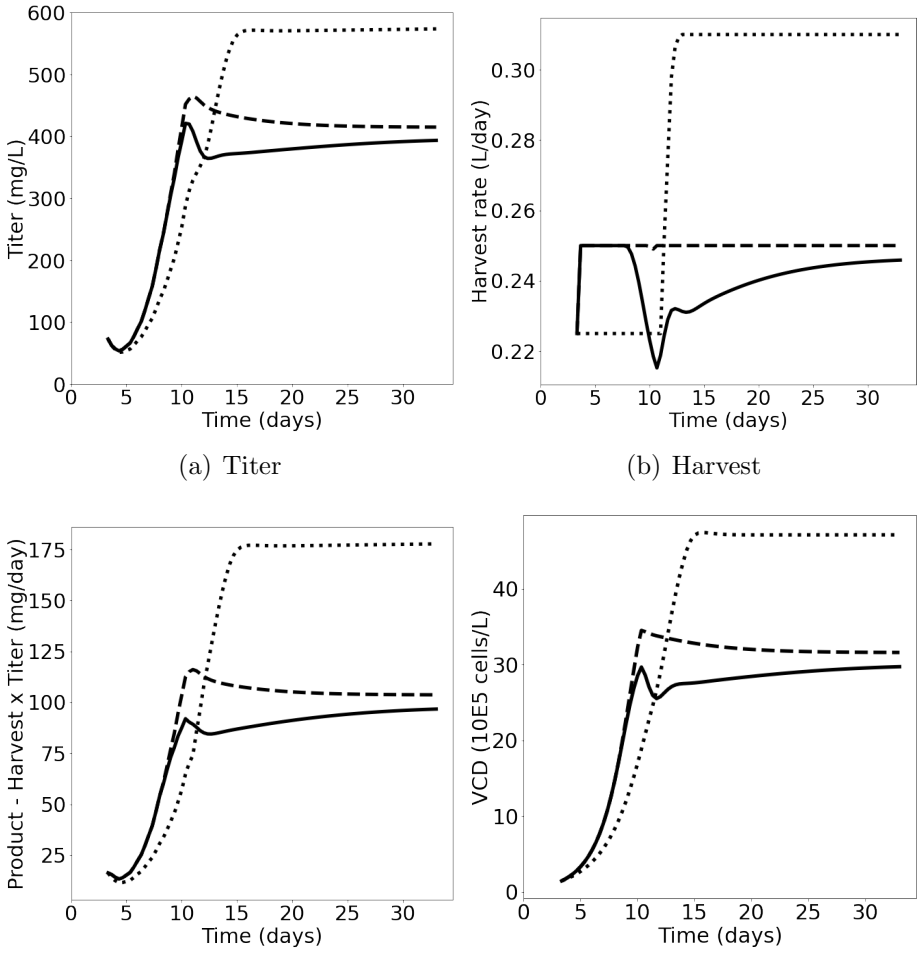
The first contribution of the work is to demonstrate the improved performance achievable using a data driven MPC implementation over the current industrial practice of PI control. In current practice, a PI control is used with a fixed VCD setpoint of

30 which it is able to achieve at the twenty five day mark as shown in Figure 7.4. Under the PI control, the bleed rate is initially kept low, and as the VCD starts to peak, the bleed rate is increased in order to hold a VCD setpoint of 30 as seen in Figure 7.5. As expected the other variables are held at their nominal values since only one PI controller is used which is linked to the bleed rate. This controller reaches a final product of 97 mg/day. In contrast, the implementation of the MPC results in a VCD over 47 but more importantly, results in the production of 178 mg/day of product as mentioned in Table 7.3. This is due to the controller's ability to shift all of the input variables while utilizing an appropriately identified process aware model. Figure 7.5 shows that increasing temperature, decreasing pH, increasing glucose feed concentration and decreasing bleed rate leads to optimal bioreactor operation and a clearly superior control strategy. Yet another benefit of the MPC implementation is the ability to simply ask it to maximize the product (through the objective function) instead of specifying a set-point.

In contrast, if the PI implementation was used to arbitrarily increase the PI setpoint to 60 (in an effort to achieve comparable product) the set-point is not met (see Figure 7.5). This of course is due to the limited control action available to the PI (the bleed rate), which it does push to zero. While the resulting final product is slightly higher than the original PI implementation at 104mg/day, the increase is marginal. Of course, in practice, the bleed rate would never be set to zero because without removing any waste from the bioreactor the build up would lead to increased cell death (thus under MPC implementation, the bleed rate is not allowed to go below 0.01 L/day as reported mentioned in Table 7.1).

Table 7.3: Constrained Subspace MPC vs PI control

Case)	Improvement (%)
Current PI	0 (current)
Higher VCD setpoint PI	7.2
Constrained subspace MPC	83.5



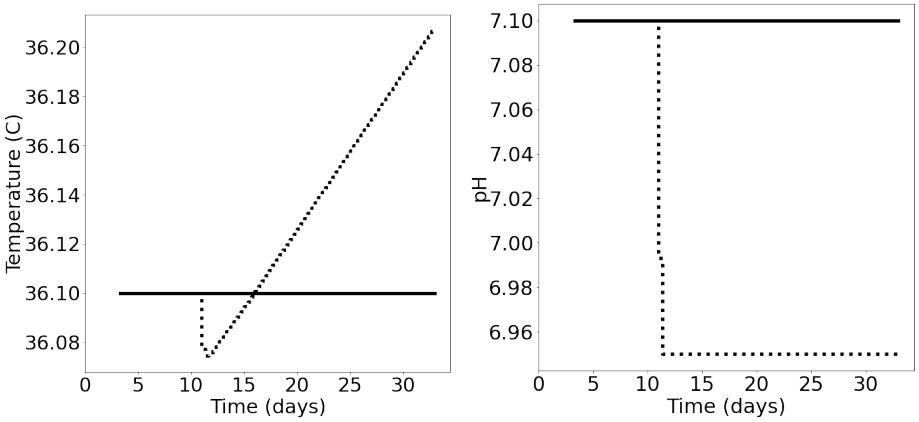
(a) Titer

(b) Harvest

(c) Product (Titer × Harvest)

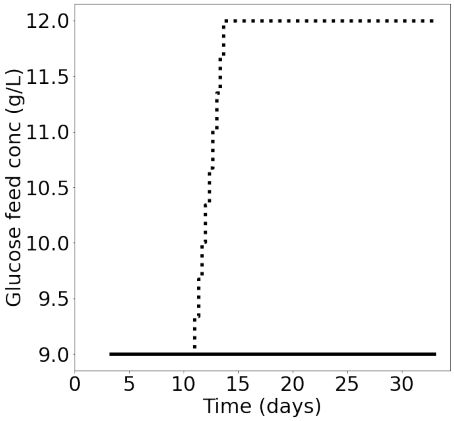
(d) Viable Cell Density (VCD)

Figure 7.4: Comparison of performance of the best MPC (dotted lines) with existing PI (solid lines) as well as PI with higher VCD setpoint (dahsed lines).

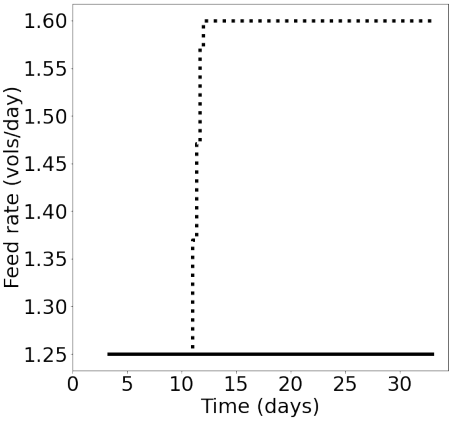


(a) Temperature

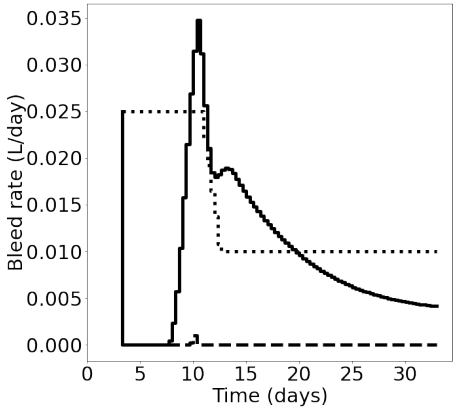
(b) pH



(c) Glucose feed concentration



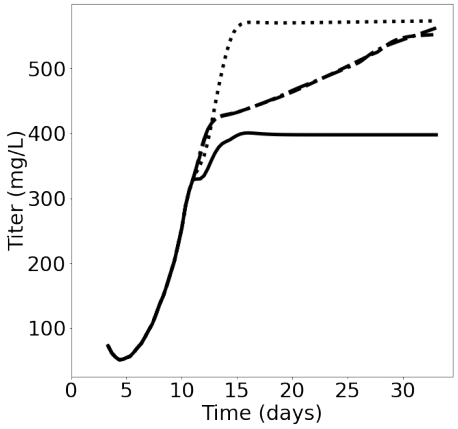
(d) Feed rate



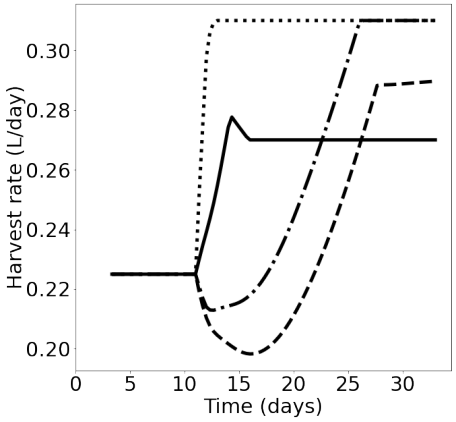
(e) Bleed rate

Figure 7.5: Comparison of inputs of the best MPC (dotted lines) with existing PI (solid lines) as well as PI with higher VCD setpoint (dashed lines).

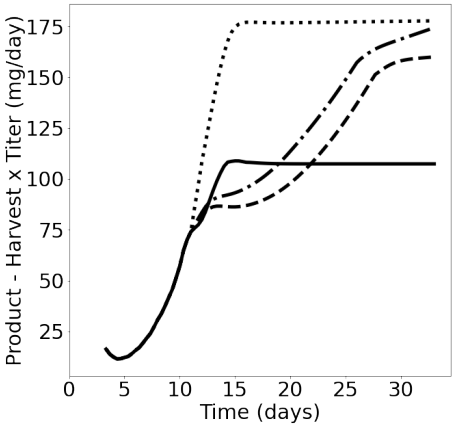
The second objective of this work is to demonstrate the necessity of a process aware constrained subspace identification technique when identifying the plant model. The key advantage in utilizing the constrained model is that a traditional unconstrained model may have incorrect steady state process gains. The process awareness comes from constrained subspace identification approach applied biologically relevant knowledge in the model identification stage by having the correct and relevant signs in the steady state gain between inputs and outputs as constraints during identification of the system matrices. Figure 7.6 clearly shows the advantage of utilizing the constrained subspace method compared to regular subspace identification (see Table 7.4). The MPC utilizing the **u**nconstrained subspace model with **s**hort **h**orizon is referred to as USH in the table for brevity. Similarly, ULH, CSH and CLH represent unconstrained long horizon, constrained short horizon and constrained long horizon respectively. Note that, in short horizon control these differences aren't as noticeable especially in the titer concentration as evident in Figure 7.6. However, when longer control horizons are utilized, the unconstrained model MPC performance deteriorates due to the effect of wrong gain signs identified, causing it to move inputs in a wrong direction. This difference is highlighted in Figure 7.7 where the unconstrained MPC fails to decrease the bleed rate as it has the wrong sign in the process gain. The longer horizon unconstrained subspace MPC thus performs very slightly better than existing PI control while the constrained model far outperforms both. Longer control horizons lead to improved performance with the process aware (constrained subspace) MPC as the controller is able to optimize the input trajectory over a longer period. Not only does the long horizon constrained subspace based MPC get the highest final product, it achieves the higher product and higher concentration both much earlier whereas the shorter horizon approaches only are able to reach values towards the end, leading to a significantly lower cumulative product compared to the longer horizon constrained MPC as shown in Table 7.5.



(a) Titer



(b) Harvest



(c) Product (Titer × Harvest)

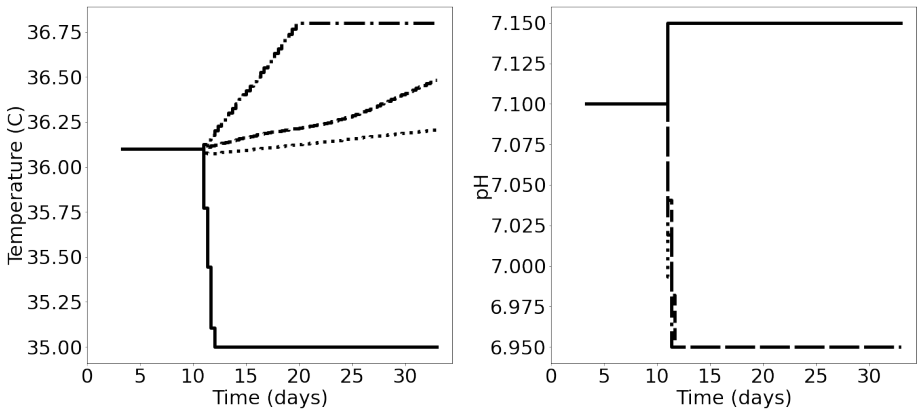
Figure 7.6: Comparison of performance of the best case i.e.longer horizon constrained subspace MPC (dotted) with MPCs based on shorter horizon constrained subspace (dash-dotted), longer horizon unconstrained i.e. regular subspace (solid) and shorter horizon unconstrained i.e. regular subspace (dashed).

Table 7.4: Unconstrained Subspace MPC vs Constrained Subspace MPC - Final Product

Case	Final Product (mg/day)	Improvement over current PI (%)	Improvement over USH MPC (%)
USH	160	65	0
ULH	107.4	10.7	-33
CSH	174	79.4	9
CLH	178	83.5	11.3

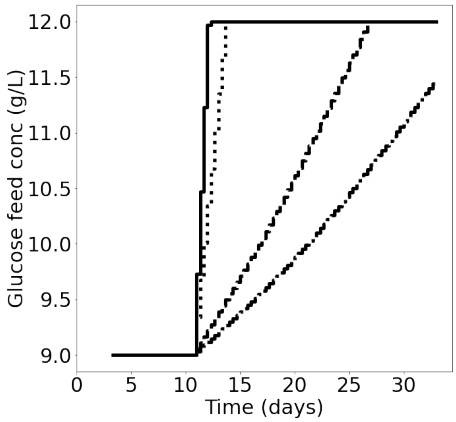
Table 7.5: Unconstrained Subspace MPC vs Constrained Subspace MPC - Average Product

Case	Average Product (mg/day)	Improvement over current PI (%)	Improvement over USH MPC (%)
USH	94	18.2	0
ULH	85.3	7.3	-9.3
CSH	103.2	29.8	9.8
CLH	132.8	67	41.3

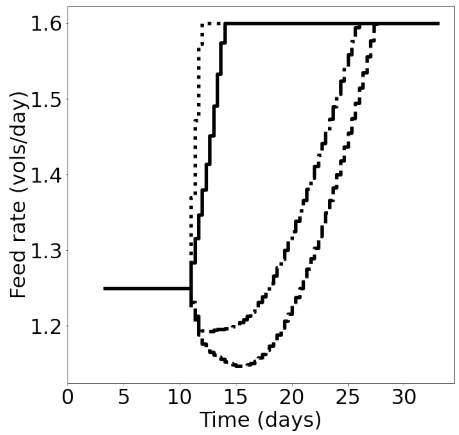


(a) Temperature

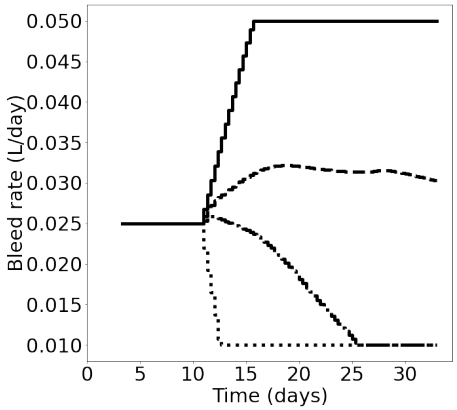
(b) pH



(c) Glucose feed concentration



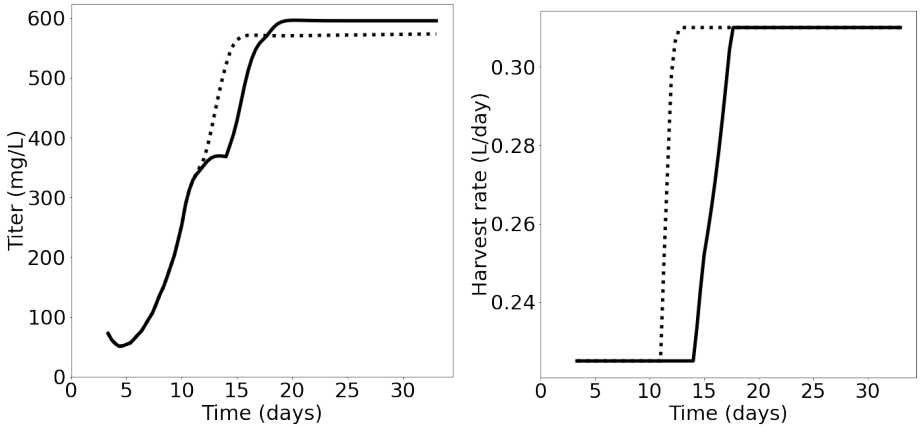
(d) Feed rate



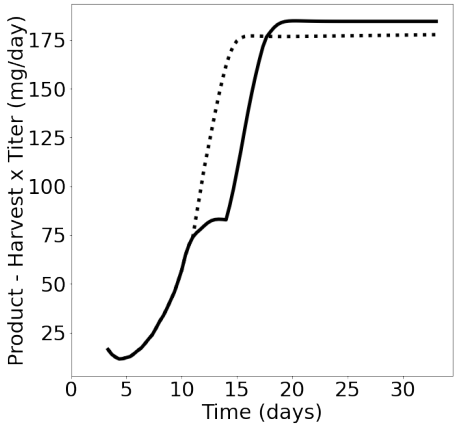
(e) Bleed rate

Figure 7.7: Comparison of inputs of the best case i.e.longer horizon constrained subspace MPC (dotted) with MPCs based on shorter horizon constrained subspace (dash-dotted), longer horizon unconstrained i.e. regular subspace (solid) and shorter horizon unconstrained i.e. regular subspace (dashed).

The final contribution of this work is to show the robustness of model predictive control to maximize the final product. In order to test the robustness of the controller the MPC using the constrained subspace model is compared against a new bioreactor process. In the new bioreactor the parameters such as growth rates and death rates are now different than the training data used to identify the constrained subspace model. This creates additional plant model mismatch and represents scenarios where the reactor may be processing new batches of cells. Figure 7.8 shows how the constrained MPC is able achieve a similar final product. When comparing the input changes made by the MPC, Figure 7.9 shows that the constrained MPC makes similar input moves in both the constrained model built on new or current data as well as a constrained model which was built on data from an older system. The MPC utilizing the constrained subspace model built on old plant data is also able to achieve a high final product though at a cost of higher temperatures which is not very desirable. But overall, the control performance remains acceptable when using the constrained subspace MPC on a different system demonstrating robustness

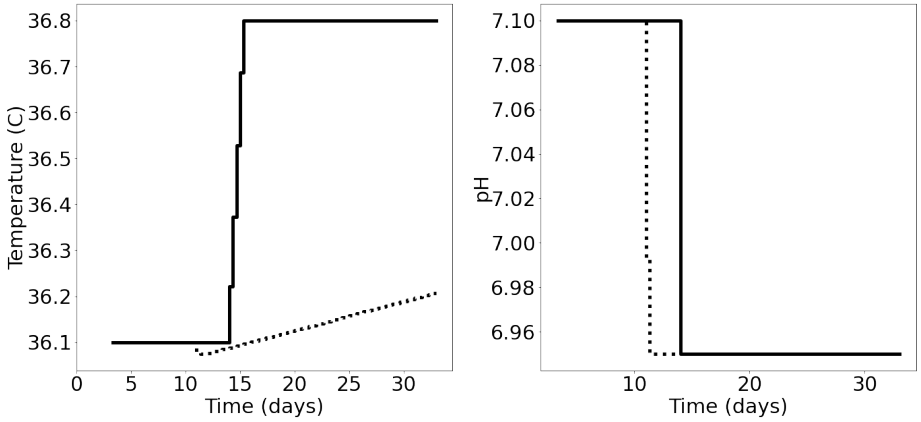


(a) Titer (b) Harvest



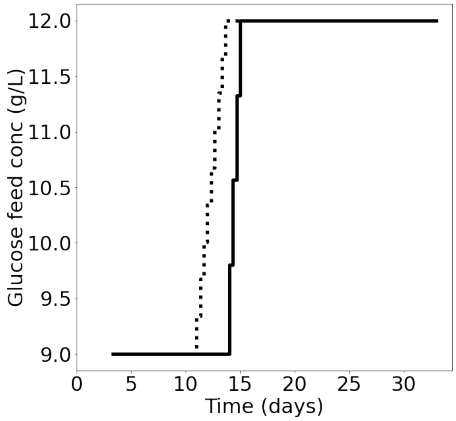
(c) Product (Titer × Harvest)

Figure 7.8: Comparison of performance of the constrained subspace MPC trained on old model plant system (solid) with performance of constrained subspace MPC trained on current model plant system (dotted) to demonstrate robustness.

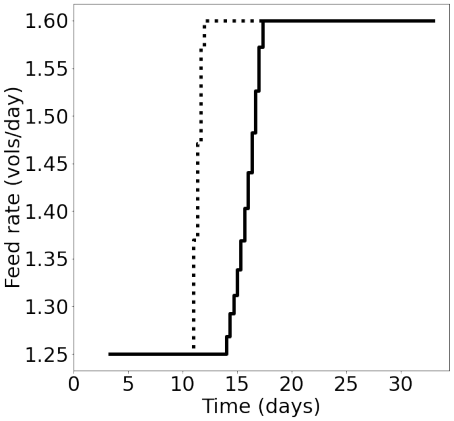


(a) Temperature

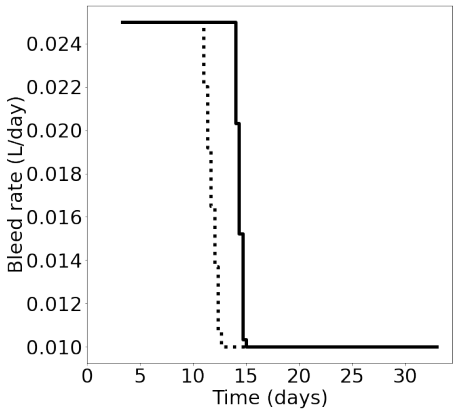
(b) pH



(c) Glucose feed concentration



(d) Feed rate



(e) Bleed rate

Figure 7.9: Comparison of inputs of the constrained subspace MPC trained on old model plant system (solid) with performance of constrained subspace MPC trained on current model plant system (dotted).

7.6 Conclusions

The present manuscript demonstrated the possibility of using a process aware data driven model predictive control scheme for bioreactors to enable performance improvement compared to industry standard of proportional-integral controller schemes. The importance of using a process aware model within model predictive control schemes was illustrated by comparing subspace model with process knowledge based constraints to standard subspace model based MPC implementation. Finally, the ability of the MPC to handle process changes was illustrated, with the MPC performance continuing to be acceptable under process changes.

7.7 Acknowledgment

Financial support from the McMaster Advanced Control Consortium is gratefully acknowledged.

Bibliography

[1] Bernard, O., Mairet, F., and Chachuat, B. (2015). Modelling of microalgae culture systems with applications to control and optimization. In *Microalgae Biotechnology*, pages 59–87. Springer.

[2] Caramihai, M. and Severin, I. (2013). Bioprocess modeling and control. *Biomass Now: Sustainable Growth and Use*, page 147.

[3] Chang, L., Liu, X., and Henson, M. A. (2016). Nonlinear model predictive control of fed-batch fermentations using dynamic flux balance models. *Journal of Process Control*, 42:137–149.

[4] Chusainow, J., Yang, Y. S., Yeo, J. H., Toh, P. C., Asvadi, P., Wong, N. S., and Yap, M. G. (2009). A study of monoclonal antibody-producing cho cell lines: What makes a stable high producer? *Biotechnology and bioengineering*, 102(4):1182–1196.

[5] Corbett, B. and Mhaskar, P. (2016). Subspace identification for data-driven modeling and quality control of batch processes. *AIChE Journal*, 62(5):1581–1601.

[6] Del Rio-Chanona, E. A., Cong, X., Bradford, E., Zhang, D., and Jing, K. (2019). Review of advanced physical and data-driven models for dynamic bioprocess simulation: Case study of algae–bacteria consortium wastewater treatment. *Biotechnology and bioengineering*, 116(2):342–353.

[7] Deschenes, J.-S., Desbiens, A., Perrier, M., and Kamen, A. (2006). Multivariable nonlinear control of biomass and metabolite concentrations in a high-cell-density perfusion bioreactor. *Industrial & engineering chemistry research*, 45(26):8985–8997.

[8] Dochain, D. and Perrier, M. (1997). Dynamical modelling, analysis, monitor-

ing and control design for nonlinear bioprocesses. In *Biotreatment, Downstream Processing and Modelling*, pages 147–197. Springer.

[9] Ferkl, L. and Širokỳ, J. (2010). Ceiling radiant cooling: Comparison of armax and subspace identification modelling methods. *Building and Environment*, 45(1):205–212.

[10] Garthwaite, P. H. (1994). An interpretation of partial least squares. *Journal of the American Statistical Association*, 89(425):122–127.

[11] Geladi, P. and Kowalski, B. R. (1986). Partial least-squares regression: a tutorial. *Analytica chimica acta*, 185:1–17.

[12] Gevers, M. (2005). Identification for control: From the early achievements to the revival of experiment design. *European journal of control*, 11(4-5):335–352.

[13] Ghosh, D., Hermonat, E., Mhaskar, P., Snowling, S., and Goel, R. (2019). Hybrid modeling approach integrating first-principles models with subspace identification. *Industrial & Engineering Chemistry Research*, 58(30):13533–13543.

[14] Karra, S., Sager, B., and Karim, M. N. (2010). Multi-scale modeling of heterogeneities in mammalian cell culture processes. *Industrial & Engineering Chemistry Research*, 49(17):7990–8006.

[15] Lee, J. H. (2011). Model predictive control: Review of the three decades of development. *International Journal of Control, Automation and Systems*, 9(3):415–424.

[16] Leib, T. M., Pereira, C. J., and Villadsen, J. (2001). Bioreactors: a chemical engineering perspective. *Chemical engineering science*, 56(19):5485–5497.

[17] Mairet, F., Bernard, O., Cameron, E., Ras, M., Lardon, L., Steyer, J.-P., and Chachuat, B. (2012). Three-reaction model for the anaerobic digestion of microalgae. *Biotechnology and Bioengineering*, 109(2):415–425.

[18] Moonen, M., De Moor, B., Vandenberghe, L., and Vandewalle, J. (1989). On-and off-line identification of linear state-space models. *International Journal of Control*, 49(1):219–232.

[19] Morari, M. and Lee, J. H. (1999). Model predictive control: past, present and future. *Computers & Chemical Engineering*, 23(4-5):667–682.

[20] Morel, E., Tartakovsky, B., Guiot, S., and Perrier, M. (2006). Design of a multi-model observer-based estimator for anaerobic reactor monitoring. *Computers & chemical engineering*, 31(2):78–85.

[21] Patel, N., Corbett, B., and Mhaskar, P. (2021). Model predictive control using subspace model identification. *Computers & Chemical Engineering*, 149:107276.

[22] Patel, N., Corbett, B., Trygg, J., McCready, C., and Mhaskar, P. (2020a). Subspace based model identification for an industrial bioreactor: Handling infrequent sampling using missing data algorithms. *Processes*, 8(12):1686.

[23] Patel, N., Nease, J., Aumi, S., Ewaschuk, C., Luo, J., and Mhaskar, P. (2020b). Integrating data-driven modeling with first-principles knowledge. *Industrial & Engineering Chemistry Research*, 59(11):5103–5113.

[24] Pörtner, R., Barradas, O. P., Frahm, B., and Hass, V. C. (2017). Advanced process and control strategies for bioreactors. In *Current Developments in Biotechnology and Bioengineering*, pages 463–493. Elsevier.

[25] Seborg, D. E., Mellichamp, D. A., Edgar, T. F., and Doyle III, F. J. (2010). *Process dynamics and control*. John Wiley & Sons.

[26] Simutis, R. and Lübbert, A. (2015). Bioreactor control improves bioprocess performance. *Biotechnology journal*, 10(8):1115–1130.

[27] Sirois, J., Perrier, M., and Archambault, J. (2000). Development of a two-step segregated model for the optimization of plant cell growth. *Control Engineering Practice*, 8(7):813–820.

[28] Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., et al. (2020). Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272.

[29] Xie, L. and Wang, D. I. (1996). High cell density and high monoclonal antibody production through medium design and rational control in a bioreactor. *Biotechnology and bioengineering*, 51(6):725–729.

[30] Zhang, D., Del Rio-Chanona, E. A., Petsagkourakis, P., and Wagner, J. (2019). Hybrid physics-based and data-driven modeling for bioprocess online simulation and optimization. *Biotechnology and bioengineering*, 116(11):2919–2930.

Chapter 8

Conclusions and Recommendations

This thesis focuses on the data-driven modeling and control of both batch and continuous processes using subspace methods. However, there are limitations to this work which are listed in remarks in each paper. One of main concerns is that this work identifies a subspace model which is a linear time invariant model. While it is possible to develop time variant models, these result in a huge optimization problem as the initial state becomes a decision variable. The first approach in Chapter 2, applied first-principles constraints to batch subspace identification techniques. As data-driven models rely solely on the mathematical correlation between the inputs and outputs they often end up identifying incorrect process trends to minimize the prediction error in the training data. The constrained subspace approach was developed in order to identify a subspace model that would always have the correct process information such as the correct sign on input/output gains. This approach was tested using a simulation example that showed the constrained model was better at predicting the process online. Chapter 3 continued to utilize this model to develop a state space model predictive controller that could handle a feedthrough matrix. Industrial MPC algorithms tend to ignore the feedthrough term when computing control action as the dynamics do not justify its use. The state space MPC in Chapter 3 uses a simple quadratic program structure and includes the feedthrough term leading to improved controller performance. The constrained model is able to reach the steady state at a faster and more efficient rate that the traditional subspace models without feedthrough terms. Chapter 4 discusses another novel contribution in the form of a regression-based subspace identification approach. The missing data problem is a big part of industrial data analysis since it is very difficult to have full data observations due to different sampling rates and equipment downtime. Traditional subspace relies on singular value decomposition and other matrix manipulations that require the matrices to be full rank. In the proposed algorithm the same subspace techniques are carried out using a series of equivalent regressions using the NIPALS algorithm for both PCA and PLS. The proposed algorithm was then tested using a polymethyl

methacrylate simulation and compared against both linear interpolation and mean replacement techniques. The proposed subspace model had the best performance in comparison to both interpolation approaches. The missing data in this context must be clarified to be present in the outputs where the inputs are available. Moreover the amount of missing data must still allow for sufficient excitation in order for trends to be determined. Another problem that benefits from this missing data approach is that of batch quality control. In order to measure the quality of the batch additional tests must be conducted meaning that quality measurements are not available at the same frequency as traditional outputs. To that end, it is important to build a model that can utilize both the quality variables and process outputs together. These quality variables often have more than 90% missing data making interpolation techniques unreliable and data subspace methods impossible. Chapter 5 takes the missing data algorithm presented in Chapter 4 and solves the quality problem for the polymethyl methacrylate process. Using the algorithm to predict the quality and output variables together resulted in a more accurate process model in comparison to interpolation and using individual models. Chapter 6 demonstrates a practical application of the missing data problem to model the Sartorius bioreactor. The bioreactor was initially modeled by a first-principles based model which was difficult to maintain as there were many differential equations with numerous parameters that were estimated. Additionally, this bioreactor has a unique infrequent sampling situation in that the key input glucose is a discrete input. To handle discrete inputs in a continuous process it was necessary to develop an update procedure to have the increased glucose measurement at the correct time interval. Using the missing data algorithm, a subspace model of the system was identified that was more accurate in comparison to the existing first principles model. Finally, in Chapter 7, a complete application of the developed modeling and control algorithm is used to model a continuous bioreactor. First-principles gain constraints were used at the model identification stage and the missing data algorithm was also utilized. Finally using the new state space MPC the

identified model was successfully able to model and control the bioreactor to increase the batch production and quality.

8.1 Future Work

This section presents recommendations for future areas of exploration based on the work defined in this thesis. Some of these areas are currently being explored by other graduate students under my supervision in the Mhaskar research group while others will be explored with future students. The first area to be investigated is based on the work done using the missing data algorithm for bioreactor modeling. Startup and shutdown procedures are a difficult part of modeling and control due to the varying dynamics and are often ignored. Currently the industrial bioreactors utilized in this thesis have been modeled after a three day initialization period has passed. When utilizing the model online the three day startup allows enough time for the states to converge making the model accurate immediately after this period ends. Building a subspace model capable of controlling the bioreactor during this initial startup period would lead to additional savings and potential terminal quality improvements. The second area of research is to work on handling batch process shifts resulting from differing batch conditions and material standards. The subspace models that were generated to include first-principles knowledge are expected to be more robust as they avoid overfitting incorrect trends. However, as time passes the batches will deviate further from the training data and even with the process knowledge retraining will be necessary. An approach to update the process model using new batches can be developed using the missing data algorithm as only certain variables might need to be accounted for. Finally, the aim of this thesis was to utilize subspace based methods to develop process models and use them in control strategies. Moving forward, these same techniques of missing data and first-principles constraints can be applied to

other types of data driven models. Specifically, neural network based model benefit from linear regression techniques like PLS and PCA making them a good candidate to expand this work.