Hybrid and data based modeling and control solutions

Hybrid and data-driven modeling and control approaches to batch and continuous processes

by

Debanjan Ghosh

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

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

TITLE:	Hybrid and data-driven modeling and control approaches to batch and continuous processes
AUTHOR:	Debanjan Ghosh (McMaster University, Hamilton, ON)
SUPERVISOR:	Dr. Prashant Mhaskar
NUMBER OF PAGES:	xvii, 175

ABSTRACT

The focus of this thesis is on building models by utilizing process information: from data, from our knowledge of physics, or both. The closer the model approximates reality, the better is the expected performance in forecasting, soft-sensing, process monitoring, optimization and advanced process control. In the domain of batch and continuous manufacturing, quality models can help in ensuring tightly controlled product quality, having safe and reliable operating conditions and reducing production/-operation costs.

To this end, first a parallel grey box model was built which makes use of a mechanistic model, and a subspace identification model for modeling a batch poly methyl methacrylate (PMMA) polymerisation process. The efficacy of such a parallel hybrid model in the context of a control problem was illustrated thereafter for reducing the volume of fines. Real-time implementation of models in many cases demand the model to be tractable and simple enough, and thus the parallel hybrid model was next adapted to have a linear representation, and then used for control computations. While the parallel hybrid modelling strategy shows great advantages in many applications, there can be other avenues of using fundamental process knowledge in conjunction with historical data. In one such approach, a unique way of adding mechanistic knowledge to improve the estimation ability of PLS models was proposed. The predictor matrix of PLS was augmented with additional trajectory information coming strategically from a mechanistic model. This augmented model was used as a soft-sensor to estimate batch end quality for a seeded batch crystallizer process. In a collaborative work with an industrial partner focusing on estimating important variables of a hydroprocessing unit, an operational data based input-output model was chosen as the right fit in the absence of available mechanistic knowledge. The usefulness of linear dynamic modeling tools for such applications was demonstrated.

ACKNOWLEDGEMENTS

I would like to begin by expressing my utmost gratitude to my supervisor Dr. Prashant Mhaskar. He has been a constant source of inspiration for me and possibly can't thank him enough for the support he has provided. His guidance not only as an academic supervisor but also on a personal level has helped me stay motivated and feel secure since the day I landed in Canada, and started my journey as an international student. His technical expertise and ability to breakdown complex ideas to simplistic terms has helped me gain clarity in my thoughts. His cheerful, enthusiastic and supporting nature has helped me stay mentally positive even during the most difficult of times. I have always felt an immensely close connection with him, and his humility and modesty has inspired me to lead life along a similar path.

Next, I would like to express my sincere gratitude to Dr. Thomas Adams and Dr. Thia Kirubarajan for accepting to be my supervisory committee members and for all the constructive feedback, discussions, thought provoking questions leading to a lot of interesting investigations in my research. Their advice and suggestions have been critical to the successful completion of my studies. I would also like to thank Dr. Brandon Corbett for all the interesting and insightful conversations, and laying the foundation of a lot of algorithmic work that I did in my thesis. I am thankful to Dr. John MacGregor and Dr. Jesus Moreira for their expert guidance in the collaborative work on a number of projects. I would like to thank Michelle Whalen, Kristina Trollip and Ellis Linda for all the administrative help during my studies.

I feel blessed to be a part of McMaster Advanced Control Consortium (MACC), and this multicultural group has been instrumental in a number of ways in developing my career. I consider myself lucky to have met the finest of people in this group who constantly inspired me and with whom I had the privilege of participating in a lot of insightful conversations. I would like to thank my friends Hesam Hassanpour, Carlos Rodriguez Sotelo, Daniela Derring De Silva, Mina Naieni, Pranav Madabhushi, Anthony Quarshie, Mahir Jalanko, Subhajit Mishra, Ali Torabi, Satyam Agnihotri and a lot of others with whom I created memories that I will cherish forever. Their presence and positive influence in my life have made my stay here pleasant and extremely enjoyable.

Finally, I would like to thank my family for believing in me and supporting me in my decisions. Their unconditional love, moral support and endless encouragement has helped me overcome the difficult of times and gave me the strength to push forward.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, the main theme and overall objectives of the thesis are presented. The motivation behind developing data and hybrid modeling and control schemes for Chemical Engineering processes are first described in 1.1 followed by the outline and objectives of the thesis chapters in 1.2.

1.1 Motivation

Models undoubtedly play a key role in modern day industrial manufacturing. The ability of models to successfully approximate the process behaviour is of paramount importance and improving their quality has been a direction of continual research for many years. In the process engineering domain, incorporation of models have lead to successful design, optimization, monitoring and control of various complex continuous and batch processes.

The traditional style of model building over many decades has been the use of physics based governing laws commonly known as first principles modeling or mechanistic modeling [3]. These models are typically quite rigorous and can approximate the process relatively well. However, the task of building such high fidelity models is not trivial and requires good knowledge and expertise in that field. They also find limited use in many real time implementations owing to the associated complexity in using them in many algorithms. The use of simplifying assumptions to obviate such complexities often lead to biases in model predictions. On the other end of the spectrum, we have models built from purely log data of historical operation, known as data based or black box models [1, 6]. These techniques have come into prominence owing to the increasing digitalization of industries, mass storage capabilities of data, and the increased computing power in the current era. A variety of such models exist in literature and is chosen depending on their suitability to a particular application. Their structure is first chosen by the practitioner and the parameters are then identified from the database. These models are much more tractable compared to mechanistic models but suffer from lack of generalizability to different operating conditions.

Hybrid [4, 5] or grey box modeling (Figure 1.1) schemes offer unique ways of synergizing mechanistic knowledge with purely data based models. These models provide avenues to better utilize all the available process knowledge along with measurement data, and try to alleviate certain shortcomings associated with the earlier described modeling strategies. Several architectures in this category have been mentioned in literature, among which the series and parallel schemes are the most prominent. These arrangements provide solutions to correct the biases in the mechanistic models, and at the same time provide easier and tractable alternatives to the problem of model maintenance (an important issue encountered in practice). One of the most important advantage that a hybrid model presents over a purely data-driven model is the need of less data for identification. In many critical processes, manipulating all the inputs for exciting the system may be quite expensive and time consuming. Since many relationships between inputs and outputs are already embedded in the hybrid model from the fundamental understanding of the process, these models in principle require less data in their development.

Apart from the above mentioned strategies of hybrid modeling, there exist other ways of incorporating fundamental knowledge into data based models where the parameters of the black box model are identified subject to physical constraints [2] and thus, leading to better predictive data-based models. However, implementation of grey box modeling techniques for predicting the behaviour of complex processes, along with their usage in soft sensing, monitoring and advanced process control has been quite limited.

In the past, researchers have made some attempts in this direction using non-linear tools like Artificial Neural Networks (ANN) as the data-driven component in grey

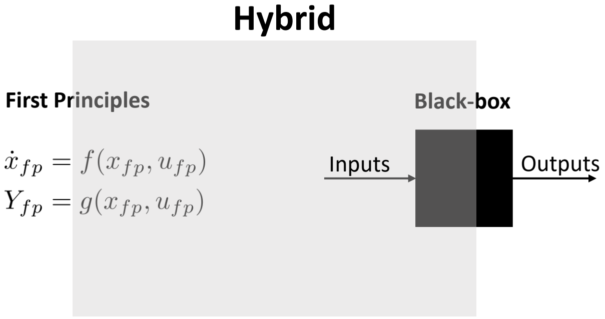

Hybrid
First Principles
$\dot{x}_{fp} = f(x_{fp}, u_{fp})$
$Y_{fp} = g(x_{fp}, u_{fp})$
Black-box
Inputs
Outputs

Figure 1.1: Generic hybrid architecture

box modeling. However, such techniques often suffer from the curse of overfitting, and their usage is limited when historical data is quite scarce owing to the expensive nature of the experiments/runs, mostly for batch processes. Statistical models like Partial Least Squares (PLS) and linear algebra based techniques like Subspace Identification offer attractive solutions in such situations. Their utility as a stand alone data based model in describing continuous, batch and batch like processes along with their efficacy in Model Predictive Control (MPC) schemes have been demonstrated rigorously in the past. However, utilization of such models in a grey box architecture has not been explored before. In this thesis, the primarily focus is on building hybrid and data-based (when mechanistic models are not available) approaches using subspace identification and PLS models as the main tool for the data-driven component. These model architectures have been used in a wide range of applications including dynamic prediction, advanced process control, soft sensing of quality variables etc. to demonstrate their efficacy, and the potential value added in the area of process systems engineering.

1.2 Outline of the thesis

Motivated by the challenges associated with the domains of mechanistic and data-driven techniques, this thesis focuses on building hybrid models with the help of linear tools (PLS and Subspace Identification) as the data-driven component, and demonstrates its usefulness for a variety of applications including prediction, control and inferential sensing of batch processes like PMMA polymerisation and seeded batch crystallization. In the case where first principles models was not readily available for industrial hydroprocessing units, the efficacy of linear dynamic modeling tools was demonstrated for building input-output models using real industrial data. The rest of the thesis is organized as follows.

In Chapter 2, the problem of building a parallel hybrid model utilizing a mechanistic model and subspace identification model for a batch PMMA polymerization process is considered. The development of this hybrid architecture is first described and its efficacy in predicting the batch dynamics is demonstrated against the mechanistic model and a purely data-based subspace model.

Chapter 3 considers the problem of implementing a parallel hybrid architecture in a MPC scheme. The control objective considered is reducing the volume of fines generated due to nucleation in a seeded batch crystallization process. One of the main objective of this study is to maintain the linearity of the model to be used inside the MPC, and in that direction, first a linear equivalent of the non-linear parallel hybrid architecture is built and then a MPC formulation embedding the model is described. Key performance indicators demonstrated the superior performance of the proposed MPC over a data-based MPC for higher quality constraints.

In Chapter 4, a simple and unique way of incorporating fundamental knowledge to improve the predictive ability of PLS models is demonstrated. The general idea behind the development of this hybrid approach is first presented , followed by implementation of this approach as a soft sensing tool in estimating the final quality for seeded batch crystallizer is demonstrated. The superior estimation ability of this model compared to an only data based PLS model is shown.

In Chapter 5, the usefulness of data-driven dynamic modeling techniques like Dynamic Partial Least Squares (DPLS) and Subspace Identification for building input-output models for industrial hydroprocessing units is presented. Challenges involved in modeling using a real data set like handling of missing data in both inputs and outputs are addressed for subspace identification. The results demonstrate the merits of using such linear dynamic methods for modeling these processes, and suggests practitioners to consider these tools in their applications.

Finally in Chapter 6, the key findings, concluding remarks along with directions of future work are presented.

Bibliography

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

[2] 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.

[3] Pinto, J. and Ray, W. (1995). The dynamic behavior of continuous solution polymerization reactors—vii. experimental study of a copolymerization reactor. *Chemical Engineering Science*, 50(4):715 – 736.

[4] Psichogios, D. C. and Ungar, L. H. (1992). A hybrid neural network-first principles approach to process modeling. *AIChE Journal*, 38(10):1499–1511.

[5] Su, H.-T., Bhat, N., Minderman, P., and McAvoy, T. (1992). Integrating neural networks with first principles models for dynamic modeling. *IFAC Proceedings Volumes*, 25(5):327 – 332. 3rd IFAC Symposium on Dynamics and Control of

Chemical Reactors, Distillation Columns and Batch Processes (DYCORD+ '92), Maryland, USA, 26-29 April.

[6] Wold, S., Geladi, P., Esbensen, K., and Öhman, J. (1987). Multi way principal components and pls analysis. *Journal of Chemometrics*, 1(1):41–56.

Chapter 2

A hybrid modeling approach integrating first principles models with subspace identification

The contents of this chapter have been published in *Industrial & Engineering Chemistry Research* Journal.

- Hybrid Modeling Approach Integrating First-Principles Models with Subspace Identification. Debanjan Ghosh, Emma Hermonat, Prashant Mhaskar, Spencer Snowling, and Rajeev Goel *Industrial & Engineering Chemistry Research* 2019, 58 (30), 13533-13543 DOI: 10.1021/acs.iecr.9b00900

Abstract

This paper addresses the problem of synergizing first principles models with data-driven models. This is achieved by building a hybrid model where the subspace model identification algorithm is used to create a model for the residuals (mismatch in the outputs generated by the first principles model and the plant output) rather than being used to create a dynamic model for the process outputs. A continuous stirred tank reactor (CSTR) setup is used to illustrate the proposed approach on a continuous system. To further evaluate its efficacy, the proposed methodology is applied on a batch polymethyl methacrylate (PMMA) polymerization reactor and the predictions are compared with that of first principles modeling and the data-driven approach alone. The paper demonstrates the improved modeling capability of the hybrid model over either of its components.

2.1 Introduction

Good models are integral to process systems engineering, often being the core around which control and optimization formulations are designed. First principles or mechanistic modeling refers to the type of modeling where explicit knowledge of the process mechanism is present and utilized. Thus, these models invoke fundamental physical and chemical laws that describe the system being considered and utilize algebraic, ordinary or partial differential equations. Often times, the fewer the assumptions made, the higher the number of model parameters and the more complex the model structure. The task of building and maintaining a first principles model [2, 27, 17, 33, 12, 22, 28] continues to involve significant effort.

Increased availability of data, along with improved computational capabilities have made the so-called data-driven methods increasingly attractive [30, 14, 4]. In this approach, a model structure is chosen a priori and parameters are identified using the available data. In this direction, several statistical based approaches exist and are distinguished by the kinds of model structures being utilized, including latent variable-based methods and subspace identification-based algorithms.

Projection to latent structures (PLS) is one of the most recognized statistical modeling methodologies and has shown significant advantages over its counterparts. PLS is one of the latent variable methods where measurement data of high dimension is projected into lower dimensional space to create simpler and effective models. Their usefulness to batch processes have been extensively demonstrated, [14, 13] with special algorithms utilized to enable handling of batches of non-uniform lengths. In this method, while process knowledge is often integrated into the modeling approach by utilizing additional calculated variables, a direct integration of first principles models and latent variables models has not been done.

Another statistical-based modeling approach is subspace identification, which enables the identification of models having state-space representations. In these methods, a state trajectory is first identified and the model parameters (for a linear time-invariant model (LTI)) are then determined (see [30, 26, 39, 18, 10] for examples of subspace identification methods). Recently, subspace identification-based methods for modeling batch processes [6, 7, 16, 15] have been proposed. A subspace identification scheme where the batch lengths do not necessarily have to be of the same size was proposed in [6] and its merits in prediction and control were demonstrated by applying it to a batch polymerization process. In addition, a modeling strategy was proposed in [7] that allows discrete addition during the batch. The usefulness of this identification scheme was shown in [16, 15] by applying it to a batch particulate process and a hydrogen plant start-up process.

The approaches described above, however, have not demonstrated the utilization of available process knowledge in synergy with the data-driven modeling approach.

There have been some examples in the past of approaches that have utilized a priori knowledge in combination with artificial neural networks (ANNs; see [32, 1, 19, 20]) to create hybrid, or grey-box, models. Readers are encouraged to go through [38], where an excellent overview of different hybrid semi-parametric modeling approaches is provided. The various applications of these models, such as process monitoring, optimization and control in the field of process systems engineering are also discussed. It should be noted that in this work, hybrid semi-parametric models are referred to as simply hybrid models. Hybrid modeling approaches can be broadly categorized as either series (where the models are used one after the other) or parallel (where the models are used in parallel to one another). Examples of the series approach are provided in [36, 29, 17] where the hybrid model is built by first by developing the first principles model and neural networks are subsequently used to determine the parameters. The approach was later demonstrated in [17] with uncertain, unmeasurable and unknown parameters.

In the parallel approach, a data-driven modeling structure is utilized to model the error between process outputs and the predicted outputs from the first principles model (referred to as residuals). Typically, a neural network model is utilized as the data-driven modeling scheme [34, 40]; however, the utility of ANN-based models to capture the process dynamics while avoiding the problem of overfitting continues to be an active research focus. On the other hand, statistical approaches such as subspace identification-based models permit a more direct handling of the overfitting problem. Therefore, the proposed approach follows a parallel hybrid modeling scheme and uses subspace identification as the data-driven component.

A key advantage to the hybrid modeling approach is that it offers an alternative approach to addressing the issue of model maintenance that is associated with using first principles models. As process conditions change and new data becomes available, a recalibration of the first principles model (re-identifying the model parameters from process data) will likely result in a more informative model. As stated earlier, the recalibration process is limited by the difficulties associated with solving a highly complex optimization problem. The hybrid modeling approach instead exploits the fact that even though the first principles model may no longer be able to predict the variables precisely, it still accounts for the qualitative nature of the (often nonlinear) process dynamics. Therefore, building a simpler data-driven model that captures the residuals negates the need for model recalibration or re-identification given that the hybrid model accounts for both the process nonlinearity, and the effect of the new process conditions.

In summary, while the ability of the subspace algorithm for modeling batch processes has been established [7, 6, 16], this effective modeling tool has not yet been integrated with first principles model in a hybrid modeling framework. Motivated by these considerations, a hybrid model integrating first principles modeling with subspace identification algorithm is proposed in this work. The rest of manuscript is organized as follows: Section 2.2.1 presents a PMMA batch process as a motivating example and a brief overview of the subspace identification method employed is given in Section 5.2.4. Section 2.3 describes the development of the proposed methodology. The hybrid approach is first illustrated using a CSTR case study in Section 2.3.4. The methodology is also applied to a PMMA polymerization process and compared with the existing subspace identification approach in Section 2.4. Finally, concluding remarks are presented in Section 2.5.

2.2 Preliminaries

In this section, we briefly review a PMMA polymerization process example to motivate the proposed research. Given that the model in the proposed methodology considers subspace methods as the identification tool, a brief review of subspace identification is presented in the following subsection.

2.2.1 Motivating example: PMMA polymerization reactor

A batch polymerization of PMMA in a stirred tank reactor is used as the motivating example in this study. The ingredients of this recipe are the monomer methyl methacrylate, the initiator AIBN and toluene as solvent. The reactor temperature is maintained using a cooling/heating jacket. To simulate the process, the mechanistic model presented in [11], with alterations taken from [12, 31] is utilized (see [5] for further details). A total of nine states are present in this mechanistic model, which includes the concentration of monomer, the concentration of initiator, the reaction temperature and six moments of living and dead polymer chains. The model with the associated parameters is considered as the 'process' in the present manuscript.

To replicate practice, data is generated for the batches using variation in the initial conditions. The measured outputs for this polymerization process are chosen to be the reaction temperature, log viscosity, and density. The manipulated input variable is the jacket temperature (T_{jk}). The process dynamics display significant nonlinearity and are highly dependent on initial conditions, making it important for the hybrid modeling approach to be implemented in a way that estimates current process states before being used for prediction purposes.

For the purpose of model identification, it is thus assumed that a total of N_T training batches and N_V validation batches are available with $D_b, b = 1 \ldots, N_T$ and $D_v, v = 1, \ldots, N_V$ being the duration of training and validation batches respectively. Note that the training and validation batches are not required to be of uniform length. The input (jacket temperature) is denoted by $u = T_{jk}$ and the measured outputs (reaction temperature, log viscosity, and density) are represented by the column vector $y = \begin{bmatrix} T_{reac} & \log \mu & \rho \end{bmatrix}^T$. It is assumed that a first principles model for the process is available, albeit with a mismatch between the process and model. Note that using existing results, a subspace model between the input and outputs can be readily determined. This paper specifically addresses the problem of developing a hybrid model that combines the first principles and data-driven models. Before proceeding to present the proposed approach, the subspace identification method is briefly reviewed.

2.2.2 Subspace identification

Subspace identification techniques are methods which use matrix decomposition concepts of linear algebra such as SVD (singular value decomposition) and QR factorization for the identification of a discrete time linear time-invariant state-space model of the form below:

$$\begin{aligned} \mathbf{x_{sd}}[k+1] &= \mathbf{A}\mathbf{x_{sd}}[k] + \mathbf{B}\mathbf{u_{sd}}[k] \\ \mathbf{y_{sd}}[k] &= \mathbf{C}\mathbf{x_{sd}}[k] + \mathbf{D}\mathbf{u_{sd}}[k] \end{aligned} \tag{2.1}$$

where $\mathbf{x_{sd}}[k]$ denote the subspace states of the system at a time instant k and is a $(n \times 1)$ vector. The subspace states of the system are abstract in nature and may not directly relate to the states described by the first principles model. The number of states, n_{sd}, is the order of the identified system and is also one of the parameters of the identification procedure. $\mathbf{y_{sd}}[k]$ and $\mathbf{u_{sd}}[k]$ are $(l_{sd} \times 1)$ and $(m \times 1)$ vectors respectively and denote the predicted outputs and manipulated inputs.

For a system of a particular order, the model parameters to be computed include the matrices A, B, C and D, and the initial condition of the state variable for the training data set. In this approach, a state trajectory is first identified and the system matrices are later computed using linear regression. These methods are non-iterative in nature, as opposed to prediction error minimization (PEM) algorithms, which involve estimation using linear and nonlinear optimization problems. Some of the widely used subspace algorithms are canonical variate analysis [21] (CVA), numerical algorithm for subspace state-space identification [24] (N4SID) and multivariate output error state-space [37] (MOESP). These methods were shown to be the interpretation of an unified framework in [25], with the different methods resulting from decomposition of weighing matrix chosen differently in each case. The subspace identification algorithm presented in [23] has been used in the present work.

2.3 Proposed modeling approach

While other instances of hybrid or grey-box models exist in the literature [3, 29, 34, 35, 40], the specific architecture proposed for the hybrid dynamic model is key to achieving the objective of enhancing the first principles models. In the present section, the hybrid modeling approach is described for the more challenging case of batch data (including data from multiple batches of non-uniform length). The first simulation example illustrates the approach using the simpler case of a continuous CSTR followed by the application to the motivating batch PMMA polymerization example.

2.3.1 Model identification

Model identification is the process of using measured data to build a mathematical model of a dynamic system. Following identification, model validation techniques are used to verify that the estimated model accurately reproduces the dynamic behaviour of the system. When identifying a dynamic model, both the model parameters and the initial state of the system are identified. This is true for dynamic models in general, regardless of the technique used, whether that be subspace identification or other techniques such as neural networks. For a new data set, the model simply cannot predict (accurately) until the states of the system have been estimated using a portion of the new data. In contrast, when developing static models, once a model has been developed/trained, it can be directly implemented for new data to predict the modeled variable. To account for this feature of dynamic models, the identification step is described first, followed by the validation step.

Recall that a total of N_T training batches and N_V validation batches are available, with $D_b, b = 1\ldots, N_T$ and $D_v, v = 1, \ldots, N_V$ being the duration of the training and validation batches, respectively. Thus, for the b^{th} batch, input and output data of the form $[u_k, y_k]^b$ is available. It is assumed that a first principles model for the process is available, albeit with mismatch between the process and model. Note that using existing results, a subspace model between the input and outputs can be readily determined. This paper specifically addresses the problem of developing a hybrid model that combines the first principles and data-driven models.

First principles model

The first step in this modeling approach is to create the first principles model using available process knowledge. Note that the key contribution of the proposed approach is not the development of a first principles model, but the utilization of the first principles model within a hybrid modeling framework. To this end, we assume the existence of a first principles model of the form:

$$\begin{aligned} \dot{\mathbf{x}}_{fp} &= f(\mathbf{x}_{fp}, \mathbf{u}_{fp}) \\ \mathbf{y}_{fp} &= g(\mathbf{x}_{fp}, \mathbf{u}_{fp}) \end{aligned} \tag{2.2}$$

where $\mathbf{x}_{fp} \in \mathbf{R}^{n_{fp}}$ denote the $\mathbf{n}_{fp}$ first principles model states, and f and g are functions (possibly nonlinear) which determine the evolution of the states and outputs. $\mathbf{y}_{fp} \in \mathbf{R}^{l_{fp}}$ are the predicted outputs and $\mathbf{u}_{fp} \in \mathbf{R}^{n_m}$ are the inputs. l_{fp} and n_m denote the number of measured outputs and manipulated inputs and are considered the same as the process. As the first step of the hybrid model identification procedure, the predictions for all the batches are generated using known initial conditions of the first principles model and the known input values for the batches. Thus $\mathbf{u}_{fp}$ are chosen to be the same as $\mathbf{u}$ (inputs to the process) to compute the first principles model prediction. The predicted values of the states and outputs at the k th instant for any batch b are respectively denoted by $\mathbf{x}_{fp}^{(b)}[k]$ and $\mathbf{y}_{fp}^{(b)}[k]$.

Remark 1. *We assume the availability of the initial states ($\mathbf{x}_{fp}^{0}$) of the first principles model for all the batches. In general, we may not have measurements of all the initial state values. In such situations, the first principles model would need to be run with an appropriate state estimator first and then be available for prediction after the outputs converge. This is not a limitation of the present work and would be required even in the absence of the hybrid modeling. In other words, if the first principles model is such that the initial states of the model are not available for measurement, the implementation of the first principles model would require the state estimation setup. Thus, the proposed hybrid modeling framework does not impose any additional requirements. An explicit illustration of this scenario is a subject of future work.*

Remark 2. *In deciding whether or not the first principles models alone, the data-driven model alone or the hybrid model alone should be utilized, the following guideline is suggested: For the cross-validation batches, compute the Mean Absolute Scaled Error (MASE) under the first principles model and the best data-driven model only (where the data-driven model is directly modeling the process inputs and outputs). Next, using the number of states in the data-driven model as the upper bound, determine the best hybrid model. The model that yields lowest MASE should be the one chosen. The expected outcomes of this exercise are one of the two: Either the hybrid model is chosen or the data-driven model alone is chosen, with the outcomes dependent on the predictive capability of the first principles model. In particular, if the first principles model is not sufficiently quantitatively informative, the data-driven model will typically* *outperform a poor first principles model and be chosen. Note that such an outcome could also be used as a trigger to re-calibrate the first principles model. On the other hand, if the first principles model has useful information, then the hybrid model is the expected chosen outcome, with the residual data-driven model improving upon the first principle model.*

Remark 3. *The first principles model does not necessarily have to be in the form of an explicitly written differential algebraic equations (DAE) set. Instead, it could very well be a complex industrial simulator which combines several interconnected units to generate the outputs. The only requirement is that the inputs and outputs should be the same as those of the plant. Again, this is not a limitation imposed by the proposed approach, but would need to be in place for the first principles model, to begin with. Finally, for the first principles model to be useful in real time for control purposes, it should be possible for the simulation for each step to be completed significantly faster than the sampling time between measurements.*

Subspace-based residual model

Consider now the scenario where plant data $(\mathbf{y}, \mathbf{u})$ for training and validation is available and the first principles predictions $(\mathbf{y}_{fp})$ have been generated according to Section 2.3.1. This section illustrates the development of the model between the residuals $\mathbf{e} = \mathbf{y} - \mathbf{y}_{fp}$ and the input. Figure 2.1 shows the layout of the model building step.

The identified model is a discrete LTI sytem of the form:

$$\begin{aligned} \mathbf{x}_{hm}[k+1] &= \mathbf{A}_{hm}\mathbf{x}_{hm}[k] + \mathbf{B}_{hm}\mathbf{u}_{hm}[k] \\ \mathbf{e}_{hm}[k] &= \mathbf{C}_{hm}\mathbf{x}_{hm}[k] + \mathbf{D}_{hm}\mathbf{u}_{hm}[k] \end{aligned} \tag{2.3}$$

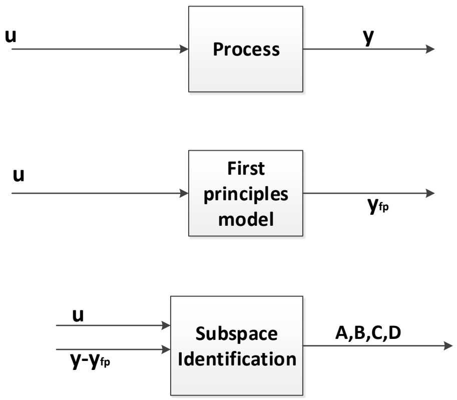


Figure 2.1: A schematic illustrating the integrated modeling approach. The top schematic shows the input-output data collected from the process. The same input is fed to the first principles model to generate the respective outputs. Finally, a model is built between the error and process inputs.

where $\mathbf{x}_{hm} \in \mathbf{R}^{n_{hm}}$ denote the $\mathbf{n}_{hm}$ subspace identified residual model states. A_{hm}, B_{hm}, C_{hm} and D_{hm} are the system matrices. $e_{hm} \in \mathbf{R}^{l_{hm}}$ is the vector of residuals where l_{hm} denotes the size of the vector and is the same as l_{fp} . The system matrices are computed by utilizing the batch subspace identification outlined below. u_{hm} is the input to the model and is same as u_{fp}.

One of the key steps in subspace identification is the arrangement of input-output data to create appropriate Hankel matrices to be used in the algorithm. Instead of using the outputs, in the proposed approach, the Hankel matrix of residuals for the b^{th} batch is formed as shown below:

$$\mathbf{E}_{1|i}^{(b)} = \begin{bmatrix} \mathbf{e}^{(b)}[1] & \mathbf{e}^{(b)}[2] & \cdots & \mathbf{e}^{(b)}[j^{(b)}] \\ \vdots & \vdots & & \vdots \\ \mathbf{e}^{(b)}[i] & \mathbf{e}^{(b)}[i+1] & \cdots & \mathbf{e}^{(b)}[i+j^{(b)}-1] \end{bmatrix} \tag{2.4}$$

A total of $1, 2, \ldots, i, \ldots, i+j-1$ continuous measurements are used to build the matrix. The subscript $1|i$ in this matrix refers to the aspect that elements 1 to i is present in the first column of the Hankel matrix. To appropriately utilize data from multiple batches, the data from each batch is concatenated [6] to create a Hankel like matrix, given by:

$$\mathbf{E}_{1|i} = \begin{bmatrix} \mathbf{E}_{1|i}^{(1)} & \mathbf{E}_{1|i}^{(2)} & \cdots & \mathbf{E}_{1|i}^{(N_T)} \end{bmatrix} \tag{2.5}$$

with N_T being the total number of batches used for training the model.

The input matrix $\mathbf{U}_{1|i}^{(b)}$ is created in a similar manner. The value of i is chosen to be slightly greater than the expected number of subspace states. This modeling approach can naturally handle batches of non-uniform length. Note that the index value i is the same for all batches to create a matrix that has an equal number of rows for all the batches. However, the number of columns can vary depending on the duration of the batches.

From Equation 2.3, through repeated substitutions, the following equation is obtained:

$$\mathbf{E}_h = \mathbf{\Gamma_i}.\mathbf{X} + \mathbf{H}_t.\mathbf{U}_h \tag{2.6}$$

where, $\mathbf{E}_h$ and $\mathbf{U}_h$ are the output and input Hankel matrices, $\mathbf{\Gamma_i}$ is the extended observability matrix, $\mathbf{X}$ is the consecutive state vector sequence and $\mathbf{H}_t$ is the lower triangular block Toeplitz matrix. Two matrices $\mathbf{H_1}$ and $\mathbf{H_2}$ are constructed via the concatenation of $\mathbf{E_{1|i}}$, $\mathbf{U_{1|i}}$ and $\mathbf{E_{i+1|2i}}$, $\mathbf{U_{i+1|2i}}$ respectively, as shown in Equation 3.14. The subscript $i+1|2i$ indicates that elements indexed from $i+1$ to $2i$ are arranged in the first column of that Hankel matrix.

$$\mathbf{H}_1 = \begin{bmatrix} \mathbf{E}_{1|i} \\ \mathbf{U}_{1|i} \end{bmatrix} ; \quad \mathbf{H}_2 = \begin{bmatrix} \mathbf{E}_{i+1|2i} \\ \mathbf{U}_{i+1|2i} \end{bmatrix} \tag{2.7}$$

Next, matrix $\mathbf{H}$ is constructed as the concatenation of $\mathbf{H_1}$ and $\mathbf{H_2}$ as shown below:

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{H}_2 \end{bmatrix}$$

The state vector is realized as the intersection of the row space of the two block Hankel matrices $\mathbf{H_1}$ and $\mathbf{H_2}$. Therefore, the state sequence vector can be obtained by performing an SVD of the matrix $\mathbf{H}$, and is given by $\mathbf{X_2} = [\ \mathbf{x}_{hm}[i+1] \ \ \mathbf{x}_{hm}[i+2] \ \cdots \ \mathbf{x}_{hm}[i+j]\ ]$.

Subsequently, the system matrices are computed using least squares of the over determined set of equations shown below:

$$\begin{bmatrix} \mathbf{x}_{hm}[i+2] & \cdots & \mathbf{x}_{hm}[i+j] \\ \mathbf{e}_{hm}[i+1] & \cdots & \mathbf{e}_{hm}[i+j-1] \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{hm} & \mathbf{B}_{hm} \\ \mathbf{C}_{hm} & \mathbf{D}_{hm} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}_{hm}[i+1] & \cdots & \mathbf{x}_{hm}[i+j-1] \\ \mathbf{u}_{hm}[i+1] & \cdots & \mathbf{u}_{hm}[i+j-1] \end{bmatrix} \tag{2.8}$$

Alternative approaches exists in literature [37], where the system matrices are computed directly from the I/O data without the requirement of first computing the state sequence.

In this subspace-based model building methodology, $\mathbf{n}_{hm}$ and i are the parameters that need to be specified and thus have to be chosen adequately for the model to provide good predictions. Both $\mathbf{n}_{hm}$ and i can be determined by using cross validation, or i chosen heuristically (while satisfying $i > \mathbf{n}_{hm}$, and for the present application, $i = \mathbf{n}_{hm} + 4$. In the existing batch subspace identification [6, 7] results, these parameters were chosen by trial and error. In the present results, a cross validation technique for determining the best number of states is utilized and presented in Section 2.3.3. Having picked the best values of $\mathbf{n}_{hm}$ using cross validation, the entire training data set is utilized to determine the system matrices. Before presenting the cross validation method, however, the model validation approach is presented in the next section to enable a clearer understanding of cross validation.

Remark 4. *One of the key difficulties with the utilization of first principles models is known to be the model maintenance aspect, more so than the model development aspect.*

In other words, while it may be possible to put in the resources to develop a detailed first principles model, practitioners struggle with the resources required to update or maintain the model as process conditions change or new data becomes available. The proposed approach is developed to address model maintenance; therefore the model is not recalibrated as new data becomes available (i.e. the parameters of the original first principles model are not re-identified). Instead, the new data is used to 'correct' the error between the existing first principles model and the process data, thus requiring the development of a model that captures the residual.

Remark 5. *It is important to recognize that the hybrid model does not try simply to model the residuals as a stochastic process. If one were to do that, one would lose the opportunity of recognizing (and quantifying) that the discrepancy between the first principles model and the process could be impacted by the inputs themselves. The proposed hybrid modeling approach builds a dynamic model between the inputs and the residuals between the process outputs and the first principles predictions, allowing the residuals to be predicted forward. Using this structure, the hybrid model is able to capture much of the nonlinear process behaviour that is accessible through the first principles model as well as the remaining (often almost linear) dynamics that is captured by the subspace identification-based model. Thus, the structure of the model chosen in the proposed hybrid model framework is key to the success of the proposed hybrid approach.*

Remark 6. *Note that a hybrid model could very well include the first principles model and an empirical nonlinear model. It is generally expected that the resultant hybrid model would still perform better than a purely first principles or purely empirical nonlinear model. The concern of overfitting with nonlinear empirical models, however, continues to be a research challenge. In contrast, the subspace identification-based approaches generally tend to not overfit, and the 'information' captured through additional states can be readily inferred from inspecting the SVD of the block Hankel matrices, and utilizing cross validation, as illustrated in the manuscript.*

Remark 7. *For continuous processes, the data sets required for building the model can be generated by operating the process around a desired operating point. However, in batch processes, the operation is typically over significantly varied dynamics. Thus, a model which captures the varying dynamics of the process over the entire range is required. Since different batches have varying initial conditions, a model built with data from a single batch would be inept to make desired predictions and, therefore, necessitates the use of training data sets collected from multiple batches. The use of data from multiple batches, especially in a way that recognizes that the endpoint of one batch is not the starting point of another batch is important and is utilized in the proposed approach.*

2.3.2 Model validation

For a fresh batch of data, the model can be used for the purpose of prediction only after the states of the residual model have been adequately computed. A state estimator, denoted by $SE_{hm}(e, u)$, is used initially to estimate the subspace state vector sequence $\bar{\mathbf{x}}_{hm}$ and generate the outputs $\bar{\mathbf{e}}_{hm}$. The estimated output of the hybrid framework is given by $\bar{\mathbf{y}} = \mathbf{y}_{fp} + \bar{\mathbf{e}}_{hm}$. The estimator is used until the output $(\bar{\mathbf{y}})$ converges to the actual process output (y), after which the hybrid model is ready to predict. From this time point on, the converged state is considered to be the initial state of the identified state-space model for prediction. The model is then used to predict the output $\mathbf{e}_{hm}$ using future values of inputs $(\mathbf{u})$. $\mathbf{y}_{pred} = \mathbf{e}_{hm} + \mathbf{y}_{fp}$ is the predicted output of the hybrid model. The above model validation procedure is schematically shown in Figure 2.2.

Thus, the validation sequence consists of first a portion where the outputs from the validation batch are being utilized for state estimation (as shown in Figure 2.8(a)) and the prediction stage where the outputs are being predicted using only the future

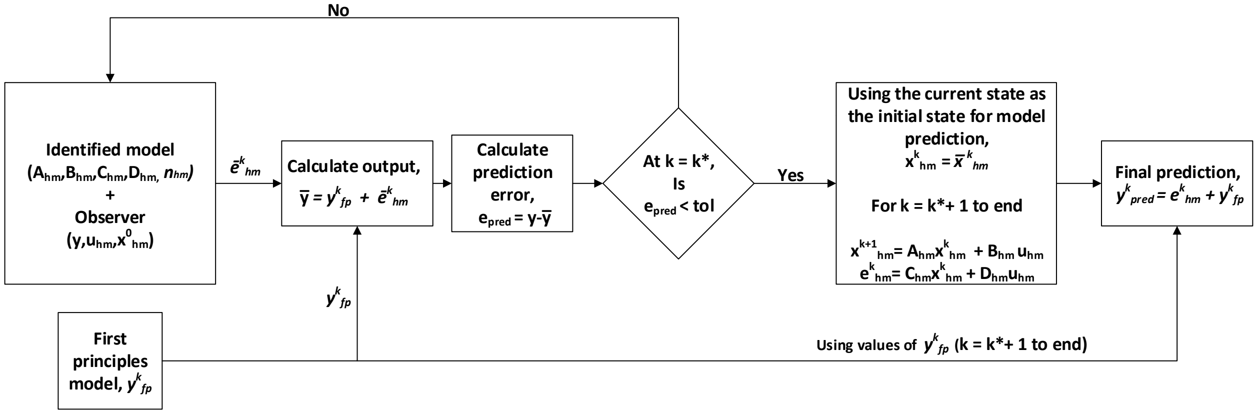
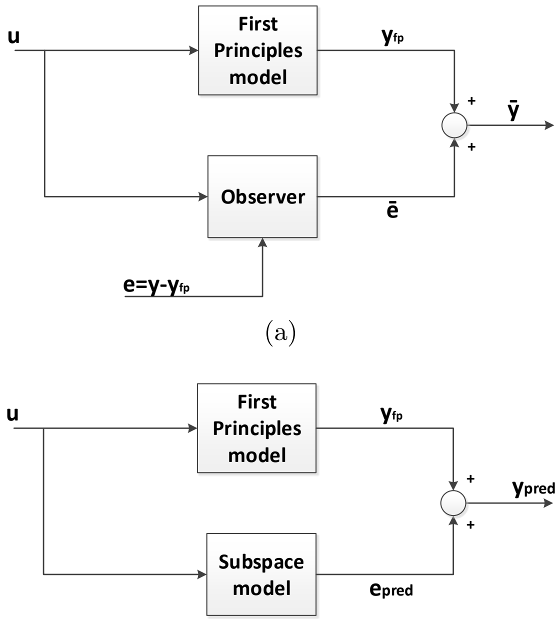

No
Identified model
(A_{hm},B_{hm},C_{hm},D_{hm}, n_{hm})
+
Observer
(y,u_{hm},x^0_{hm})
$\bar{e}^k_{hm}$
Calculate output,
$\bar{y} = y^k_{fp} + \bar{e}^k_{hm}$
Calculate prediction error,
$e_{pred} = y-\bar{y}$
At k = k*,
Is
e_{pred} < tol
Yes
Using the current state as the initial state for model prediction,
$x^k_{hm} = \bar{x}^k_{hm}$
For k = k*+ 1 to end
$x^{k+1}_{hm} = A_{hm}x^k_{hm} + B_{hm}u_{hm}$
$e^k_{hm} = C_{hm}x^k_{hm} + D_{hm}u_{hm}$
Final prediction,
$y^k_{pred} = e^k_{hm} + y^k_{fp}$
y^k_{fp}
First principles model, y^k_{fp}
Using values of y^k_{fp} (k = k*+ 1 to end)

Figure 2.2: A schematic illustrating the validation procedure

inputs and the state estimated during the state estimation step (as shown in Figure 2.8(b)).



(a)

(b)

Figure 2.3: A schematic illustrating the model validation procedure, where (a) shows how the measured outputs and inputs from the present data are used to compute the state estimate and (b) depicts how the model can be used for prediction purposes.

Remark 8. *In this work, a Kalman filter is used for state estimation. The error* *tolerance value considered for convergence using the Kalman filter is chosen be a user specified value in the two case studies. However, it can be set by finding the average standard deviation values of the error trajectories of the training batches. This method, however, is not restricted to this particular choice of the state estimator and any other suitable state estimation scheme can be utilized as a part of the hybrid model.*

Remark 9. *The ability of the model to be able to predict (without requiring measured outputs and observer) is of key importance to their potential use in advanced control strategies. In particular, for instance, at a particular time step, a model predictive control (MPC) formulation utilizes the model and candidate future inputs to predict future outputs and evaluate objective functions. It is this predictive capability of the model that imbues the MPC with all the optimality benefits. Thus, for cross validation purposes, the model's predictive capability is evaluated, not just the ability of the state estimator to estimate outputs and states using measured outputs.*

2.3.3 Cross validation methodology

The cross validation approach to determine the order of the subspace based model is as follows. The batches used for model building are first divided randomly into k groups or partitions. For the purpose of cross validation, these k groups are divided into training groups, which are collectively represented as $CV_{training}$, and the validation or testing group is represented as CV_{test}. For the first run, the first group is chosen to be the validation set (CV_{test}) and the rest of the $k-1$ groups constitute the training set ($CV_{training}$). With a particular choice of the number of states in the identified model, $\mathbf{n}_{hm}$, the model is built using the batches present in the training set. The model is then tested/validated on the validation set to evaluate its predictive capability.

The prediction error is quantified by calculating the mean absolute scaled error of the square error (MASE) between the model prediction and the plant output data and is given by MASE = $\frac{\sum_{t=1}^{T}|e_t|}{\frac{T}{T-1}\sum_{t=2}^{T}|Y_t - Y_{t-1}|}$ where $e_t = Y_t - Y_t^{pred}$, Y_t and Y_t^{pred} being the process output and the predicted output respectively at the t th sampling instant and T is the range over which the error is calculated. This procedure is repeated k times so that each group has acted as the validation set once. The cumulative error value is then calculated for that particular state value. Once this is done, the number of states is increased by one and all the steps are repeated and the final error value recorded. Finally, the model order (number of subspace states) is chosen as the one that yields the least error during the process.

2.3.4 Simulation results for a CSTR

Consider the case where a first order exothermic irreversible reaction A → B takes place in a CSTR that is surrounded by a jacket. The first principles model is given by the equations below and the parameter values are listed in Table 2.1.

$$\begin{aligned}\frac{dC_A}{dt} &= \frac{F}{V}(C_{A,in} - C_a) - k_0 \exp\left(\frac{-E}{RT}\right)C_A \\ \frac{dT_{reac}}{dt} &= \frac{F}{V}(T_{in} - T_{reac}) - \frac{\Delta H}{\rho C_p}k_0 \exp\left(\frac{-E}{RT_{reac}}\right)C_A + \frac{Q}{\rho C_p V}\end{aligned} \tag{2.9}$$

Parameter nomenclature	
$C_{A,in}$: Concentration of A in feed stream	ΔE: Activation energy
C_A: Concentration of A in reactor	V: Reactor volume
ΔH: Heat of reaction	ρ: Density
C_p: Heat capacity	Q: Heat removal/addition
k_0: Pre-exponential factor	F: Volumetric flowrate
R: Ideal gas constant	
T_{reac}: Temperature in the reactor	
T_{in}: Inlet Feed temperature	

Table 2.1: CSTR parameters

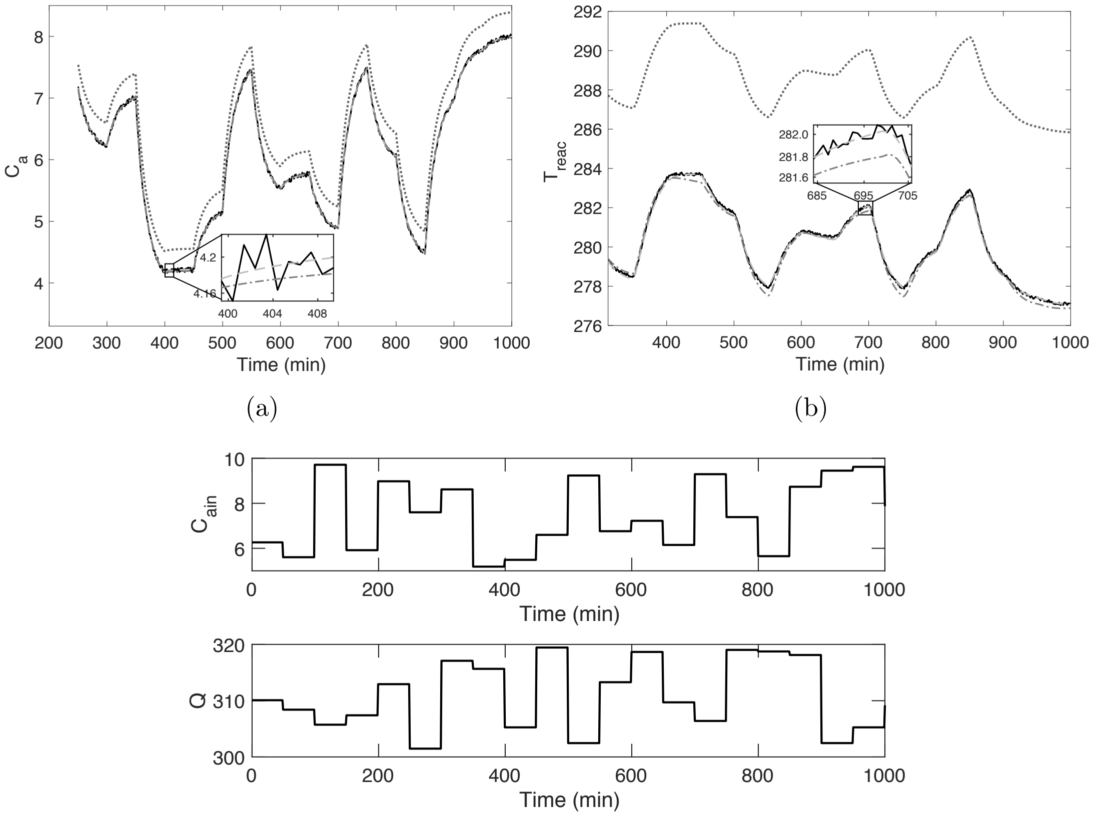
The CSTR setup as described above is utilized as a test bed for the simulation study. All process data is generated by using the above set of differential equations for the CSTR. The inputs considered are the heat removal/addition (Q) and inlet concentration of species A ($C_{A,in}$). The inputs are constrained between bounds (Table 2.5) and their initial value is randomly chosen from a uniform distribution between the boundary values. They are implemented in a way such that they are held constant over a period of 50 time steps, where each step is one minute. The measured outputs are the concentration of species A (C_A) in the reactor and reaction temperature (T_{reac}). The initial values of C_A and the T_{reac} are randomly chosen each time from a normal distribution with a standard deviation value provided in Table 2.6. Plant output data is considered to be corrupted by measurement noise. The noise is assumed to be Gaussian with zero mean and the individual standard deviation value are listed in Table 2.6. A moving average filter is used to attenuate the effect of noise on the output signals. The process is run for 1000 minutes with data sampled every one minute to generate 12 sets of data.

We also assume that we know a model of the process, albeit with error to replicate a real life scenario. The first principles model has the same model structure as the process model, except with 5% deviation in its parameters (5% for ΔE and -5% for k_0). The parameter values considered for this case study are listed in Table 2.4. The training and cross validation is carried out as described in the previous sections. The cross validation approach is implemented to find the best choice of states for both the hybrid and subspace case. To this end, the 12 data sets are randomly divided into four groups as shown in Figure 2.6(a). An initial choice of 7 Hankel rows is considered for both the methods. As explained in the previous Section 2.3.3, the model is trained on three groups and validated on the remaining one group. This training/validation method is carried out until each of the four groups have acted once as the testing set. Model validation is then carried out as per the steps explained in Section 2.3.2. A Kalman filter is used to estimate the states using the knowledge of system matrices, the output measurements, and noise data. The measurement noise data is listed in Table 2.6. The initial state value of the Kalman filter is considered to be a zero vector. The model is used to predict for the rest of the batch once the outputs have converged. For the validation batches, the outputs converge at different time instances. In order to have a fair comparison, all the error values in the different simulation runs were computed after the first 250 time instances.

The cross validation step yields 3 as the best choice of subspace states. The validation results for a fresh run are shown in Figure 2.4 and the prediction error values are listed in Table 2.2. In order to correctly judge the quality of the model, only the model prediction part of the two methods is considered for comparative analysis. The Kalman filter is used till the outputs for both the methods converged and is kept in closed loop up to $k = 250$ th instant. The error values for the model prediction part are calculated from that point onward to the end of the batch. There are a number of observations to be made from the validation results in Figure 2.4. The first is that the first principles model, while not predicting accurately, still captures the general nature of the process, thus making it a useful component of the hybrid model. This is corroborated by the fact that the results from the hybrid model predictions are better when compared to the predictions using a classical subspace model alone. In summary, it can be seen that the hybrid model enables improved predictive capability with respect to both the first principles model and the direct modeling of the process outputs using the subspace-based approach.

Table 2.2: Cumulative MASE Error Analysis

Model Type	Cumulative MASE (12 batches)
First Principles	1620.47
Subspace Identification	72.3703
Proposed Hybrid Modeling Approach	23.3738



(a)

(b)

(c)

Figure 2.4: Validation results for a fresh batch using three different modeling techniques: first principles ($\cdots$), subspace identification ($-.-$) and the proposed hybrid modeling approach ($--$). The process output is represented by the solid black line (—) for comparison. Figures (a) and (b) show the prediction results (excluding the initial state estimation portion) for outputs C_A (mol/m^3) and T_{reac} (K) respectively. The input sequences for $C_{A,in}$ (mol/m^3) (top) and Q (J/hr) (bottom) are shown in (c).

2.4 Applications to the PMMA polymerization reactor

In this section, the proposed methodology is applied to the PMMA batch polymerization process. To this end, first the database is described and later the validation results for new batches are shown.

2.4.1 Database description

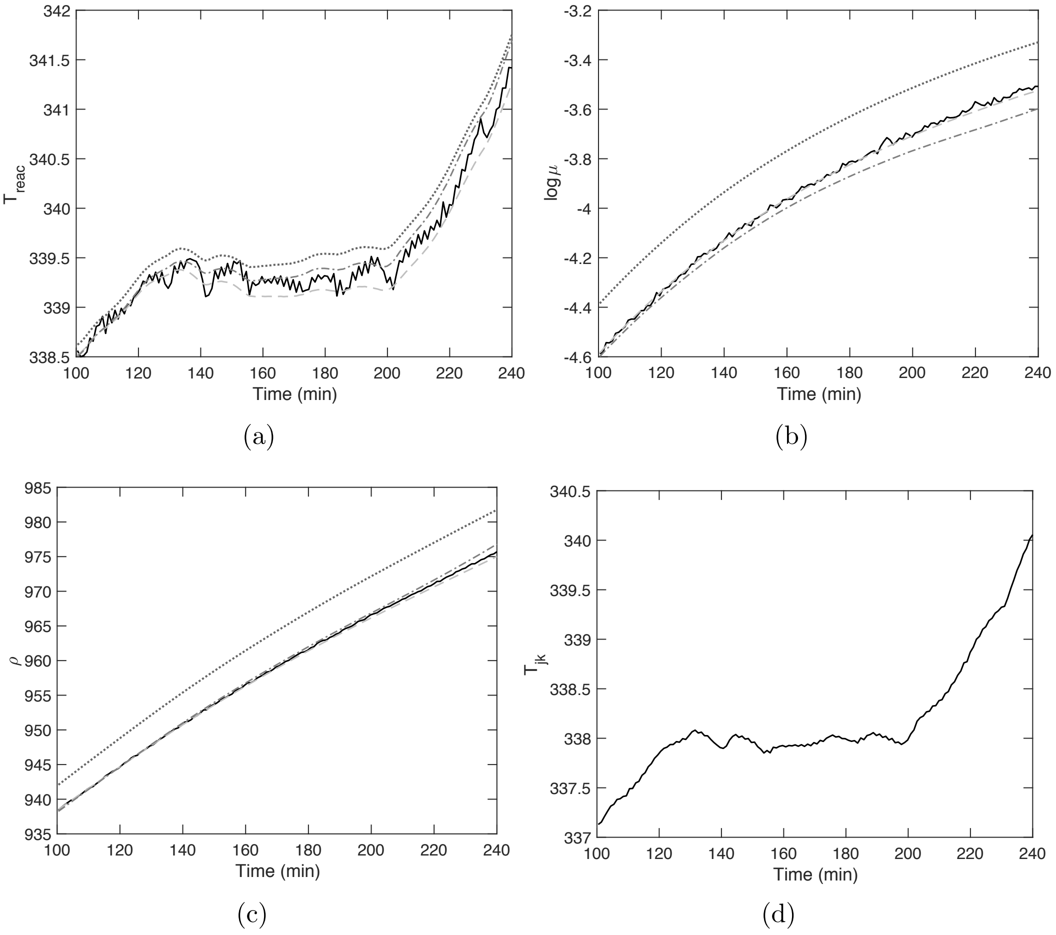
A first principles model for the PMMA polymerization process is used to generate the data. This mechanistic model consists of a series of differential equations and algebraic equations. For model building, the input-output (I/O) data was generated for 18 batches similar to the database [6] generated in an earlier study. Of these 18 batches, six batches were operated with nominal inputs perturbed by a pseudo random binary sequence (PRBS). Note that PRBS are simply used in the present work to illustrate the proposed approach. Specifically designed input signals to yield rich data sets exist in the literature [8, 9]. Nine historical batches were subjected to Proportional Integral (PI) set-point trajectory tracking and rest three were produced by superimposing a PRBS signal to the PI trajectory tracking input. Table 2.7 tabulates the composition of the training data set and also the PI settings used for trajectory tracking. Measurement noise was added to the output data. The noise is considered to be Gaussian white noise with zero mean and standard deviation reported in Table 2.9. As is common practice - a filter, in the present case - a moving average filter is used to attenuate the effect of the noise under PI control. All the batches were started with different initial conditions to imitate real life scenario. The initial batch conditions and the standard deviation data are shown in Table 2.8. For the testing set, I/O data was generated for six batches in a similar fashion and two batches of each type were chosen, as described earlier. The process is run for 240 minutes with data sampled every minute.

2.4.2 Hybrid data-driven modeling of PMMA polymerization

A first principles model is assumed to be available and, in the present example, has the same structure as the process model, but with erroneous parameter values listed in Table 2.10. A model with a greater number of simplifying assumptions that results in a lower value of states, or, in general, with a different structure can also be chosen as the first principles model. Demonstrating the utility of the hybrid modeling approach for such a case remains the subject of future study.

The cross validation methodology is implemented with an initial choice of 12 Hankel rows and 3 subspace states for both the hybrid and subspace model. The 18 training batches considered in this study were divided into six groups with the random selection of three batches in each group as shown in Figure 2.7(a). For the first iteration, the first group is taken as the validation or testing set. Using the subspace identification algorithm [23], the model is built with the remaining five groups listed in Figure 2.7(b).

As explained previously, the initial state estimation for the validation set is carried out with the aid of a Kalman filter. The measurement noise data used in its formulation is listed in Table 2.9 and process noise standard deviation is chosen to be 0.001. The initial state value of the Kalman filter is a zero vector and the error tolerance value for convergence is kept to be $\begin{bmatrix} 0.375 & 0.375 & 0.375 \end{bmatrix}$. Once the outputs have converged, the model is used to predict for the rest of the batch. In cross validation methodology, as the outputs converge at different time instances for different testing groups, all the error values in the different simulation runs in this case are computed for the first 100 time instances. We get the least prediction error using 8 as the number of states. For both modeling approaches (considering 8 states), the model is built (using data from all 18 batches) and the outputs are generated for the validation set (30 batches). The model predictions after convergence for a representative batch in the validation set are shown in Figure 2.5. The cumulative prediction error values for all the 30 validation batches for the first principles, subspace, and hybrid approach are listed in Table 2.3. As can be seen from the MASE values in Table 2.3 and the profiles in Figure 2.5, the hybrid model yields a significant improvement over the subspace model alone or the first principles models alone.



(a)

(b)

(c)

(d)

Figure 2.5: Validation results for one of the batches using the three different modeling techniques: first principles $(\cdots)$, subspace identification $(-.-)$ and the proposed hybrid modeling approach $(--)$. The process output is represented by the solid black line for comparison. Figures (a), (b) and (c) show the predicted results (excluding the initial state estimation portion) for outputs $T_{reac}(K)$, $\log\mu$ and $\rho(Kg/m^3)$ respectively. The input sequence for the jacket temperature $T_{jk}(K)$ is shown in (d).

Table 2.3: Cumulative MASE Error Analysis

Model Type	Cumulative MASE (30 batches)
First Principles	1160.14
Subspace Identification	217.429
Proposed Hybrid Modeling Approach	156.534

2.5 Conclusions and future work

In this manuscript, a hybrid modeling approach was presented that judiciously synergizes first principles models with subspace-based data-driven model identification. The validity of the approach for prediction purposes (not just process monitoring) was illustrated using a CSTR example and demonstrated on a PMMA batch process example. An important aspect in the implementation of the hybrid modeling approach is that the quality of predictions is not greatly hindered by the choice of states in the residual model. Thus, in the simulation results, choosing a number of states even as low as 2 yields a model that is significantly better than the subspace model for the process outputs. It thus provides the flexibility of working with lower order models without significantly diminishing the predictive quality. A more automated approach to select the number of states and observer parameters remains the subject of future work. In addition, the illustration of the proposed modeling approach in a feedback control scenario remains the subject of future work. Additional issues that need to be investigated include the ability to handle missing data and delays, as well as automating the choice (offline *and* online) of using the data driven alone, the first principles alone or the hybrid model based on an appropriate monitoring framework.

2.6 Acknowledgments

Financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC) through the Engage Grant, and the McMaster Advanced Control Consortium (MACC) is gratefully acknowledged.

Bibliography

[1] Aggarwal, C. C. (2018). *Neural Networks and Deep Learning*. Springer International Publishing.

[2] Bonvin, D. (1998). Optimal operation of batch reactors—a personal view. *Journal of Process Control*, 8(5):355 – 368. ADCHEM '97 IFAC Symposium: Advanced Control of Chemical Processes.

[3] Braake, H., van Can, H., and Verbruggen, H. (1998). Semi-mechanistic modeling of chemical processes with neural networks. *Engineering Applications of Artificial Intelligence*, 11(4):507 – 515.

[4] Chen, L., Khatibisepehr, S., Huang, B., Liu, F., and Ding, Y. (2015). Nonlinear process identification in the presence of multiple correlated hidden scheduling variables with missing data. *AIChE Journal*, 61.

[5] Corbett, B., Macdonald, B., and Mhaskar, P. (2015). Model predictive quality control of polymethyl methacrylate. *IEEE Transactions on Control Systems Technology*, 23(2):687–692.

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

[7] 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.

[8] Darby, M. and Nikolaou, M. (2009). Multivariable system identification for integral controllability. *Automatica*, 45:2194–2204.

[9] Darby, M. and Nikolaou, M. (2013). Identification test design for multivariable model-based control: An industrial perspective. *Control Engineering Practice*, 22.

[10] 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.

[11] Ekpo, E. and Mujtaba, I. (2008). Evaluation of neural networks-based controllers in batch polymerisation of methyl methacrylate. *Neurocomputing*, 71(7):1401 – 1412. Progress in Modeling, Theory, and Application of Computational Intelligenc.

[12] 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.

[13] Flores-Cerrillo, J. and MacGregor, J. F. (2002). Control of particle size distributions in emulsion semibatch polymerization using mid-course correction policies. *Industrial & Engineering Chemistry Research*, 41(7):1805–1814.

[14] Flores-Cerrillo, J. and MacGregor, J. F. (2005). Iterative learning control for final batch product quality using partial least squares models. *Industrial & Engineering Chemistry Research*, 44(24):9146–9155.

[15] Garg, A., Corbett, B., Mhaskar, P., Hu, G., and Flores-Cerrillo, J. (2017). Subspace-based model identification of a hydrogen plant startup dynamics. *Computers & Chemical Engineering*, 106:183 – 190. ESCAPE-26.

[16] Garg, A. and Mhaskar, P. (2017). Subspace identification-based modeling and control of batch particulate processes. *Industrial & Engineering Chemistry Research*, 56(26):7491–7502.

[17] Hosen, M. A., Hussain, M. A., and Mjalli, F. S. (2011). Control of polystyrene batch reactors using neural network based model predictive control (nnmpc): An experimental investigation. *Control Engineering Practice*, 19(5):454 – 467.

[18] Katayama, T. (2006). *Subspace Methods for System Identification*. Communications and Control Engineering. Springer London.

[19] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). ImageNet Classification with Deep Convolutional Neural Networks. *COMMUNICATIONS OF THE ACM*, 60(6):84–90.

[20] Kumar, S. S. P., Tulsyan, A., Gopaluni, B., and Loewen, P. (2018). A deep learning architecture for predictive control. *IFAC-PapersOnLine*, 51(18):512 – 517. 10th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2018.

[21] Larimore, W. E. (1990). Canonical variate analysis in identification, filtering, and adaptive control. In *29th IEEE Conference on Decision and Control*, pages 596–604 vol.2.

[22] Li, R., Prasad, V., and Huang, B. (2016). Gaussian mixture model-based ensemble kalman filtering for state and parameter estimation for a pmma process. *Processes*, 4(2).

[23] Moonen M, Demoor B, V. L. V. J. (1989). Online and off-line identication of linear state-space models international journal of control. *International Journal of Control*, pages 49:219–232.

[24] Overschee, P. V. and Moor, B. D. (1992). Two subspace algorithms for the identification of combined deterministic-stochastic systems. In *[1992] Proceedings of the 31st IEEE Conference on Decision and Control*, pages 511–516 vol.1.

[25] Overschee, P. V. and Moor, B. D. (1995). A unifying theorem for three subspace system identification algorithms. *Automatica*, 31(12):1853 – 1864. Trends in System Identification.

[26] Peter van Overschee, B. d. M. (1996). *Subspace Identification for Linear Systems*. Springer US.

[27] Pinto, J. and Ray, W. (1995). The dynamic behavior of continuous solution polymerization reactors—vii. experimental study of a copolymerization reactor. *Chemical Engineering Science*, 50(4):715 – 736.

[28] Poyton, A., Varziri, M., McAuley, K., McLellan, P., and Ramsay, J. (2006). Parameter estimation in continuous-time dynamic models using principal differential analysis. *Computers & Chemical Engineering*, 30(4):698 – 708.

[29] Psichogios, D. C. and Ungar, L. H. (1992). A hybrid neural network-first principles approach to process modeling. *AIChE Journal*, 38(10):1499–1511.

[30] Qin, S. J. (2006). An overview of subspace identification. *Computers & Chemical Engineering*, 30(10):1502 – 1513. Papers form Chemical Process Control VII.

[31] 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.

[32] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representation by back-propagating errors. *Nature*, 323(6088):533–536.

[33] Soroush, M. and Kravaris, C. (1992). Nonlinear control of a batch polymerization reactor: An experimental study. *AIChE Journal*, 38(9):1429–1448.

[34] Su, H.-T., Bhat, N., Minderman, P., and McAvoy, T. (1992). Integrating neural networks with first principles models for dynamic modeling. *IFAC Proceedings Volumes*, 25(5):327 – 332. 3rd IFAC Symposium on Dynamics and Control of Chemical Reactors, Distillation Columns and Batch Processes (DYCORD+ '92), Maryland, USA, 26-29 April.

[35] Thompson, M. L. and Kramer, M. A. (1994). Modeling chemical processes using prior knowledge and neural networks. *AIChE Journal*, 40(8):1328–1340.

[36] Tsen, A. Y.-D., Jang, S. S., Wong, D. S. H., and Joseph, B. (1996). Predictive control of quality in batch polymerization using hybrid ann models. *AIChE Journal*, 42(2):455–465.

[37] Verhaegen, M. and DeWilde, P. (1992). Subspace model identification part 1. the output-error state-space model identification class of algorithms. *International Journal of Control*, 56(5):1187–1210.

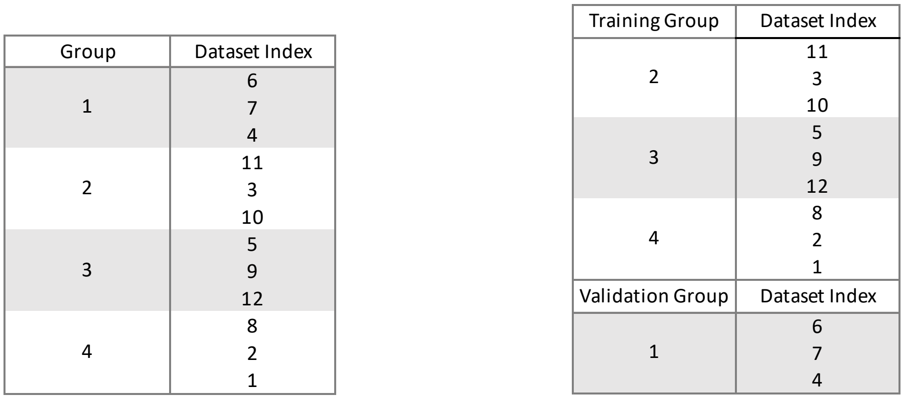
[38] von Stosch, M., Oliveira, R., Peres, J., and de Azevedo, S. F. (2014). Hybrid semi-parametric modeling in process systems engineering: Past, present and future. *Computers & Chemical Engineering*, 60:86 – 101.

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

[40] Wilson, J. and Zorzetto, L. (1997). A generalised approach to process state estimation using hybrid artificial neural network mechanistic models. *Computers & Chemical Engineering*, 21(9):951 – 963.

2.7 Appendices

2.7.1 CSTR simulation



Group	Dataset Index
1	6 7 4
2	11 3 10
3	5 9 12
4	8 2 1

(a)

Training Group	Dataset Index
2	11 3 10
3	5 9 12
4	8 2 1
Validation Group	Dataset Index
1	6 7 4

(b)

Figure 2.6: (a) A schematic showing the groups and the randomly distributed data sets used in the simulation. (b) Data set arrangement for the first iteration in the cross validation methodology showing that the model is built using the training groups (top) and then validated on the remaining group (bottom).

Table 2.4: Parameter values for the CSTR example of Equation 2.9.

Parameter	Value	Unit
F	50	m^3/hr
ρ	1000	kg/m^3
V	0.1	m^3
C_p	0.239	$J/(kgK)$
ΔH	4780	J/mol
k_0	7.2e9	hr^{-1}
T_{in}	310	K
$\Delta E/R$	5000	K

Table 2.5: Input parameters and their bounds.

Input Variable	Bounds	Unit
$C_{A,in}$	[5 10]	mol/m^3
Q	[300 320]	J/hr

Table 2.6: Initial conditions of measured outputs and noise parameters.

Output Variable	Initial values , σ	σ_{noise}	Unit
C_A	2 , 0.5	0.1	mol/m^3
T_{reac}	310 , 5	0.2	K

2.7.2 PMMA batch polymerization simulation

Input policy	Number of Batches
PI trajectory tracking($K_C = -0.1$, $T_I = 0.5$)	9
PI trajectory tracking superimposed with PRBS	3
Nominal input superimposed with PRBS	6
Total batches	18

Table 2.7: Model Identification database summary

Variable	Value	σ	Units
Initiator concentration	$2.06 * 10^{-2}$	.00316	kg/m^3
Monomer concentration	4.57	.07071	kg/m^3
Temperature	61	.15811	°C

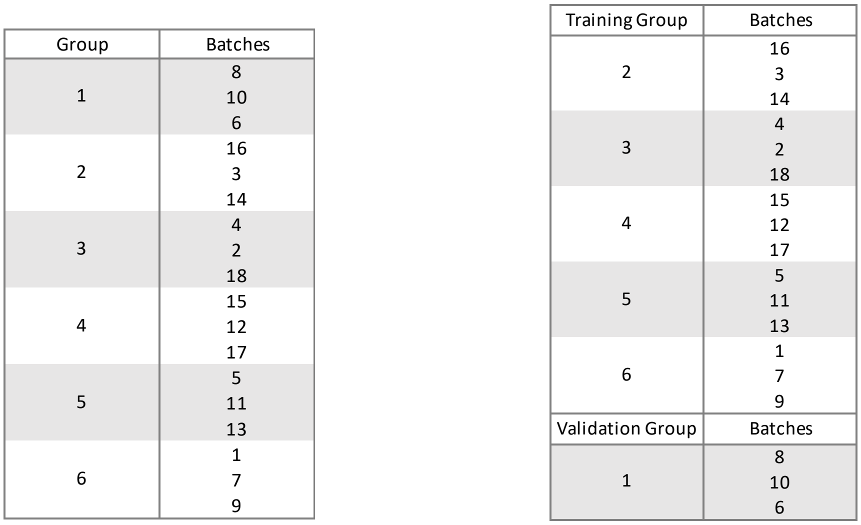
Table 2.8: Initial values and standard deviation data

Output Variable	σ
Temperature	.1
Log viscosity	.01
Density	.1

Table 2.9: Measurement noise data

Table 2.10: The erroneous parameters used in the PMMA first principles model (the original values of these parameters are listed in [11]).

Parameter	% Mismatch
k_d	5
k_{p0}	5
k_{t0}	-5
k_{fm}	5
k_{fs}	5
$k_{\theta p}$	5
$k_{\theta t}$	-5



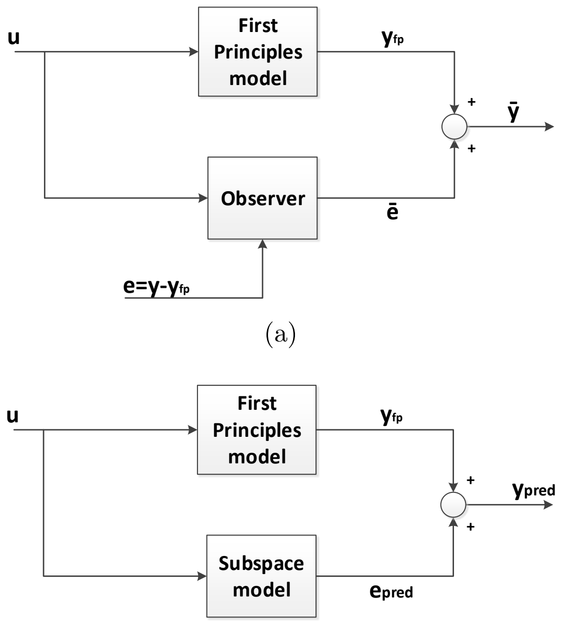
Group	Batches
1	8 10 6
2	16 3 14
3	4 2 18
4	15 12 17
5	5 11 13
6	1 7 9

(a)

Training Group	Batches
2	16 3 14
3	4 2 18
4	15 12 17
5	5 11 13
6	1 7 9
Validation Group	Batches
1	8 10 6

(b)

Figure 2.7: (a) A schematic showing the random arrangement of batches into groups used in the simulation. (b) Arrangement of the groups for the first iteration in the cross validation methodology showing that the model is built using the training groups (top) and then validated on the remaining group (bottom).



(a)

(b)

Figure 2.8: For table of contents only

Chapter 3

Model predictive control embedding a parallel hybrid modeling strategy

The contents of this chapter have been published in *Industrial* & *Engineering Chemistry Research* Journal.

- Model Predictive Control Embedding a Parallel Hybrid Modeling Strategy. Debanjan Ghosh, Jesús Moreira, and Prashant Mhaskar *Industrial* & *Engineering Chemistry Research*2021, 60 (6), 2547-2562 DOI: 10.1021/acs.iecr.0c05208

Abstract

This paper addresses the problem of implementing a model predictive control (MPC) scheme embedding a parallel hybrid subspace model as the predictive component of the control strategy. The hybrid model considered here is inspired by the framework proposed in [1], but it is adapted to make it amenable to online control. In particular, the framework uses a first principles model and a subspace based residual model (built with error between the process measurement data and the first principles output of historical batches) in a parallel fashion. The present manuscript adapts this framework in a way that retains the linearity of the model utilized within the MPC. This is achieved by first building a subspace model (built with output data of first principles model) and then appending it with the residual model to have the same parallel hybrid model structure. This linear hybrid MPC is applied on a seeded batch crystallization process to reduce the volume of fines or crystals generated due to nucleation during the crystallization process, while maintaining a desired product quality at batch termination. The closed loop results using the proposed control methodology are compared with purely data driven subspace based model predictive controller.

3.1 Introduction

The need for high purity products in pharmaceutical, polymer and food industries (to name a few) has really pushed the frontiers in the development of high quality models and control strategies. The finite duration, highly precise batch and fed-batch processes with the small scale nature of their reactors facilitate on-spec product and provide flexibility to the ever changing market demand, thus, making them the ideal reactors for manufacturing these products. Many such applications employ crystallization and the desired product has very specific requirements where the quality is defined by the final crystal size distribution, average particle size, morphology of the crystals, bulk density, etc.

In the past, open-loop control policies of such processes were implemented to get the desired batch end product properties. The idea was that the input policy, which led to a successful batch, would ensure desired quality even for future such applications. However, these strategies fail to accommodate the batch to batch variations and within batch disturbances leading to off-spec product. These challenges motivated the use of closed loop feedback control strategies. In a closed loop proportional-integral (PI) control implementation [2], the controller is made to track a predefined set-point trajectory of a process variable. With proper tuning, these controllers were quite effective in rejecting disturbances and tracking the set-point profile. However, this does not necessarily ensure that tight quality requirements are met as inevitable changes in initial batch and process operating conditions could lead to situations where the predefined manipulated variable trajectory may not effect the process variables as expected. Such implementations are also not very well suited for handling multi-variable control problems and the input constraints are enforced in an ad-hoc fashion which is not ideal.

To address these drawbacks, model based control approaches [3, 4, 5] are implemented which typically have some form of the process model embedded in them. Availability of the process model [6] facilitates generation of optimum operation policies of system variables which can be performed off-line. Model predictive control (MPC) is one widely used model based control technique which is perfect for handling the multi-variate nature of complex industrial control problems. The predictive model is the cornerstone of such strategies and thus significant effort is being continually put to build MPC relevant high quality models. In the context of crystallization processes, the models typically utilize population balance equations [4, 7, 5] along with mass and energy balances, and are written in the form of partial and ordinary differential equations. Such models describe the process dynamics with relatively good accuracy, but are not suitable for real time optimization and control due to their distributed parameter nature. Finite dimensional approximations of these models also lead to very high order ODE systems which are not suitable for control applications and typically involve large computation times but remain difficult to develop and maintain. Reduced order moment models described by ordinary differential equations (ODE) can capture the dominant dynamics of the process fairly well and have been successfully used in the development of efficient controllers. Such simplified first principle models have found applicability in MPC schemes (e.g., [5, 4, 8]).

With the rapid developments in sensing, data storage and computation facilities, there has been a significant push towards building data based/black-box models. Various such modeling techniques (latent variable methods, subspace identification, neural networks etc.) exist which make use of historical batch data to identify their model parameters. Projection to latent structures (PLS) is one such latent variable statistical technique which creates a simple linear model and has found extensive applicability in modeling, monitoring and control of batch processes [9, 10]. These models are inherently static, and in order to accommodate the time varying nature of batch processes, they are modified to behave like time indexed dynamic models with the help of missing data algorithms. Since batches can have variable completion times, proper alignment approaches (either by finding an appropriate alignment variable or dynamic time warping approaches) are also necessary before building the models. Latent variable modeling based MPC (LV-MPC) [10, 11] were used successfully in the past to track batch trajectories. These computed control actions over the prediction horizon are generated using the missing data algorithms and in some sense are influenced by the correlations in historical batches. Multimodel approaches [12, 13, 14] exist which exploit the different features and qualities of respective data models involved in the framework. [12, 13] successfully implemented such techniques where Auto-Regressive Exogenous (ARX) and PLS models have been applied for modeling and control of batch polymerization processes.

Subspace Identification [15, 16, 17, 18] is another statistical modeling approach which allows identification of dynamic models having a linear time invariant state-space (LTI) structure. Identification involves first finding a state trajectory that captures the time varying dynamics and then the model parameters are generated through an appropriate regression. [19, 20] used this identification scheme to successfully model and control continuous processes.

This identification technique was adapted for batch processes in [21] and appropriate adjustments were made to accommodate batches of different sizes. This adaptation has been extensively used with great success for modeling and control of batch and batch-like processes [22, 23, 24, 25]. An MPC based on a subspace identified model for a seeded batch crystallizer was presented in [24] and the goal was to reduce the volume of fines generated due to nucleation. More recently, an MPC for an uniaxial rotomolding process was presented in [25] where a subspace model augmented with a quality based PLS model was built with historical data and then embedded in a MPC to drive the process towards a desired quality. Koopman operator approach is another technique that identifies linear state space models and recent contributions [26] have shown their application in predictive modeling and control schemes.

Non-linear data based models like Artificial Neural Network (ANN) [27, 28], non-linear PLS [29, 30] have been used in the past to capture batch process dynamics using non-linear functions. [28] used a Recurrent Neural Network (RNN) based model for a fed-batch multistage sugar crystallization process and an adaptation of non-linear MPC for the control problem. However, finding the right parameters for the neural network based models largely depend on the amount and quality of the data available. Solving the associated overfitting issue is still a big challenge and an area of active research. The purely data-driven modeling approaches also mostly do not harness the plethora of domain knowledge, making them difficult to interpret.

Grey-box or hybrid modeling technique is one such approach that takes advantage of both worlds and allows integration of process knowledge through physics based models with data-driven modeling techniques. They are broadly classified into series[31, 32, 33] and parallel [34, 35, 1] approaches based on the orientation of individual components of the model framework. An excellent overview of such models and their comparison is presented in [36, 37]. Some examples of series approach can be found in [33, 32] where ANN's were used in series with a mechanistic model to estimate its parameters and later implemented in a control framework. Deep neural network (DNN) was considered for such an approach in a recent study [38], where it successfully applied to a hydraulic fracturing case study. [34] proposed the parallel hybrid approach where an ANN was trained with the residual or error between a first principles model's output and the actual process output. The ANN model prediction was then used in conjunction with a reduced order first principle model's output to predict the outputs for a continuous polymerization reactor. Most of these parallel-hybrid approaches used in the past involve non-linear ANN models and thus may still face the overfitting issue. [35, 39] used a PLS based residual model in the parallel framework for batch to batch and within batch control implementations respectively.

Process knowledge aided with the superior modeling capabilities of subspace identified models was explored recently and a subspace based parallel hybrid scheme was proposed in [1] for a batch PMMA polymerization reactor. These statistical subspace models, elegantly handle the over-fitting issue by choosing the model order via a cross validation scheme. The hybrid model predicted the outputs with much better accuracy than the purely data-based subspace model and the non-linear first principles model. The parallel hybrid approach presented in [1] can make use of any available process knowledge, be it, dynamic equations (written down in the form of PDE, ODE, algebraic equations) or even existing complex industrial simulators. However, usage of such high fidelity models for real time monitoring and control comes with several challenges involving large computation times, numerical issues etc., and is often not a pragmatic option. Therein lies the utility of creating simple yet effective models that can be readily implemented for feedback control.

Motivated by the above consideration, the present work proposes a hybrid model based linear MPC design. The key is the use of a linear hybrid model for MPC calculations and this involves an appropriate identification and utilization of the detailed non-linear model. The rest of the manuscript is organized as follows: Section 4.2.1 presents the first principle based dynamic equations for seeded batch crystallization process (to be used as the test bed) followed by an overview of subspace based identification methods, subspace based parallel hybrid modeling strategy and the basics of model predictive control. In Section 4.3, the identification of parameters of the parallel hybrid model using subspace based methods is reviewed followed by the validation schemes. Section 3.3.3 presents the hybrid architecture of the predictive model inside the MPC along with the closed loop structure in Section 3.3.4. In Section 3.4.1, the proposed linear hybrid modeling strategy is applied on the batch crystallization process and its predictions are compared with other methodologies. Section 3.4.2 presents the optimization formulation for the current study and the results are presented in Section 3.5. Finally, the conclusion is presented in Section 5.4.

3.2 Preliminaries

In this section, a brief review of the first-principle-based equations dictating the dynamics of seeded batch crystallizer is followed by a brief overview on subspace based identification methods and model predictive control.

3.2.1 Motivating example: Seeded Batch Crystallization process

A seeded batch crystallizer is chosen as the simulation test bed for motivating the research. The crystallizer is initiated with a solution and nucleation is induced artificially by seeding (introducing seeds of certain quantity and desired quality in the solution). Seeding is particularly essential in such processes to prohibit unwanted nucleation and achieve desired particle size distribution at batch end. The batch crystallizer process is modeled by a population balance equation (PDE) which describes the evolution of the crystal size distribution $n(r,t)$ over time and two ODE's describing the evolution of solute concentration (C) and reaction temperature (T) respectively as shown in Equation (4.1) (see Table 4.3 for the process parameters and notation). Temperature inside the reactor is regulated by a coolant (at temperature T_j). In this case study, agglomeration and breakage of crystals are considered to be negligible- note that this is simply the case for the test bed and any other test bed (that includes these effects) can be readily utilized. This model is considered as the 'true process/plant' and the data generated via simulating this model (with Gaussian noise added) as 'measurement data'. All the process variables and parameters with their notations are explained in Table 4.3. In particular, the model for the test bed is as follows:

$$\begin{aligned}
&\frac{\partial n(r,t)}{\partial t} + G(t)\frac{\partial n(r,t)}{\partial r} = 0, \quad n(0,t) = \frac{B(t)}{G(t)} \\
&\frac{dC}{dt} = -3\rho k_v G(t)\mu_2(t) \\
&\frac{dT}{dt} = -\frac{UA}{MC_p}(T - T_j) - \frac{\Delta H}{C_p}3\rho k_v G(t)\mu_2(t)
\end{aligned} \tag{3.1}$$

where the nucleation rate $B(t)$ and the growth rate $G(t)$ are respectively given by

Equation (4.2):

$$\begin{aligned} B(t) &= k_b e^{-E_b/RT} (\frac{C - C_s(T)}{C_s(T)})^b \mu_3 \\ G(t) &= k_g e^{-E_g/RT} (\frac{C - C_s(T)}{C_s(T)})^g \end{aligned} \tag{3.2}$$

where b and g are exponents relating the nucleation rate and growth rate to supersaturation.

The equations describing the moments is given by Equation (4.3)

$$\mu_i = \int_0^\infty r^i n(r,t) dr \qquad\qquad i = 0, 1, 2, 3 \cdots \tag{3.3}$$

For crystals to grow, the solution concentration (C) must be in between the saturation and metastable concentrations, C_s and C_m, respectively, i.e., $C_s \leq C \leq C_m$ at all times during the batch run. The saturation and metastable concentration dependence on temperature is given as follows:

$$\begin{aligned} C_s(T) &= 6.29 * 10^{-2} + 2.46 * 10^{-3} T - 7.14 * 10^{-6} T^2 \\ C_m(T) &= 7.76 * 10^{-2} + 2.46 * 10^{-3} T - 8.10 * 10^{-6} T^2 \end{aligned} \tag{3.4}$$

The PDE model can be approximated by a simpler ODE moments model with fewer degrees of freedom [4] and for most practical engineering purposes, only knowledge about certain dominant aspects of the distribution is necessary to determine and manipulate quantity and quality of the product. The set of differential equations describe the evolution of those moments over time and are given by Equation (4.7), where the process model states (x) is a vector given by : $x = \begin{bmatrix} T & C & \mu_0^n & \mu_i^n & \mu_0^s & \mu_i^s \end{bmatrix}^T$, $i = 1, 2, 3$. This reduced order model is considered as the first principle model and assumed to be available, although with erroneous parameter values.

$$\begin{aligned}
&\frac{d\mu_0^n}{dt} = B(t) \\
&\frac{d\mu_i^n}{dt} = iG(t)\mu_{i-1}^n(t) \qquad\qquad i = 1,2,3 \\
&\mu_0^s = k_4 \\
&\frac{d\mu_i^s}{dt} = iG(t)\mu_{i-1}^s(t) \qquad\qquad i = 1,2,3 \\
&\frac{dC}{dt} = -3\rho k_v G(t)(\mu_2^n(t) + \mu_2^s(t) \\
&\frac{dT}{dt} = -\frac{UA}{MC_p}(T - T_j) - \frac{\Delta H}{C_p}3\rho k_v G(t)(\mu_2^n(t) + \mu_2^s(t))
\end{aligned} \tag{3.5}$$

$$\begin{aligned}
\mu_i^n &= \int_0^{r_g} r^i n(r,t)dr \\
\mu_i^s &= \int_{r_g}^{\infty} r^i n(r,t)dr \qquad\qquad i = 0,1,2,3\cdots
\end{aligned} \tag{3.6}$$

μ_i^n and μ_i^s are the statistical moments of the crystal size distribution $n(r,t)$ generated due to nucleation and seeding respectively. These moments can be derived from the crystal size distribution and are defined using Equation (4.5) where r_g is the user specified radius below which the crystals are considered as fines.

An additional condition is enforced on both the models which ensures that no further crystals are formed when the reaction concentration goes below the saturation concentration and is given by the condition shown in Algorithm 2 below. This is particularly important with respect to the simulation as $C \leq C_s$ would lead to complex values for nucleation and growth rate as can be seen from Equation (4.2) owing to fractional value of exponents b and g. In the case of the PDE model, the value of G is further chosen not to be 0 to avoid NaN values during the calculation of crystal size distribution as per Equation (4.1).

The PDE model is solved to generate data that resemble online process sensor record-

Algorithm 1 Growth rate and nucleation rate dependence on reaction concentration

```
if $C \leq C_s$ then
    $B = 0$
    $G = \begin{cases} 1e-6 & \text{PDE model} \\ 0 & \text{Moments model} \end{cases}$
else
    Use the values generated by Equation (4.2)
end if
```

ings. The value of parameters are mentioned in Table 4.4. In this case study, online measurements of all the first principle model states are assumed to be available and the output vector is given by, $y = \begin{bmatrix} \mu_0^n & \mu_i^n & \mu_0^s & \mu_i^s & C & T & C_s & C_m \end{bmatrix}^T$, $i = 1, 2, 3$; input vector is given by $u = T_j$. The input-output (I/O) time series batch data is generated while having batch to batch variations in initial conditions (Tables 4.4 and 4.6) to imitate realistic situations. All of these batches were operated in closed loop under Proportional Integral (PI) control and the PI settings are provided in Table 4.7. An input saturation constraint is also present which bounds the manipulated input within $[50\ 30]°C$. The seed distribution at batch initialization in the crystallizer is considered to be parabolic similar to the one used in[4, 24]. The radius range considered is 300 to 350 μm with the peak occurring at 325μm and having 2/μm g solvent. The initial seed distribution has variations in each batch and all the generated measurement data also have random Gaussian noise to reflect the presence of sensor induced noise in true measurements. The crystal size distribution at batch initialization is given by the Equation (4.8) and a schematic of the same is shown in Figure 3.1.

$$n(r,0) = \begin{cases} 0 & r < 300\mu m \\ 0.0032(350-r)(r-300) & 300\mu m \leq r \leq 350\mu m \\ 0 & r > 350\mu m \end{cases} \tag{3.7}$$

Data from $N_T = 40$ training batches and $N_V = 15$ validation batches of equal duration

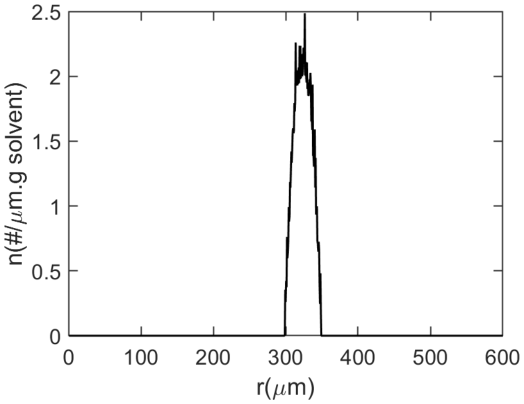


Figure 3.1: Nominal seed distribution at batch initialisation

are respectively used to identify and test the data based models. Out of these 40 training batches, 30 were operated having a optimum temperature (T_{opt}) set-point trajectory 1 as shown in Figure 3.3. To improve the richness of data, the other 10 batches had a set-point trajectory profile 2 as shown in Figure 3.3.

Each batch was initialized with a different condition and Gaussian white noise was added to the output data to imitate real life process conditions. To replicate the effect of impurities in raw materials which induces variations among batches, the exponent g in Equation (4.2) was chosen randomly from a normal distribution with a mean (1.5) and standard deviation (0.001) for different batches. The variations in initial conditions of the measured outputs and noise parameters are reported in Table 4.6.

A moving average filter of window 5 was used to diminish the effect of noisy measurements under PI control. The batches were simulated with the process conditions described in Table 4.6 and run for 30 minutes with a certain sampling interval (dt). Figure 4.2 show the batch wise variations of some of the outputs and inputs. The final crystal size distribution profiles of all the training batches is also shown in Figure 4.2.

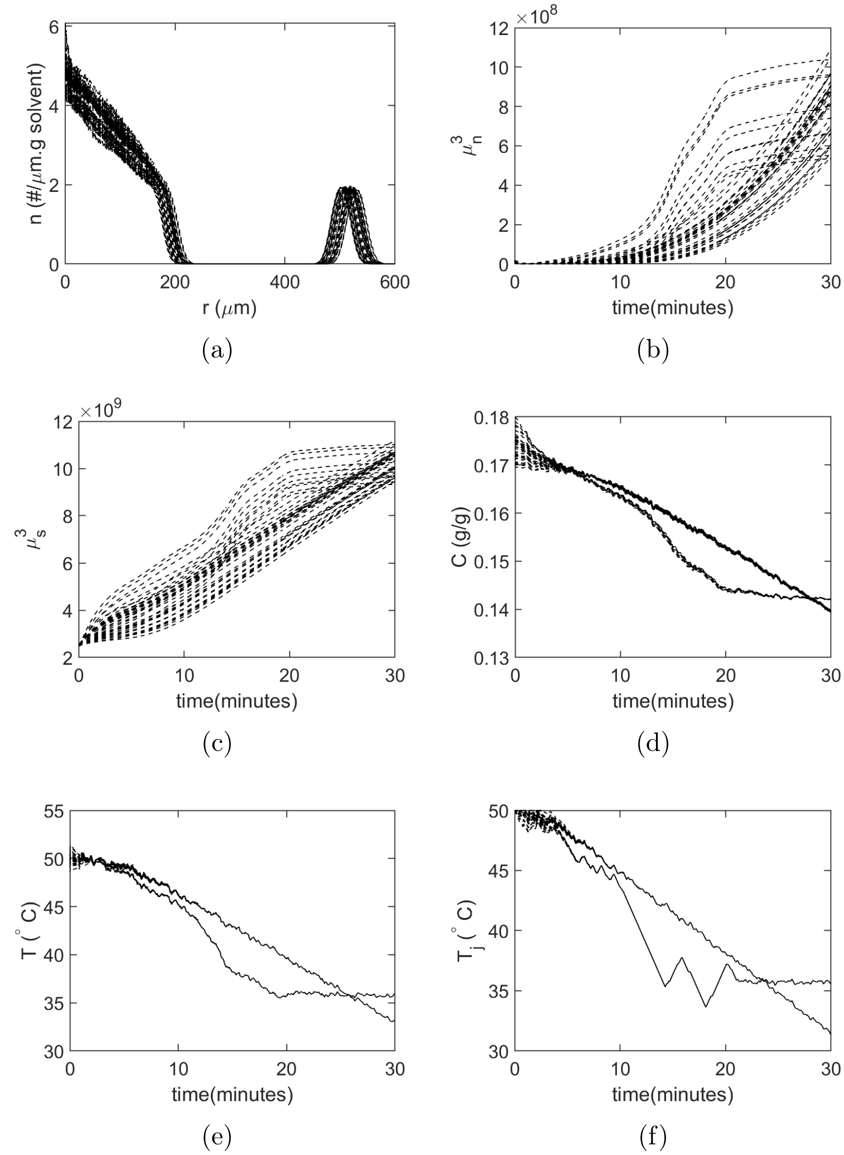
The crystals growing from unwanted nucleation during the crystallization process happen to be of smaller size which degrade the final batch quality (wider PSD) and also cause issues in downstream processing. The aim of the control strategy in this work is to minimize the volume of fines (specifically μ_n^3) while maintaining a certain product quality. The basis for quality here is chosen to be the volume of crystals growing from seeds and specifically given by μ_s^3. This is to be achieved by finding an optimum cooling policy (inputs) while respecting certain constraints pertaining to both inputs (T_j) and certain outputs (C, μ_s^3).

3.2.2 Subspace Identification

Subspace identification methods (SIM)[40, 17, 16, 41] are statistical system identification techniques and identify a discrete time LTI state-space model from I/O data of the form given by Equation (4.10). These approaches utilize mathematical decomposition techniques like singular value decomposition (SVD) or QR factorization to find appropriate model parameters from subspaces of certain data matrices. These algorithms are not iterative unlike prediction error minimization (PEM) algorithms (which use non-linear iterative optimization techniques to find model parameters). SIM's are found to be numerically very robust and efficient for handling large data sets. In this work, the approach proposed by [16] has been adopted to determine a discrete time linear time invariant model of the following form:

$$\begin{aligned} \mathbf{x_{sd}}[k+1] &= \mathbf{A}\mathbf{x_{sd}}[k] + \mathbf{B}\mathbf{u_{sd}}[k] \\ \mathbf{y_{sd}}[k] &= \mathbf{C}\mathbf{x_{sd}}[k] + \mathbf{D}\mathbf{u_{sd}}[k] \end{aligned} \tag{3.8}$$

where $\mathbf{x_{sd}}[k] \in \mathbb{R}^{n_{sd}\times 1}$ represent the subspace states of the identified model at any time instant k and is a $n_{sd}\times 1$ vector, where n_{sd} is the identified model order. $\mathbf{y_{sd}}[k] \in \mathbb{R}^{l_{sd}\times 1}$ and $\mathbf{u_{sd}}[k] \in \mathbb{R}^{m_{sd}\times 1}$ are the output and input vectors where l_{sd} and m_{sd} represent the number of measured outputs and inputs respectively. $\mathbf{A} \in \mathbb{R}^{\mathbf{n_{sd}}\times\mathbf{n_{sd}}}$, $\mathbf{B} \in \mathbb{R}^{\mathbf{n_{sd}}\times\mathbf{m_{sd}}}$,$\mathbf{C} \in \mathbb{R}^{\mathbf{l_{sd}}\times\mathbf{n_{sd}}}$, $\mathbf{D} \in \mathbb{R}^{\mathbf{l_{sd}}\times\mathbf{m_{sd}}}$ are primarily the identified model parameters by this identification procedure.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 3.2: Few output and input profiles generated by the PDE model considered as process data for 40 batches are shown in the figures. Figures (a) represents the crystal size distribution at batch end; (b) and (c) show the profile of 3rd moment due to nucleation (μ_n^3) and 3rd moment due to growth (μ_s^3) respectively; (d) and (e) show the concentration and temperature profile data over the course of batch; Figure (f) represent the input profile trajectory

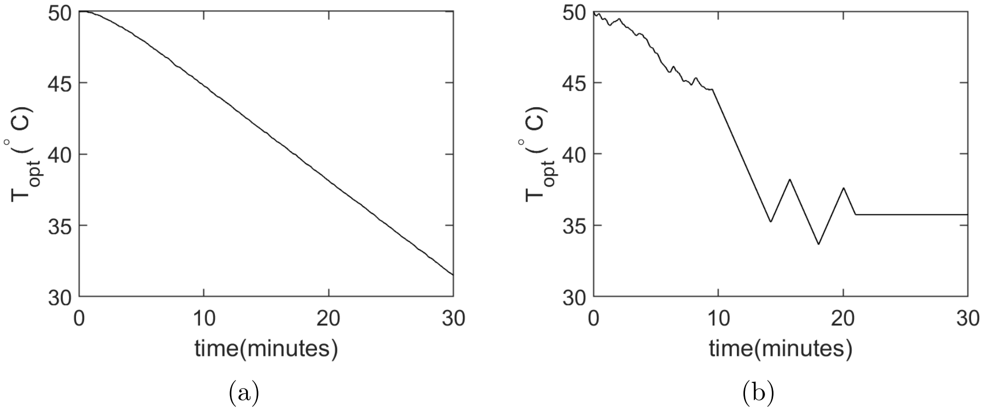


Figure 3.3: Figure (a) and (b) show the set point profile 1 and 2 respectively of the reactor temperature used to generate data for identification

3.2.3 Parallel hybrid modeling framework involving subspace identification

A parallel hybrid modeling strategy utilizing a first principles model to work in conjunction with a subspace based dynamic model, where the subspace model leverages residual information (difference between the first principles model outputs and the process outputs) to improve the predictions, and is next reviewed.

The approach was proposed in [1], where the residual model was built based on a subspace based identification scheme and then added to the non-linear first principles model in a parallel fashion. This framework consists of two distinct models (first principles and residual) and is illustrated briefly below. To this end, consider a generic form of the first principles model (that is consistent with the first principles model of the crystallization process described in Section 4.2.1) given by Equation (3.9):

$$\begin{aligned}\dot{\mathbf{x}}_{fp} &= f(\mathbf{x}_{fp}, \mathbf{u}_{fp}) \\ \mathbf{y}_{fp} &= g(\mathbf{x}_{fp}, \mathbf{u}_{fp})\end{aligned} \tag{3.9}$$

where $\mathbf{x}_{fp} \in \mathbb{R}^{n_{fp}\times 1}$ is the vector of n_{fp} first principles model states. The time evolution of the states and outputs are given by the functions (linear/non-linear) f and g respectively. $\mathbf{y}_{fp} \in \mathbb{R}^{l_{fp}\times 1}$ are the first principle predictions of the process outputs and $\mathbf{u}_{fp} \in \mathbb{R}^{n_m\times 1}$ are the inputs.

The availability of the first principles model permits the usage of an error/residual model. The residual model captures the information not explained by the first principles model and is built with residual/error data $(\mathbf{y} - \mathbf{y}_{fp})$ and the inputs $(\mathbf{u})$ of the historical batches. A subspace identification algorithm is employed to build the model and has the same state space structure as given by Equation (4.10), where (A,B,C,D) are the identified model matrices and the outputs are denoted by $\mathbf{e}_{res} \in \mathbb{R}^{l_{sd}\times 1}$ in recognition that they are residuals (instead of conventional process outputs $\mathbf{y}_{sd}$). l_{sd} is therefore the equal to l_{fp}. Once the model is built, the validation for a new batch is carried out by feeding the same inputs $(\mathbf{u}_{fp} = \mathbf{u}_{sd})$ to both the first principles as well as the residual model to generate outputs of the respective models. The process output predicted by the hybrid model is given by Equation (3.10).

$$\mathbf{y}[\mathbf{k}] = \mathbf{y_{fp}}[k] + \mathbf{e_{res}}[k] \tag{3.10}$$

One of the biggest advantage of this framework is the ease of maintaining such models over a period of time. Here, rather than the conventional and challenging way of re-evaluating the parameters of the first principles model, the maintenance is done by re-identifying the model matrices of the residual model using data from the latest batches to account for the mismatch.

3.2.4 Model Predictive Control

Model Predictive Control (MPC) is a model based control strategy which employs a dynamic optimization to compute the optimal input profiles to steer the process towards a desired objective. Typically, the objective is to drive the system towards an equilibrium point or track a desired output set point profile by minimizing the error between model predictions and the reference. However, it can also address economic objectives. For batch processes, for instance, these include maximizing the quality of the final product or minimizing an impurity formed during the process. A generic representation of the formulation is presented below:

$$\begin{aligned} \min_{u} \quad & f(x_i, u_i) \\ s.t. \quad & g_i(x_i, u_i) \geq 0 \\ & p_i(x_i, u_i) = 0 \\ & x_{i+1} = h(x_i, u_i) \end{aligned} \tag{3.11}$$

$f(x_i, u_i)$ represents the objective function which is to be minimized and is a function of the current states (x_i) and inputs (u_i). The functions $g(x_i, u_i)$ and $p_i(x_i, u_i)$ represent the inequality and equality constraints respectively pertaining to certain outputs and inputs. x_{i+1} is the future state value and calculated by the function $h(x_i, u_i)$ with use of the current state and input values. This function is a generic representation of the dynamic model equations (linear/non-linear set of equations) embedded inside the MPC. The present work makes use of a linear model, and a linear objective with linear constraints makes the MPC optimization a linear program, which is fast, easy to solve and readily implementable. Typically, the constraints and objective function values are evaluated over a certain prediction horizon with the aid of the predictive model. The optimal inputs are generated for the entire horizon (typically till the end for a batch process) and only the first control move is implemented on the process.

The optimization is repeated at each time step after updating the states using online measurements. In case the states are not measured, appropriate state estimation techniques are used to reconstruct the states using available measurements.

Complex non-linear models for optimization introduce non-convexity in the problem which suffer from global optimality issues rendering such problems hard to solve. Developing efficient algorithms for these problems is an area of active research and current techniques are not ideal for real time optimization and control of processes where dynamics change quite fast and require decisions frequently. Complex industrial simulators also face significant challenges in such implementations due to their rigorous and detailed nature leading to large run times in their usage for optimization. These challenges motivate the use of simple, linear models and form the rationale behind developing the proposed linear hybrid model presented in the following section.

3.3 Proposed hybrid modeling and control approach

A first principles model is assumed to be available along with historical time series measurement data of batches. The proposed model, although linear, has an architecture similar to the form presented in [1]. In this section, first the model identification part is described in detail followed by the validation. The two different arrangements of the linear hybrid model are presented in the final subsection followed by the closed loop feedback control structure.

3.3.1 Model identification

A total of N_T training batches is assumed to be available and the first principles model is simulated using initial conditions of those historical batches.

A generic identification scheme is presented which holds true for the two different subspace identified models (built with different type of output data) presented in the following subsections. The identification procedure begins with arranging the data in a Hankel matrix. Conventional subspace techniques use steady state data (operation around a nominal point) from continuous processes to configure the Hankel matrices. To translate that to batch processes, one approach could be to incorporate information of one batch into the matrix. A Hankel matrix built with such data will not have sufficient information about variations and disturbances encountered in multiple batches. The conventional approach is thus not adequate to handle the varied nature and non-uniform batch lengths of different batches. To alleviate this issue, a pseudo-Hankel matrix was proposed in [21] which appropriately assigns respective batch data to distinct positions in the matrix. The approach is adopted in this work and thus briefly illustrated below:

The Hankel matrix for any output data (y) pertaining to a batch (b) is given by Equation (3.12)

$$\mathbf{Y}_{1|i}^{(b)} = \begin{bmatrix} \mathbf{y}^{(b)}[1] & \mathbf{y}^{(b)}[2] & \cdots & \mathbf{y}^{(b)}[j^{(b)}] \\ \vdots & \vdots & & \vdots \\ \mathbf{y}^{(b)}[i] & \mathbf{y}^{(b)}[i+1] & \cdots & \mathbf{y}^{(b)}[i+j^{(b)}-1] \end{bmatrix} \tag{3.12}$$

The number of Hankel rows 'i' is a tuning parameter and has to satisfy the condition ($i > n$), where n (also a parameter) is the order of the model. The number of columns 'j' should be sufficiently larger than 'i', so that the matrix can have as much dynamic information about the batch as possible. The pseudo-Hankel matrix concatenating data from all the N_T training batches is given by Equation (3.13).

$$\mathbf{Y}_{1|i} = \begin{bmatrix} \mathbf{Y}_{1|i}^{(1)} & \mathbf{Y}_{1|i}^{(2)} & \cdots & \mathbf{Y}_{1|i}^{(N_T)} \end{bmatrix} \tag{3.13}$$

This matrix $\mathbf{Y}_{1|i}$ thus contains information of all the training batches and elegantly handles the variable batch duration by aligning them with respect to a constant parameter (Hankel rows). The batches only differ in the number of columns depending on their duration.

The input pseudo-Hankel matrix $\mathbf{U}_{1|i}^{(b)}$ with input data (u) is similarly constructed. The subscript $1|i$ refers to the continuous measurements ranging from 1 to i that are stacked up in the first column. Both the input and output pesudo-Hankel matrix are stacked vertically to create a block Hankel matrix $\mathbf{H_1}$ given by Equation (3.14). $\mathbf{H_2}$ is similarly constructed where the subscript $i+1|2i$ refers to the elements in the first column (input or output pseudo-Hankel matrix) which are indexed from $i+1$ to $2i$. The matrix $\mathbf{H}$ is created by the vertical concatenation of matrices $\mathbf{H_1}$ and $\mathbf{H_2}$.

$$\mathbf{H}_1 = \begin{bmatrix} \mathbf{Y}_{1|i} \\ \mathbf{U}_{1|i} \end{bmatrix}; \quad \mathbf{H}_2 = \begin{bmatrix} \mathbf{Y}_{i+1|2i} \\ \mathbf{U}_{i+1|2i} \end{bmatrix}; \quad \mathbf{H} = \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{H}_2 \end{bmatrix} \tag{3.14}$$

The aim of this identification technique is to first generate a valid state vector trajectory and then to find the model matrices (A,B,C,D) using the state sequence and I/O data. A general state-output equation is first generated by repeated substitution of Equation (4.10) and given by Equation (3.15).

$$\mathbf{Y}_h = \mathbf{\Gamma_i}.\mathbf{X} + \mathbf{H}_t.\mathbf{U}_h \tag{3.15}$$

where, $\mathbf{Y}_h$ and $\mathbf{U}_h$ are the output and input Hankel matrices, $\mathbf{\Gamma_i}$ denotes the extended observability matrix and $\mathbf{H}_t$ represents the lower triangular block Toeplitz matrix and $\mathbf{X}$ is the consecutive state vector sequence. Detailed structure of these matrices is presented in [16].

For a b^{th} batch, a state vector sequence $\mathbf{X_2^{(b)}} = [\ \mathbf{x}^{(b)}[i+1]\ \ \mathbf{x}^{(b)}[i+2]\ \ \cdots\ \ \mathbf{x}^{(b)}[i+j]\ ]$ can be generated at the intersection of the row space of the matrix $\mathbf{H}$ by appropriate projections of matrices using concepts of linear algebra. Alternately, a more practical and robust approach of finding the intersection is to perform Singular Value Decomposition (SVD) on the matrix **H**. A detailed analysis of the approach is presented in [16] and has been omitted in this work for the sake of brevity.

The system matrices are next computed using least squares regression of the over determined set of equations given by Equation (3.16).

$$\begin{bmatrix} \mathbf{X_{out}} \\ \mathbf{y_{out}} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} . \begin{bmatrix} \mathbf{X_{pred}} \\ \mathbf{u_{pred}} \end{bmatrix} \tag{3.16}$$

The approach proposed in [21] accommodates multiple batches together and, thus, for N_T training batches the regression is performed using the matrices given by Equations (3.17) and (3.18).

$$\mathbf{X_{pred}} = \begin{bmatrix} \mathbf{X_{pred}}^{(1)} & \mathbf{X_{pred}}^{(2)} & \cdots & \mathbf{X_{pred}}^{(N_T)} \end{bmatrix}; \tag{3.17}$$

$$\mathbf{X_{out}} = \begin{bmatrix} \mathbf{X_{out}}^{(1)} & \mathbf{X_{out}}^{(2)} & \cdots & \mathbf{X_{out}}^{(N_T)} \end{bmatrix}; \tag{3.18}$$

where the predictor and outcome state variable vector for a batch b is respectively given by $\mathbf{X_{pred}}^{(b)} = [\ \mathbf{x}^{(b)}[i+2] \ \ \mathbf{x}^{(b)}[i+3] \ \ \cdots \ \ \mathbf{x}^{(b)}[i+j]\ ]$ and $\mathbf{X_{out}}^{(b)} = [\ \mathbf{x}^{(b)}[i+1] \ \ \mathbf{x}^{(b)}[i+2] \ \ \cdots \ \ \mathbf{x}^{(b)}[i+j-1]\ ]$. The output and input vectors are similarly created to accommodate for multiple batches.

The model order (n) and Hankel rows (i) are parameters which need to be specified/evaluated during the identification stage. Identifying the right order is a crucial step in all system identification methods as the predictive quality of the model largely depends on this choice. A cross validation methodology proposed in [1] is adopted to evaluate the same which utilizes the training data to figure the best model order according to the Mean Absolute Scaled Error (MASE) value given by Equation (3.19).

$$\textbf{MASE} = \frac{\sum_{t=1}^{T} |e_t|}{\frac{T}{T-1}\sum_{t=2}^{T} |Y_t - Y_{t-1}|}, \quad \text{where } e_t = Y_t - Y_t^p \tag{3.19}$$

Y_t and Y_t^p are the measured and predicted output data respectively at the t^{th} sampling instant. T is the range (typically duration of the batch) over which this error metric is calculated. The number of Hankel rows i has a constraint of $i > n$ and an optimum value can be evaluated from cross validation methodology. However, in this current application, only the number of states is considered as the tuning parameter and the number of Hankel rows is chosen as $i = n + 4$. The number of hankel rows was manually varied over a small range and didn't lead to any significant variation and hence was chosen in this fashion. Once the order and the number of Hankel rows are fixed, the identification procedure is employed to evaluate the system matrices. Next, the quality of the model is tested by making use of the validation set and is presented in the following subsection.

3.3.2 Model validation

The utility of a model in a MPC framework lies in its quality of imitating the process dynamics once initialized. To test the performance of a subspace based model for a batch process, the model is tested on validation data of a new batch. In [1], a detailed schematic is presented for the validation procedure of a parallel hybrid subspace model. The same approach is followed in this application and briefly illustrated below. This validation procedure primarily includes two steps:

1. **State estimation**: Subspace models with a state space structure uses its model states and future candidate inputs to be able to predict. These states, however, do not necessarily have a physical realization and thus for a fresh batch there is

no a priori information available about the initial conditions. The first part in the validation scheme makes use of a state estimator that estimates the states ($\hat{X}$) making use of the model parameters (A,B,C,D), the measured outputs (Y) and an initial guess of the state vector ($\hat{X}_0$). The estimated state make use of the output equation to generate the estimated outputs. This estimation step does not put any limitation on the type of estimation technique to be deployed and any approach like luenberger observer, kalman filtering, moving horizon estimator etc. can be appropriately chosen. In this application, a kalman filter is used as the tool for state estimation. State estimation is carried out till the point where the estimated outputs converge with the measured outputs over a significant amount of time. The subspace state at that mark is considered a usable estimate to be used as the initial state for subsequent model prediction.

2. **Model prediction**: This is the second step in the validation procedure where once the initial state is available, the model is allowed to predict (using its parameters) till the end of the batch. This aspect is particularly important not only in judging the quality of predictive models (done offline), but also in its usage inside the MPC (done online) where predictions are made to evaluate future state values, quality etc. and the current control actions are determined based on such forecast. The linear parallel hybrid model is used for that purpose and is presented in Section 3.3.3.

3.3.3 Predictive model in the MPC

A linear predictive model with a simple model structure is formulated as it provides several practical on-line implementation benefits over their non-linear counterpart. The parallel hybrid model [1] being non-linear in nature (owing to the first principles component) renders the MPC optimization as a non-convex optimization problem and is often time consuming and difficult to solve. A linear equivalent of the non-linear

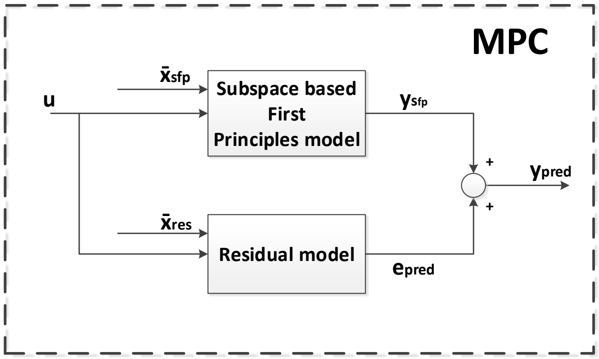


Figure 3.4: Schematic of the predictive model embedded in the MPC framework

component of the hybrid model is thus proposed where the first principles model is appropriately modeled by a linear surrogate model and then considered as a part of the hybrid framework. This is achieved by building an LTI subspace based model using I/O data generated by the mechanistic model and will be referred to as the subspace based first principles (sfp) from here onward. Thus, essentially the hybrid model structure used in the MPC framework consists of two linear subspace based models as shown in Figure 3.4 joined together in a parallel fashion. Two such linear hybrid frameworks are considered each having the sfp model as the common unit in their framework.

Remark 10. *It is important to note that the linear model (sfp) is not a linearization of a non-linear set of equations about any steady state operating point. Such an approximation would not work well in batch processes as the batch dynamics change significantly over the course of its completion. The proposed approach rather captures the non-linear time varying dynamics of the batch process by creating a linear invariant (LTI) model with a potentially high model order, i.e. the number of states in such models is typically higher than the number of states in the first principles model.*

- **Hybrid-1:** In this parallel hybrid framework, the residual considered is the difference between the measurement data (y) and the output of the sfp model (y_{sfp}) rather than the non-linear first principles model as used in [1]. Equation (3.20) presents the state-space model structure for the sfp model (x_{sfp} are the states, A_{sfp}, B_{sfp}, C_{sfp}, D_{sfp} are the identified model matrices and y_{sfp} is the output) and the one for the residual model (x_{res} are the states, A_{res}, B_{res}, C_{res}, D_{res} are the system matrices and e_{res} is the output). The final model output (y_{hm}) is the summation of the individual model outputs as given by Equation (3.21)

$$\begin{aligned} x_{sfp}[k+1] &= A_{sfp}x_{sfp}[k] + B_{sfp}u_{sfp}[k] \\ y_{sfp}[k] &= C_{sfp}x_{sfp}[k] + D_{sfp}u_{sfp}[k] \\ x_{res}[k+1] &= A_{res}x_{res}[k] + B_{res}u_{res}[k] \\ e_{res}[k] &= C_{res}x_{res}[k] + D_{res}u_{res}[k] \end{aligned} \tag{3.20}$$

$$y_{hm}[k] = y_{sfp}[k] + e_{res}[k] \tag{3.21}$$

- **Hybrid-2:** This architecture of this hybrid model is similar to the previous one and is only different with regard to the data used to train the residual model. In this case the residual model is built with historical inputs and error data between the process output (y) and the original (in this case non-linear) first principles model (y_{fp}).

Remark 11. *The first principles model considered in this work is 'flawed' by introducing structural as well as parametric uncertainty in the model. This situation of an imperfect first principles is quite common in practice for complex batch process modeling and thus motivates the use of an error model that corrects its predictions. A subspace model built with I/O data of such a first principles model can at best equal the predictions of this flawed model but will still have biases in its predictions when compared to the true process measurements. In the event a high fidelity mechanistic model is available that can predict the dynamics with good accuracy, an error model predicting the residuals won't be necessary. However, in the present instance, the* *residual model is in place to correct the error between the first principles model (or its linear surrogate form) and the process.*

3.3.4 Closed loop control

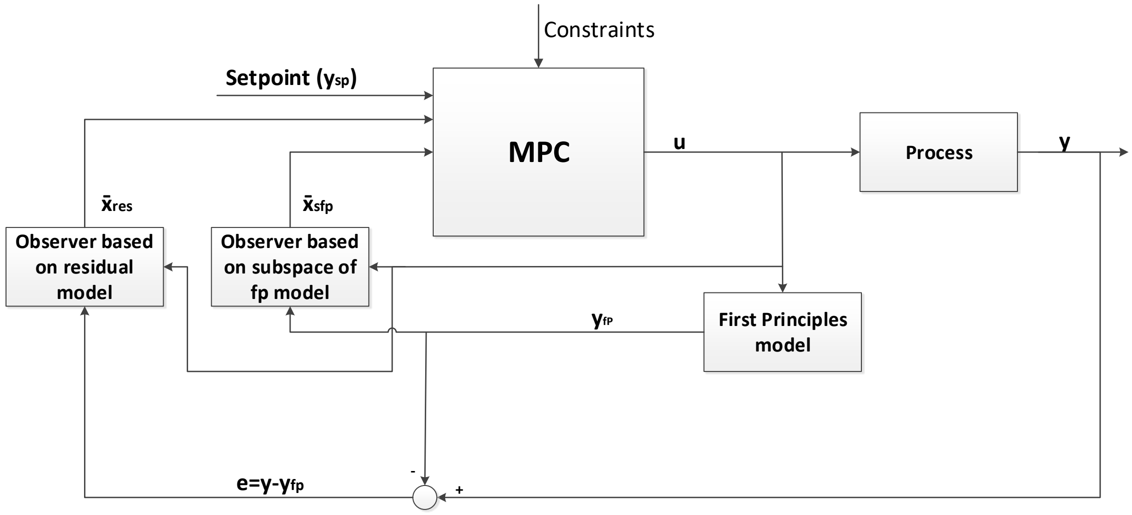


Figure 3.5: Schematic of the system under closed loop control

The schematic of the hybrid MPC enabled closed loop system is shown in Figure 3.5. In order to implement the MPC in real time, the non-linear first principles model is run online for a validation batch. It is imperative that such model equations can be solved within the sampling interval and necessitates the use of a model structure with more simplifying assumptions. The simplified ODE model is considered for this purpose in the current work. It is initiated with the same conditions as the true process (PDE model) and fed the same inputs. Two state estimators are run simultaneously to estimate the individual model states (pertaining to the two components of the linear hybrid model) to be fed to the MPC algorithm. The linear nature of the predictive model inside the MPC (along with linear constraints and linear/quadratic objective function) facilitates the optimization formulation be a linear or quadratic program which can be readily solved to find the optimum set of input values.

Remark 12. *There exist instances of MPC implementation using a parallel hybrid model (using a simplified first principles with neural network as the residual [42, 43], a first principles model with a PLS model [35, 39] etc). However, all these models have at least one non-linear component in their architecture, rendering the optimization a non-convex problem. This is an issue for systems where process dynamics change fast and optimum control actions need to be taken at smaller sampling intervals. The models involving ANN might further face challenges of finding the right parameters to prevent overfitting when the data available for training is limited and of low quality. On the other hand, subspace models can elegantly handle the overfitting issue due to the limited number of model parameters that need to be selected which is practically just one (the number of states). The design and implementation of such a linear hybrid model in a MPC framework is one of the contributions of the present manuscript.*

Remark 13. *The control problem considered in the current study is not of regulatory nature (driving a system to an equilibrium point) or a set-point tracking problem. Although, the linear parallel hybrid MPC would be very well suited for such an application, an economic objective (concerning the quality) is considered and its efficacy as an economic MPC demonstrated through simulation case studies.*

Remark 14. *A model monitoring mechanism embedded in the MPC framework in order to check the health of the model over a period of time is extremely important in all practical situations. One such monitoring scheme for subspace based MPC was presented in [44] where some performance metrics dictated the current health of the model suggesting the need for re-identification. The availability of first principle model for a parallel architecture will greatly aid to create such monitoring schemes and adequate metrics can be similarly created. This important aspect of model maintenance will be explored in future studies.*

3.4 Applications to the seeded batch crystallization

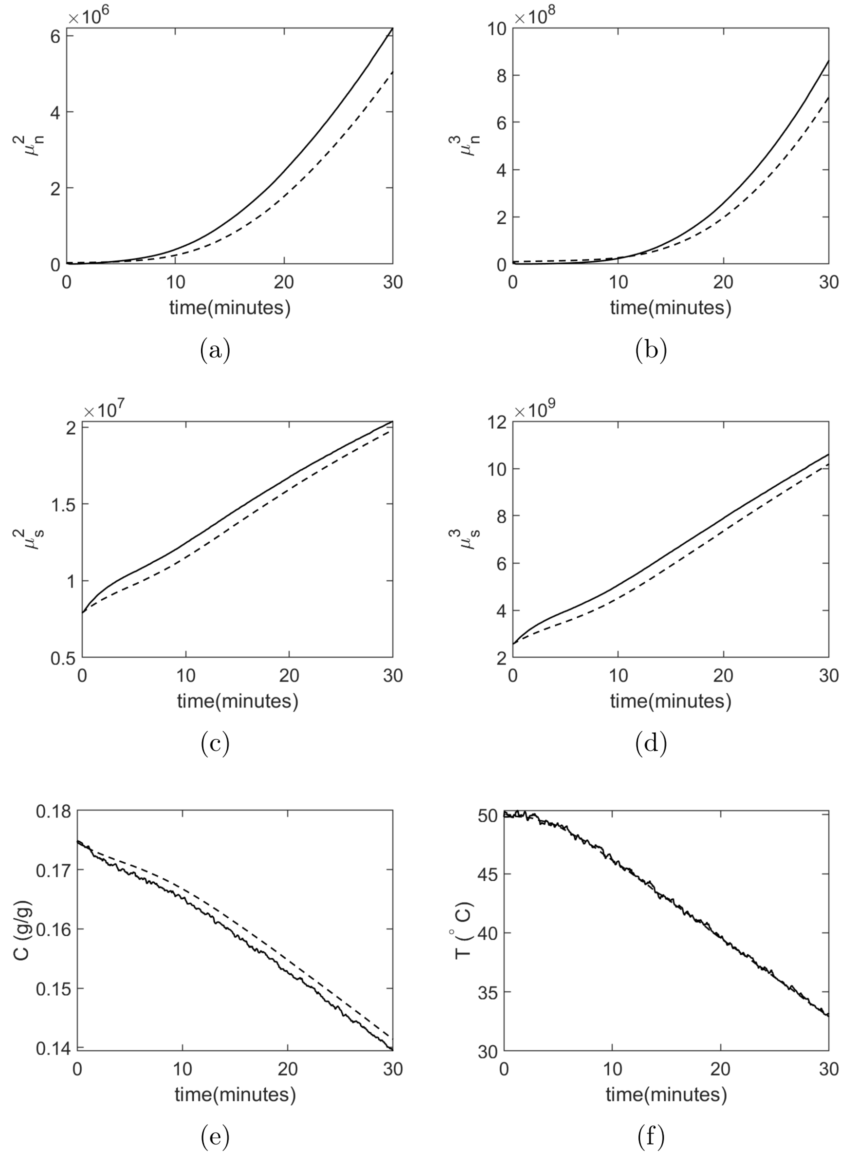
In this section, the proposed modeling and control methodology is applied to the seeded batch crystallization process. To this end, the development of the linear hybrid model is first described in Section 3.4.1, followed by the validation results for new batches where the proposed linear hybrid model is compared with other modeling approaches. The mathematical formulation related to the MPC optimization for the current problem is subsequently described in Section 3.4.2.

3.4.1 Hybrid data-driven modeling of seeded batch crystallizer

In this section, the development of the linear parallel hybrid model for the batch crystallizer is presented. The simpler ODE model is assumed to be available for the batch crystallizer and is considered as the first principles model. Some of the parameters in the nucleation and growth rate of this model are also considered to be erroneous, suggestive of the incorrect estimation of the parameters of the model and the mismatch is listed in Table 4.5. The first principles model predictions are compared against the process measurements and shown in Figure 3.6. In essence, this scenario reflects the situation where a first principles model of a process may be available in which the model captures the key characteristics but incorrectly predicts the variable values.

The outputs of the linear parallel hybrid model is considered the same as the first principles model (mentioned in Section 4.2.1). The two components of this framework and their development is discussed below:

- **Subspace based first principles (sfp) model** : A subspace model is trained using 40 batches of I/O data based on simulations by the ODE model. The dif-



(a)

(b)

(c)

(d)

(e)

(f)

Figure 3.6: Few candidate profiles generated by the PDE model (considered as process $-$) and the ODE model (considered as the first principles model $--$) are shown in the figures. Figures (a),(b) represent the profile of 2nd and 3rd moment due to nucleation; (c),(d) represent the profile of 2nd and 3rd moment due to growth; (e) and (f) show the concentration and temperature profile data for a particular batch

ferential equations are simulated using ODE45 in Matlab and the initial conditions for all the batches are kept similar to process (PDE model). The batch data is centered and scaled with respect to average trajectory value of a candidate batch before the identification step. With $n = 30$ (figured by cross-validation methodology[1]), the identification algorithm (as described in Section 3.3.1) is used to generate the model matrices and training states. The number of Hankel rows is considered to be 34 for the current application.

- **Residual model** : The residual model is developed according to the Hybrid-1 and Hybrid-2 modeling strategy as discussed in Section 3.3.3. For the Hybrid-1 case, training data comprises of the residual information $(y - y_{sfp})$ and input profiles (u) of the training batches. y_{sfp} data is generated by the sfp model using the states generated from the identification step. These states capture the dynamics of the training batches and the outputs for each batch are generated using the Equation (4.10). 61 Hankel rows were used to build the Hankel matrix and the identified residual model has a model order of 57 (determined by cross validation). The residual model for Hybrid-2 was similarly constructed but with residuals $(y - y_{fp})$ and having a model order of 60.

Once both the components of the framework is identified, the linear parallel hybrid model $f(y_{sfp}, (y - y_{sfp}), u)$ is validated on 15 fresh batches. A kalman filter is used in closed loop till the first 10 minutes of the run. The filter is initialized with a zero vector and the tuning parameters, process noise variance and measurement noise variance, are respectively chosen to be 0.001 and [0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.03 0.09 0.09 0.09]. Once the estimated outputs have converged with the process measurements, the model is allowed to predict from the 10^{th} minute till the end of batch. The MASE values are calculated for the model prediction duration only and the cumulative values for all 15 batches are recorded and presented in Table 3.1.

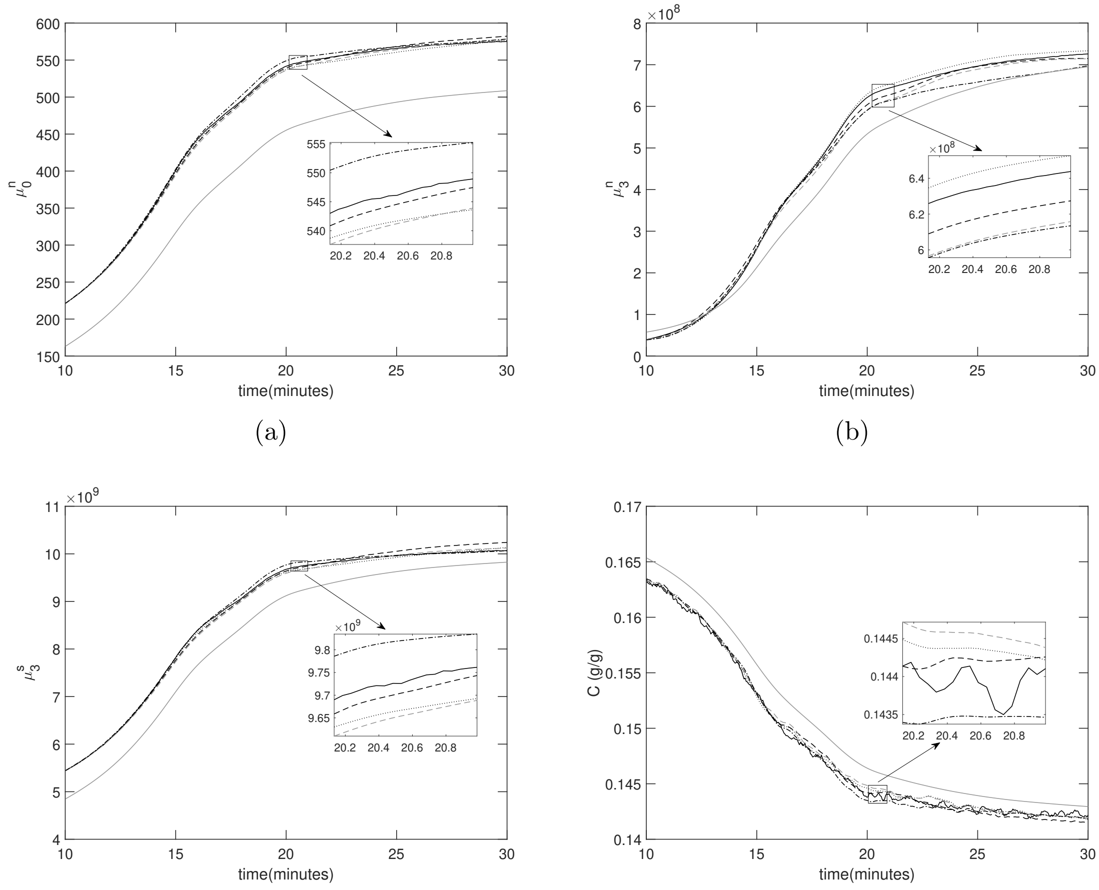
The predictions using the proposed approach is compared with the predictions of a non-linear parallel hybrid model[1] $f(y_{fp}, (y - y_{fp}), u)$, an only data based subspace model $f(y, u)$ and the first principles model $f(y_{fp}, u)$ and shown in Figure 3.7. All the identification parameters relevant to these models are tabulated in Table 3.9. The kalman tuning parameters for these models are chosen to be the same as the linear parallel hybrid model. The MASE values of the different models are presented in Table 3.1 and it can be seen that the linear parallel hybrid model has lower error values compared to the other models. The proposed framework only fairs second to the non-linear parallel hybrid model and is expcted as this linear model is only an approximation of its non-linear counterpart. The high quality prediction of the linear parallel hybrid model makes it an ideal candidate for a predictive model to be used inside an MPC for closed loop control and is demonstrated in the subsequent sections.

Table 3.1: Cumulative MASE Error Analysis

Model Type	Cumulative MASE (15 batches)
Non-linear Hybrid Model	347.25
Hybrid-1	444.17
Hybrid-2	446.24
Subspace Identification	561.19
First Principles	5578.9

Remark 15. *One of the most important characteristics of this parallel framework is that it allows the flexibility of working with a reduced order mechanistic model with good enough predictive capabilities where the residual model corrects its biases in predictions. In [1], the first principles model component of the hybrid framework was assumed to have faulty parameter values and had the same model structure as the complex process model (used for generating data mimicking the real measurements). In this current example, the case of using a simplified first principles model is taken into consideration. This reduced order model is quite tractable and easy to solve and thus, has significant advantages in real time optimization and control implementations.*

Remark 16. *Standard subspace algorithms cannot directly handle the missing data problem in the historical database. Such issues are quite a common occurrence while handling data from industrial processes. In a recent study [45], these practical concerns are addressed and it proposes an methodology that makes use of NIPALS algorithm*



(a)

(b)

(c)

(d)

Figure 3.7: Predicted profiles of few outputs for a validation batch generated by 5 different modeling approaches are shown in the figures : first principles (gray −), subspace identification (black−.−), non-linear hybrid (black :), Hybrid-1 (black −−), Hybrid-2 (gray −−). The process data generated by the PDE model is represented by the solid (black −) line. Figures (a),(b) represent the profile of zeroth and third moment due to nucleation; (c) represents the profile of 3rd moment due to seeding; (d) shows the concentration profile

to handle the missing values during one of the model identification stages of subspace identification. The algorithm was compared with mean replacement and linear interpolation techniques and its superiority was demonstrated. The use of such robust techniques to identify LTI state space matrices for such scenarios can be very well extended to the hybrid modeling approach proposed in this work.

3.4.2 Formulation of the MPC optimization problem

The MPC optimization formulation adopted for this case study is presented in this section. To handle the plant model mismatch, five sequential layers of optimization are formulated for the current problem and the subsequent layers are executed only if the preceding one fails to find a feasible solution. A linear program (LP) is formulated taking advantage of the linear nature of the model along with a linear objective and constraints. The output vector (y) mentioned in Section 4.2.1 is notated by Equation (3.22) for ease of usage. Continuous measurement of these outputs is assumed to be available.

$$y = \begin{bmatrix} y_1 & y_2 & y_3 & y_4 & y_5 & y_6 & y_7 & y_8 & y_9 & y_{10} & y_{11} & y_{12} \end{bmatrix}^T \tag{3.22}$$

The outputs are measured and the inputs are calculated at time instances indexed by l $(1, 2, \cdots, l_f)$. l_f is the final sampling instance given by $l_f = t_f/\Delta$, where t_f is the final time and Δ the sampling interval. k is a variable used for indexing time instances for MPC predictions and for any sampling instance l, the optimum input trajectory vector U_f consists of elements indexed from 0 to $l_f - l$. H_c is termed as the constraint horizon over which certain constraints are enforced. The states $\hat{X}(l)$ are obtained from the corrected states at the last time step $l - 1$. The computations involving state estimation and generation of the optimum input policy are done during the sampling interval (Δ) between samples l and $l - 1$, so that at the instance l, the first move of optimum input trajectory U_f is implemented on the process/PDE model to obtain future measurements. The measurements available at l are used to correct the state estimates $\hat{X}(l)$ and the same steps are repeated every Δ.

The different layers are described below:

- **Case A: Primary optimization layer** The optimization problem is formulated similar to [24] and is presented in Equation (3.24). The formulation although uses mean and centered scaled variables, is written in this manuscript in terms of original variables for sake of simplicity. The mean centered and scaled variables are related to the original variables by the following relations :

$$\begin{aligned} y_i &= \bar{y}_i * \sigma_i + \mu_i; \;\; i = 1, 2, \cdots, 12 \\ u &= \bar{u} * \sigma_u + \mu_u \end{aligned} \tag{3.23}$$

where $\bar{u}, \bar{y}_i$ are respectively the mean centered and scaled value of the input and the i^{th} output. σ_u, μ_u are the mean and standard deviation value used for the input and σ_i, μ_i are the ones used for the corresponding output.

The optimal MPC input trajectory is calculated using the formulation as follows:

$$\begin{aligned} \min_{U_f} \;\; & y_4[l_f - l] \\ s.t. \;\; & T_{j,min} \leq u_f[k] \leq T_{j,max}, \;\; \forall \;\; 0 \leq k \leq l_f - l \\ & y_{11}[k] + \epsilon_1 \leq y_9[k] \leq y_{12}[k] + \epsilon_2, \;\; \forall \;\; k \leq H_c \\ & |u_f[0] - u[l-1]| \leq \delta, \\ & |u_f[k] - u_f[k-1]| \leq \delta, \; \forall \;\; 1 \leq k \leq l_f - l \\ & y_8[l_f - l] \geq \gamma + \epsilon_3, \\ & x[0] = \hat{x}[l] \\ & x_{sfp}[k+1] = A_{sfp} x_{sfp}[k] + B_{sfp} u[k] \\ & e_{sfp}[k] = C_{sfp} x_{sfp}[k] + D_{sfp} u[k] \\ & x_{res}[k+1] = A_{res} x_{res}[k] + B_{res} u[k] \\ & y_{res}[k] = C_{res} x_{res}[k] + D_{res} u[k] \\ & y_{hm}[k] = y_{sfp}[k] + e_{res}[k], \; \forall \;\; 0 \leq k \leq l_f - l \end{aligned} \tag{3.24}$$

The components of this optimization formulation are briefly summarized below:

1. **MPC objective:** The aim of the optimization is to reduce the amount of fines in the product and is achieved by minimizing y_4 at batch end, the third moment due to nucleation which is directly correlated to the volume of the crystals formed during the nucleation phase.

2. **Constraints:**

   (a) *Output constraints*:

      i. Concentration constraint: y_9 is the solution concentration and should be between the saturation concentration (y_{11}) and metastable concentration (y_{12}) at all instances throughout the batch.

      ii. Quality constraint: y_8 is the product quality and should be above a certain desired value γ at batch end.

   (b) *Input constraints*:

      i. Saturation constraints on the inputs (u) restrict them to be between bounds [$T_{j,min}$ $T_{j,max}$].

      ii. Rate of change of the manipulated input, $\frac{du}{dt}$, over the prediction horizon and also between the current and last implemented input is chosen to be a value less than or equal to δ to capture physical limitations on the control input.

   (c) *Model equations:* The state-space model equations dictate the evolution of the subspace states and the final model output (y_{hm}) is the summation of individual outputs (y_{sfp} and e_{res}) of the two models that comprise the hybrid framework.

ϵ_1 and ϵ_2 are the tuning parameters on the reaction concentration constraint which accommodates for the plant-model mismatch. Similarly, ϵ_3 is the tuning parameter for the quality constraint.

The p-step ahead model prediction can be written by the equation below by iterative substitution of Equation (3.20) and is given by equation Equation (3.25).

$$Y[p] = C_{sfp}{A_{sfp}}^p x_{sfp}[l] + \phi_{sfp} U_f^p + C_{res}{A_{res}}^p x_{res}[l] + \phi_{res} U_f^p \tag{3.25}$$

where,

$$\begin{aligned}
\phi_{sfp} &= \begin{bmatrix} C_{sfp}{A_{sfp}}^{p-1}B_{sfp} & C_{sfp}{A_{sfp}}^{p-2}B_{sfp} & \dots & C_{sfp}B_{sfp} & D_{sfp} \end{bmatrix} \\
\phi_{res} &= \begin{bmatrix} C_{res}{A_{res}}^{p-1}B_{res} & C_{res}{A_{res}}^{p-2}B_{res} & \dots & C_{res}B_{res} & D_{res} \end{bmatrix} \\
U_f^p &= \{u_f[0], u_f[1], \cdots, u_f[p]\}
\end{aligned}$$

The value of the i^{th} output p steps ahead in the future can be as follows :

$$\begin{aligned}
Y_i[p] &= L_i Y[p], \\
\text{where,} \quad L_i &\triangleq \begin{bmatrix} 0_{1\times(i-1)} & 1 & 0_{1\times(12-i)} \end{bmatrix} \;\; \forall\, i = 1, \dots, 12
\end{aligned} \tag{3.26}$$

The terminal constraint on the seeded crystals is given by:

$$\begin{aligned}
& L_8 Y[l_f - l] \geq \gamma \\
\implies & -L_8 \phi_{l_f - l} U_f^{l_f - l} \leq -\gamma + L_8 C_{sfp}{A_{sfp}}^{l_f - l} x_{sfp}[0] + L_8 C_{res}{A_{res}}^{l_f - l} x_{res}[0]
\end{aligned} \tag{3.27}$$

Constraints on the input move is formulated as follows :

$$| \begin{bmatrix} u[l] - u_f[0] \\ u_f[0] - u_f[1] \\ \vdots \\ u_f[l_f - l - 1] - u_f[l_f - l] \end{bmatrix} | \leq \Delta_u \tag{3.28}$$

$$\implies \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 \\ 0 & 1 & -1 & \cdots & 0 \\ & \ddots & \ddots & \ddots & \\ 0 & \cdots & 0 & 1 & -1 \\ -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \cdots & 0 \\ & \ddots & \ddots & \ddots & \\ 0 & \cdots & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} u[l] \\ U_f^{l_f - l} \end{bmatrix} \leq \Delta_u \tag{3.29}$$

where Δ_u is a vector of δ.

Further, the constraints on the concentration, $y_{11} \leq y_9$ can be formulated as

$$\begin{bmatrix} L_{11} - L_9 \\ \vdots \\ L_{11} - L_9 \end{bmatrix} \circ \Pi_{sfp} U_{(H_c-1)} + \begin{bmatrix} L_{11} - L_9 \\ \vdots \\ L_{11} - L_9 \end{bmatrix} \circ \Pi_{res} U_{(H_c-1)} \leq - \begin{bmatrix} L_{11} - L_9 \\ \vdots \\ L_{11} - L_9 \end{bmatrix} \circ \begin{bmatrix} C_{sfp} \\ C_{sfp} A_{sfp} \\ \vdots \\ C_{sfp} {A_{sfp}}^{(H_c-1)} \end{bmatrix} x_{sfp}[0] \tag{3.30}$$

$$- \begin{bmatrix} L_{11} - L_9 \\ \vdots \\ L_{11} - L_9 \end{bmatrix} \circ \begin{bmatrix} C_{res} \\ C_{res} A_{res} \\ \vdots \\ C_{res} {A_{res}}^{(H_c-1)} \end{bmatrix} x_{res}[0]$$

where, Π_{sfp} is a Toeplitz matrix given by

$$\Pi_{sfp} = \begin{bmatrix} D_{sfp} & 0 & 0 & \cdots & 0 \\ C_{sfp} B_{sfp} & D_{sfp} & 0 & \cdots & 0 \\ C_{sfp} A_{sfp} B_{sfp} & C_{sfp} B_{sfp} & D_{sfp} & \cdots & 0 \\ & \ddots & \ddots & \ddots & \\ C_{sfp} {A_{sfp}}^{(H_c-2)} B_{sfp} & \cdots & C_{sfp} A_{sfp} B_{sfp} & C_{sfp} B_{sfp} & D_{sfp} \end{bmatrix} \tag{3.31}$$

Π_{res} is a similar matrix made with the matrices $A_{res}, B_{res}, C_{res}, D_{res}$. The constraint

$y_9 \leq y_{12}$ can be similarly written in the following fashion:

$$\begin{bmatrix} L_9 - L_{12} \\ \vdots \\ L_9 - L_{12} \end{bmatrix} \circ \Pi_{sfp} U_{(H_c-1)} + \begin{bmatrix} L_9 - L_{12} \\ \vdots \\ L_9 - L_{12} \end{bmatrix} \circ \Pi_{res} U_{(H_c-1)} \leq - \begin{bmatrix} L_9 - L_{12} \\ \vdots \\ L_9 - L_{12} \end{bmatrix} \circ \begin{bmatrix} C_{sfp} \\ C_{sfp} A_{sfp} \\ \vdots \\ C_{sfp} {A_{sfp}}^{(H_c-1)} \end{bmatrix} x_{sfp}[0] \tag{3.32}$$

$$- \begin{bmatrix} L_9 - L_{12} \\ \vdots \\ L_9 - L_{12} \end{bmatrix} \circ \begin{bmatrix} C_{res} \\ C_{res} A_{res} \\ \vdots \\ C_{res} {A_{res}}^{(H_c-1)} \end{bmatrix} x_{res}[0]$$

- **Case B: Second optimization layer**

  In this layer, the tuning parameters on the concentration constraint, ϵ_1 and ϵ_2 are kept as zero and all other aspects are kept similar to the primary layer.

- **Case C: Third optimization layer**

  In this layer, the quality constraint is removed and an additional term which maximizes the quality at batch end is added to the objective function having the form as shown below. The two terms in the objective function are given different weights in accordance to their importance in the optimization formulation.

$$\min_{U_f} \; 1000 * y_4[l_f - l] - 500 * y_8[l_f - l] \tag{3.33}$$

- **Case D: Fourth optimization layer**

  In this layer, the tuning parameters (ϵ_1 and ϵ_2) from the concentration constraint, and the constraint on the rate of change of input over the prediction horizon are removed. The rest is kept similar to the primary layer.

- **Case E: Fifth optimization layer**

  In this layer, only the constraint involving the rate of change of input between the last implemented input and the current step is kept in the formulation. The objective function aims to maximize the difference between the reaction concentration C and the saturation concentration C_s at each time step over the prediction horizon, represented mathematically as $\min\limits_{U_f} \quad -\sum_{k=0}^{l_f-l}(y_9[k] - y_{11}[k])$, with the model equations same as the primary layer, and the rate of input change constraint represented as $|u_f[k] - u_f[k-1]| \leq \delta, \;\; \forall \;\; 1 \leq k \leq l_f - l$.

3.5 Closed loop simulation results

The optimization formulation presented in the previous section is a linear program and is solved in MATLAB using the linprog function. The parameters related to the optimization are tabulated in Table 3.10. The tuning parameters used to tighten the constraints to accommodate for the plant-model mismatch is provided in Equation (3.34). The constraint horizon (H_c) for the reaction concentration constraint is considered to be of 25 time steps for $l_f - l \geq 25$ and due to the shrinking nature of the prediction horizon, this constraint is active for $l_f - l$ time steps when $l_f - l < 25$.

The two different hybrid model MPC are compared with an only data based subspace MPC for 100 validation batches to compare the quality of the different control strategies. The initial process conditions are kept the same in the three strategies for a fair comparison.

A PI based controller (settings provided in Table 4.7) is run for the first 10 minutes in all the cases to obtain a good estimate of the initial state for the subspace based models, and the MPC is run online from that point onward till the end of the batch. In case of both the hybrid based MPC's, the ODE model is run online to generate the first principle model outputs using the same initial conditions as the process. The state estimators are appropriately run in closed loop and use the current inputs, measurements and first principle model outputs to generate states needed for MPC calculations at each sampling interval. Kalman filter was used as the state estimator and the tuning parameters are presented in Table 3.11.

Quality is denoted by the value of third moment due to seeding (μ_s^3) at batch end and three different cases are considered where the quality requirement is sequentially incremented (1.6e9, 2.6e9, 3e9 respectively for the three cases) on top of the base value 8.3301e9. These increments in quality are representative of instances where market conditions may dictate a different product, and a rapid change in process operation to meet the new quality requirements without running extensive experiments and re-identifying the model. The solution concentration profiles along with the implemented input trajectories and final crystal size distribution for a particular batch (for the cases 2.6e9 and 3.0e9) are presented in Figure 3.8 and Figure 3.10. The batches were run under closed loop control of each of the three control strategies and the histograms presented in Figure 3.9 and Figure 3.11 show the distribution of the amount of fines at batch end for all of the 100 batches considered. In order to compare the different control strategies, the mean of the objective function value (MOV) or the volume of fines at batch end given by μ_n^3 averaged over 100 batches is presented in Table 3.2 for all the three quality cases. The average was calculated using data from batches that were successful in obtaining the desired quality at batch end.

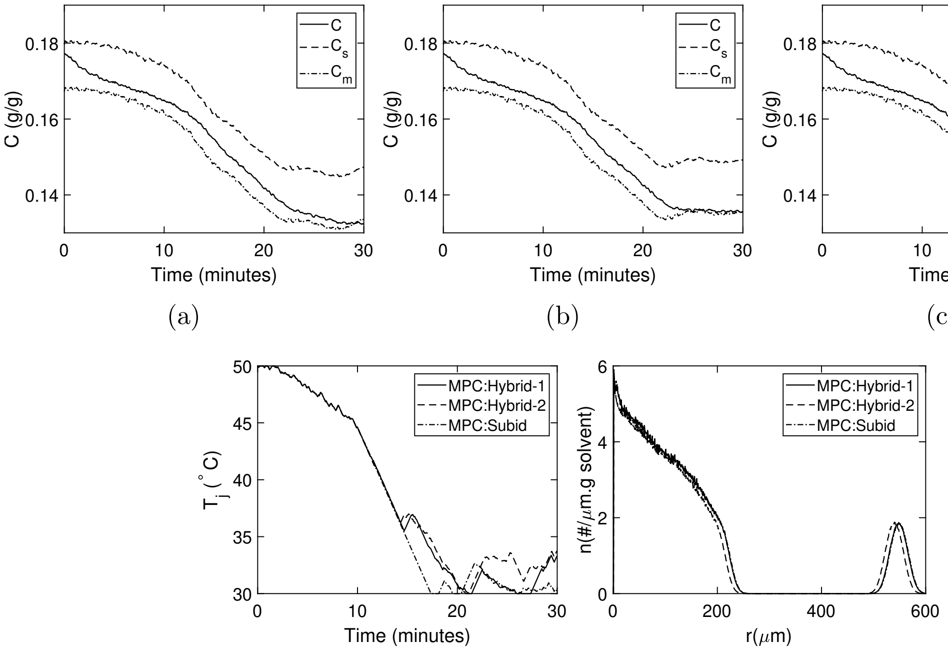
High purity products are manufactured in batch processes where tight quality requirements are an absolute necessity, and thus robustness to failures is the one of the most important criteria in deciding the efficacy of the control strategy. From Table 3.2, it can be seen that there is only 1 failed batch for the Hybrid-2 case and no failed batches in the case of Hybrid-1 and subspace strategy considering the 1.6e9 quality case. The Hybrid-2 seem to perform the best followed by the subspace and Hybrid-1 with respect to minimizing the volume of fines in the final product. However, with the increase in quality requirement the Hybrid-1 outperforms the only subspace based strategy. The number of failed batches is also the minimal for Hybrid-1 as compared to Hybrid-2 and at par with the subspace based MPC. This advantage in performance of Hybrid-1 over Hybrid-2 is logical as Hybrid-2 is rather a naive construction of the parallel framework, trained with residuals between process and first principles instead of process and sfp model (as in the case of Hybrid-1). Since the linearized first principles or sfp model is common in both the hybrid architectures, the error model of Hybrid-1 constructed with residuals $(y - y_{sfp})$ is theoretically more appropriate in correcting the shortcomings of the sfp model and is also well supported by the results. Thus, only Hybrid-1 is further considered for comparison studies. The advantages of hybrid MPC over subspace based MPC is evident not only from the MOV values tabulated in Table 3.2 but also from the histograms in Figure 3.9, Figure 3.11, where the distribution clearly shows a significantly higher number of batches have lower volume of fines describing a much smaller spread under the closed loop control of linear parallel hybrid based MPC as compared to the batches under subspace based MPC. It can be seen from Table 3.3 that the variance of fines and final product quality is also lower in the case of the Hybrid-1 MPC, adding to its advantages over the only data-based subspace MPC.

Table 3.2: Mean of the objective function value (MOV) and failed batch statistics over 100 batches for 3 different quality requirements

MPC Type	Quality: 1.6e9		Quality: 2.6e9		Quality: 3e9	
	MOV	Failed batch	MOV	Failed batch	MOV	Failed batch
Hybrid-1	$1.2326e+09$	0	$1.1457e+09$	4	$1.2218e+09$	11
Hybrid-2	$1.1876e+09$	1	$1.1400e+09$	8	$1.2452e+09$	17
Subid	$1.2009e+09$	0	$1.2284e+09$	4	$1.2788e+09$	11

Table 3.3: Variance of fines and the final quality at batch end of the successful batches

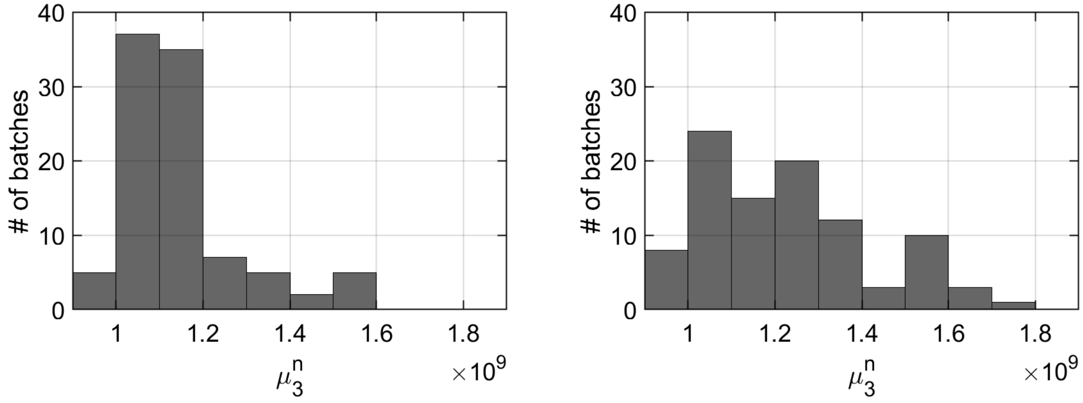
MPC Type	Quality: 1.6e9		Quality: 2.6e9		Quality: 3e9	
	Var(μ_n^3)	Var(μ_s^3)	Var(μ_n^3)	Var(μ_s^3)	Var(μ_n^3)	Var(μ_s^3)
Hybrid-1	$5.0041e+16$	$2.7881e+17$	$17.989e+15$	$8.9099e+16$	$3.2054e+15$	$9.8035e+15$
Subid	$5.5383e+16$	$3.2469e+17$	$36.293e+15$	$18.225e+16$	$18.298e+15$	$80.504e+15$



(a) (b) (c)

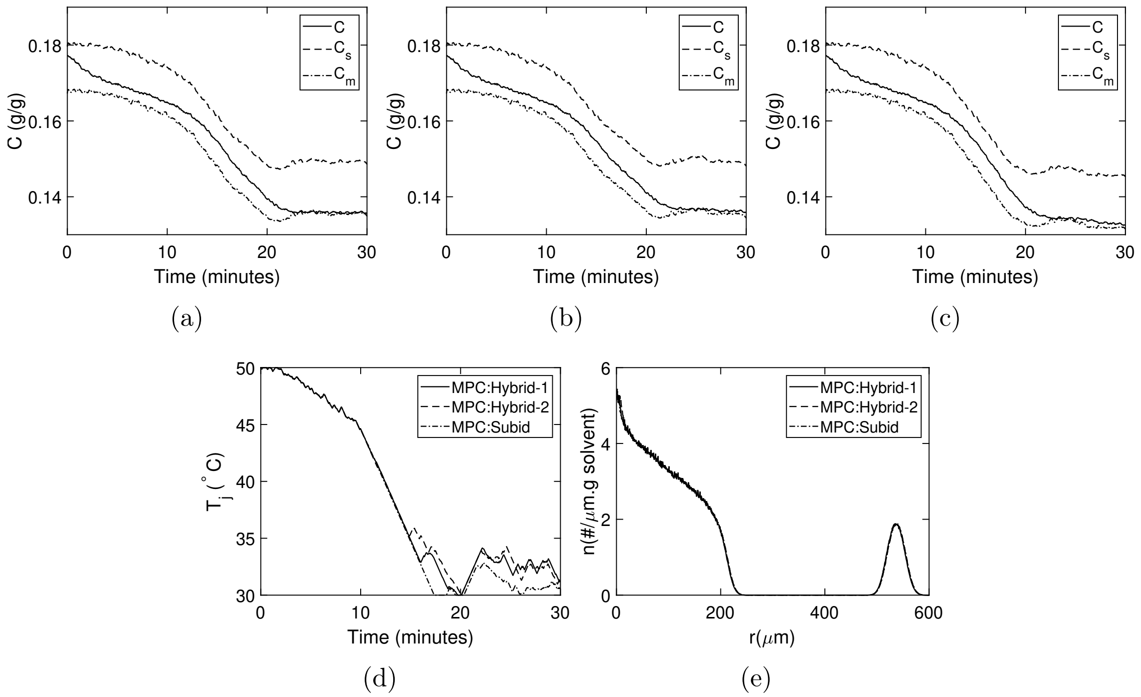
(d) (e)

Figure 3.8: Closed loop validation results for Quality 2.6e9 case: Figure (a) represent the solution concentration profile (−) along with saturation concentration (−−) and metastable concentration (−.) of a candidate batch using Hybrid-1 based MPC; Figures (b) and (c) represent the same profile using Hybrid-2 and Subid; Figure (d) represents the input trajectory generated by the MPC and Figure (e) represents the crystal size distribution at batch end using the Hybrid-1 (−), Hybrid-2 (−−) and Subid (−.) MPC



(a) (b)

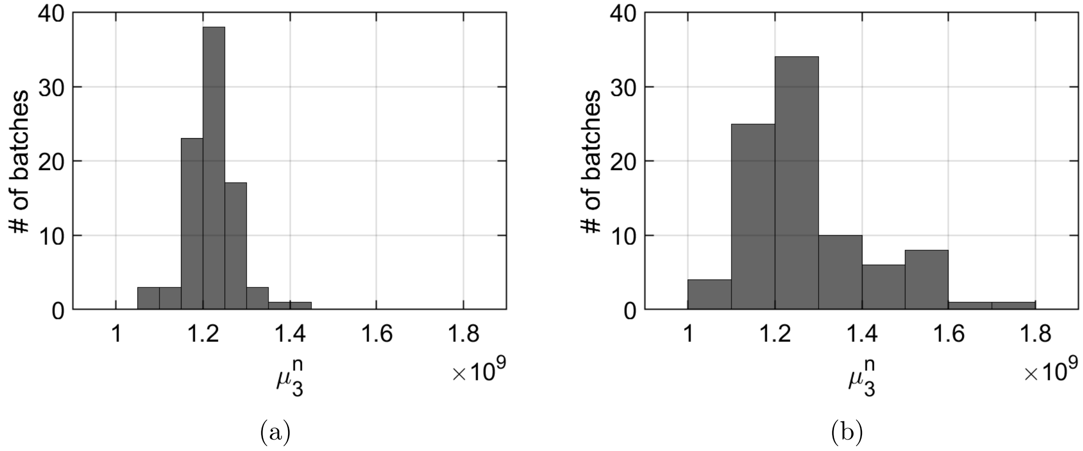
Figure 3.9: Distribution of μ_n^3 at batch end of successful batches for the closed loop cases: (a) Hybrid-1 (b) Subid, for the quality 2.6e9 case



(a) (b) (c)

(d) (e)

Figure 3.10: Closed loop validation results for Quality 3.0e9 case: Figure (a) represent the solution concentration profile (−) along with saturation concentration (−−) and metastable concentration (−.) of a candidate batch using Hybrid-1 based MPC; Figures (b) and (c) represent the same profile using Hybrid-2 and Subid; Figure (d) represents the input trajectory generated by the MPC and Figure (e) represent the crystal size distribution at batch end using the Hybrid-1 (−), Hybrid-2 (−−) and Subid (−.) MPC



(a) (b)

Figure 3.11: Distribution of μ_n^3 at batch end of successful batches for the closed loop cases: (a) Hybrid-1 (b) Subid, for the quality 3.0e9 case

3.6 Conclusions

A novel MPC utilizing a linear parallel subspace based hybrid model for controlling the volume of fines during a batch crystallization process is presented in this work. To this end, the superior predictive ability of this model in capturing the batch dynamics is first demonstrated and then an optimization framework embedding this model is proposed. This control strategy, through simulations of different quality based cases, has been shown to perform better than a purely data driven subspace model based one.

Supporting Information

Seeded batch crystallizer parameters; Seeded batch crystallizer parameter values and simulation settings; Erroneous parameters in the first principles model; Batch initial conditions and noise parameters; Model identification database summary; Model and parameters; Optimization parameters; MPC tuning parameter values; Kalman tuning parameters used in the state estimation during MPC implementation for the 3 different MPC's. This information is available free of charge via the Internet at http://pubs.acs.org/.

Acknowledgments

Financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC) through the Engage Grant, and the McMaster Advanced Control Consortium (MACC) is gratefully acknowledged.

Bibliography

[1] Debanjan Ghosh, Emma Hermonat, Prashant Mhaskar, Spencer Snowling, and Rajeev Goel. Hybrid modeling approach integrating first-principles models with subspace identification. *Industrial & Engineering Chemistry Research*, 58(30):13533–13543, 2019.

[2] Daniele Semino and W.Harmon Ray. Control of systems described by population balance equations—ii. emulsion polymerization with constrained control action. *Chemical Engineering Science*, 50(11):1825 – 1839, 1995.

[3] Michael J. Kurtz, Guang-Yan Zhu, Abdelqader Zamamiri, Michael A. Henson, and Martin A. Hjortsø. Control of oscillating microbial cultures described by population balance models. *Industrial & Engineering Chemistry Research*, 37(10):4059–4070, 1998.

[4] Dan Shi, Nael H. El-Farra, Mingheng Li, Prashant Mhaskar, and Panagiotis D. Christofides. Predictive control of particle size distribution in particulate processes. *Chemical Engineering Science*, 61(1):268 – 281, 2006. Advances in population balance modelling.

[5] Timothy Chiu and Panagiotis D. Christofides. Nonlinear control of particulate processes. *AIChE Journal*, 45(6):1279–1297, 1999.

[6] D. Bonvin, B. Srinivasan, and D. Hunkeler. Control and optimization of batch processes. *IEEE Control Systems Magazine*, 26(6):34–45, 2006.

[7] Jörg Worlitschek and Marco Mazzotti. Model-based optimization of particle size distribution in batch-cooling crystallization of paracetamol. *Crystal Growth & Design*, 4(5):891–903, 2004.

[8] Joseph Sang-II Kwon, Michael Nayhouse, Gerassimos Orkoulas, and Panagiotis D. Christofides. Enhancing the crystal production rate and reducing polydispersity in continuous protein crystallization. *Industrial & Engineering Chemistry Research*, 53(40):15538–15548, 2014.

[9] Jesus Flores-Cerrillo and John F. MacGregor. Control of particle size distributions in emulsion semibatch polymerization using mid-course correction policies. *Industrial & Engineering Chemistry Research*, 41(7):1805–1814, 2002.

[10] Jesus Flores Cerrillo and John F. Macgregor. Latent variable mpc for trajectory tracking in batch processes. *Journal of Process Control*, 15(6):651–663, 2005.

[11] Masoud Golshan, John F. MacGregor, Mark-John Bruwer, and Prashant Mhaskar. Latent variable model predictive control (lv-mpc) for trajectory tracking in batch processes. *Journal of Process Control*, 20(4):538 – 550, 2010.

[12] S. Aumi, B. Corbett, P. Mhaskar, and T. Clarke-Pringle. Data-based modeling and control of nylon-6, 6 batch polymerization. *IEEE Transactions on Control Systems Technology*, 21(1):94–106, 2013.

[13] B. Corbett, B. Macdonald, and P. Mhaskar. Model predictive quality control of polymethyl methacrylate. *IEEE Transactions on Control Systems Technology*, 23(2):687–692, March 2015.

[14] Xia Yu, Kamuran Turksoy, Mudassir Rashid, Jianyuan Feng, Nicole Hobbs, Iman Hajizadeh, Sediqeh Samadi, Mert Sevil, Caterina Lazaro, Zacharie Maloney, Elizabeth Littlejohn, Laurie Quinn, and Ali Cinar. Model-fusion-based online glucose concentration predictions in people with type 1 diabetes. *Control Engineering Practice*, 71:129 – 141, 2018.

[15] S. Joe Qin. An overview of subspace identification. *Computers & Chemical Engineering*, 30(10):1502 – 1513, 2006. Papers form Chemical Process Control VII.

[16] Vandenberghe L Vandewalle J. Moonen M, Demoor B. Online and off-line identication of linear state-space models international journal of control. *International Journal of Control*, pages 49:219–232, 1989.

[17] P. Van Overschee and B. De Moor. Two subspace algorithms for the identification of combined deterministic-stochastic systems. In *[1992] Proceedings of the 31st IEEE Conference on Decision and Control*, pages 511–516 vol.1, Dec 1992.

[18] Peter Van Overschee and Bart De Moor. A unifying theorem for three subspace system identification algorithms. *Automatica*, 31(12):1853 – 1864, 1995. Trends in System Identification.

[19] Ramesh Kadali, Biao Huang, and Anthony Rossiter. A data driven subspace approach to predictive controller design. *Control Engineering Practice*, 11(3):261 – 278, 2003. Advances in Automotive Control.

[20] Nima Danesh Pour, Biao Huang, and Sirish L. Shah. Subspace approach to identification of step-response model from closed-loop data. *Industrial & Engineering Chemistry Research*, 49(18):8558–8567, 2010.

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

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

[23] Abhinav Garg, Brandon Corbett, Prashant Mhaskar, Gangshi Hu, and Jesus Flores-Cerrillo. Subspace-based model identification of a hydrogen plant startup dynamics. *Computers & Chemical Engineering*, 106:183 – 190, 2017. ESCAPE-26.

[24] Abhinav Garg and Prashant Mhaskar. Subspace identification-based modeling and control of batch particulate processes. *Industrial & Engineering Chemistry Research*, 56(26):7491–7502, 2017.

[25] Abhinav Garg, Felipe P.C. Gomes, Prashant Mhaskar, and Michael R. Thompson. Model predictive control of uni-axial rotational molding process. *Computers & Chemical Engineering*, 121:306 – 316, 2019.

[26] Abhinav Narasingam and Joseph Sang-Il Kwon. Application of koopman operator for model-based control of fracture propagation and proppant transport in hydraulic fracturing operation. *Journal of Process Control*, 91:25 – 36, 2020.

[27] Babu Joseph and Frieda Wang Hanratty. Predictive control of quality in a batch manufacturing process using artificial neural network models. *Industrial & Engineering Chemistry Research*, 32(9):1951–1961, 1993.

[28] Luis Alberto Paz Suárez, Petia Georgieva, and Sebastião Feyo de Azevedo. Nonlinear mpc for fed-batch multiple stages sugar crystallization. *Chemical Engineering Research and Design*, 89(6):753 – 767, 2011.

[29] Svante Wold, Nouna Kettaneh-Wold, and Bert Skagerberg. Nonlinear pls modeling. *Chemometrics and Intelligent Laboratory Systems*, 7(1):53 – 65, 1989. Proceedings of the First Scandinavian Symposium on Chemometrics.

[30] S.J. Qin and T.J. McAvoy. Nonlinear pls modeling using neural networks. *Computers & Chemical Engineering*, 16(4):379 – 391, 1992. Neutral network applications in chemical engineering.

[31] Dimitris C. Psichogios and Lyle H. Ungar. A hybrid neural network-first principles approach to process modeling. *AIChE Journal*, 38(10):1499–1511, 1992.

[32] Mohammad Anwar Hosen, Mohd Azlan Hussain, and Farouq S. Mjalli. Control of polystyrene batch reactors using neural network based model predictive control

(nnmpc): An experimental investigation. *Control Engineering Practice*, 19(5):454 – 467, 2011.

[33] Andy Yen-Di Tsen, Shi Shang Jang, David Shan Hill Wong, and Babu Joseph. Predictive control of quality in batch polymerization using hybrid ann models. *AIChE Journal*, 42(2):455–465, 1996.

[34] Hong-Te Su, N. Bhat, P.A. Minderman, and T.J. McAvoy. Integrating neural networks with first principles models for dynamic modeling. *IFAC Proceedings Volumes*, 25(5):327 – 332, 1992. 3rd IFAC Symposium on Dynamics and Control of Chemical Reactors, Distillation Columns and Batch Processes (DYCORD+ '92), Maryland, USA, 26-29 April.

[35] Francis J. Doyle, Christopher A. Harrison, and Timothy J. Crowley. Hybrid model-based approach to batch-to-batch control of particle size distribution in emulsion polymerization. *Computers & Chemical Engineering*, 27(8):1153 – 1163, 2003. 2nd Pan American Workshop in Process Systems Engineering.

[36] Moritz von Stosch, Rui Oliveira, Joana Peres, and Sebastião Feyo de Azevedo. Hybrid semi-parametric modeling in process systems engineering: Past, present and future. *Computers & Chemical Engineering*, 60:86 – 101, 2014.

[37] N. Bhutani, G. P. Rangaiah, and A. K. Ray. First-principles, data-based, and hybrid modeling and optimization of an industrial hydrocracking unit. *Industrial & Engineering Chemistry Research*, 45(23):7807–7816, 2006.

[38] Mohammed Saad Faizan Bangi and Joseph Sang-II Kwon. Deep hybrid modeling of chemical process: Application to hydraulic fracturing. *Computers & Chemical Engineering*, 134:106696, 2020.

[39] Martin Wijaya Hermanto, Richard D. Braatz, and Min-Sen Chiu. Integrated batch-to-batch and nonlinear model predictive control for polymorphic trans-

formation in pharmaceutical crystallization. *AIChE Journal*, 57(4):1008–1019, 2011.

[40] Peter Van Overschee and B.L. de Moor. *Subspace Identification for Linear Systems*. Springer US, 1996.

[41] Michel Verhaegen and Patrick DeWilde. Subspace model identification part 1. the output-error state-space model identification class of algorithms. *International Journal of Control*, 56(5):1187–1210, 1992.

[42] J. S. Anderson, T. J. McAvoy, and O. J. Hao. Use of hybrid models in wastewater systems. *Industrial & Engineering Chemistry Research*, 39(6):1694–1704, 2000.

[43] Henricus J. L. Van Can, Chris Hellinga, Karel Ch. A. M. Luyben, Joseph J. Heijnen, and Hubert A. B. Te Braake. Strategy for dynamic process modeling based on neural networks in macroscopic balances. *AIChE Journal*, 42(12):3403–3418, 1996.

[44] Masoud Kheradmandi and Prashant Mhaskar. Model predictive control with closed-loop re-identification. *Computers & Chemical Engineering*, 109:249 – 260, 2018.

[45] Nikesh Patel, Prashant Mhaskar, and Brandon Corbett. Subspace based model identification for missing data. *AIChE Journal*, 66(10):e16538, 2020.

3.7 Supporting Information for Publication

Table 3.4: Seeded batch crystallizer parameters

Parameter nomenclature	
C: Solute concentration	T: Reaction temperature
C_m: Metastable concentration	C_s: Saturation concentration
μ_i: ith moment of the particle size distribution	T_{jk}: Jacket temperature
μ_i^n: ith moment corresponding to nucleation	ρ : Density of crystals
μ_i^s: ith moment corresponding to seed/desired	k_v: Volumetric shape factor
U: Overall heat transfer co-efficient	A: Total heat transfer surface area
M: Mass of solvent in crystallizer	C_p: Heat capacity of solution
ΔH: Heat of reaction	E_b: Nucleation activation energy
E_g: Growth activation energy	

Table 3.5: Seeded batch crystallizer parameter values and simulation settings

Parameters	Nominal Values
b	1.45
g	1.5
k_b	285 $s^{-1}\mu m^{-3}$
k_g	1.44e8 $\mu m/s$
ρ	2.66e-12 g/m^3
UA	0.8 KJ s K
k_v	1.5
C_p	3.8 KJ/(Kg K)
ΔH	44.5 KJ/Kg
E_b/R	7517 K
E_g/R	4859 K
M	27
dt	3.005 seconds
$T(0)$	50°C
$C(0)$	0.1742 g/g

Table 3.6: Erroneous parameters in the first principles model (original values of these parameters are listed in Table 4.4).

Parameter	% Mismatch
E_b/R	0.8
E_g/R	0.8
g	10
b	10

Table 3.7: Batch initial conditions and noise parameters

Variable	$\sigma_{batchwise}$	σ_{noise}
n	0.2	0.05
C	0.0028	0.008
T	0.52	0.003
$\mu_i^n, \mu_i^s (i = 1, 2, 3)$	-	0.001

Table 3.8: Model Identification database summary

Input policy	Set-point profile	Number of Batches
PI trajectory tracking($K_C = 0.85$, $T_I = 38$)	1	30
PI trajectory tracking ($K_C = 0.85$, $T_I = 38$)	2	10
Total batches		40

Table 3.9: Model and parameters

Model	Hankel rows	Model order
Hybrid-1,2: sfp model	34	30
Hybrid-1: residual model	61	57
Hybrid-2: residual model	64	60
Non-linear hybrid: Residual model	64	60
Subspace model	59	55

Table 3.10: Optimization Parameters

Parameters	Case -1	Case-2	Case-3
δ	$2°C/min$	$2°C/min$	$2°C/min$
γ	$9.930e9$	$1.093e10$	$1.133e10$

MPC tuning parameter values (Implemented as per the different layers of optimization)

$$\begin{aligned}
\epsilon_1 &= \begin{cases} 0.0035 \text{ g/g}, & \text{from } 10 - 22.5 \text{ min} \\ 0.0015 \text{ g/g} & \text{from } 22.5 - 30 \text{ min} \end{cases} \\
\epsilon_2 &= \begin{cases} -0.0035 \text{ g/g}, & \text{from } 10 - 22.5 \text{ min} \\ -0.0015 \text{ g/g}, & \text{from } 22.5 - 30 \text{ min} \end{cases} \\
\epsilon_3 &= 0.4e9, \text{ from } 10 - 30 \text{ min}
\end{aligned} \tag{3.34}$$

Table 3.11: Kalman tuning parameters used in the state estimation during MPC implementation for the 3 different MPC's

MPC - type	Model	Process noise (σ^2)	Measurement noise (σ^2)											
Hybrid-1	sfp	$1e-6$	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.003	0.1	0.1	0.1
	res	$1e-4$	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.003	0.1	0.1	0.1
Hybrid-2	sfp	$1e-6$	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.003	0.1	0.1	0.1
	res	$1e-4$	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.003	0.1	0.1	0.1
Subid	Subspace	$1e-4$	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.003	0.1	0.1	0.1

Chapter 4

Hybrid Partial Least Squares models for batch processes: Integrating data with process knowledge

The contents of this chapter have been published in *Industrial & Engineering Chemistry Research* Journal.

- Hybrid Partial Least Squares Models for Batch Processes: Integrating Data with Process Knowledge Debanjan Ghosh, Prashant Mhaskar, and John F. MacGregor *Industrial & Engineering Chemistry Research* 2021 60 (26), 9508-9520 DOI: 10.1021/acs.iecr.1c00865

Abstract

This paper presents a unique strategy for integrating fundamental process knowledge with measurement data to build a Partial Least Squares (PLS) model with improved estimation capability. To this end, variables from two different sources are combined to create the predictor data matrix for the PLS model. Measurement data from sensors is stored and used as inputs to a modified first principles model to generate trajectory data of unmeasured variables. Then the traditional X data matrix (built with measured data) is augmented with batch trajectory data of the calculated variables. The PLS model built with this augmented matrix is referred to as Hybrid/Augmented PLS and this proposed methodology is tested on a seeded batch crystallization process to illustrate this straightforward but powerful approach to estimate the final crystal size distribution. The efficacy of the proposed approach is demonstrated using simulation studies by comparing the results with the standard PLS and subspace based quality model.

4.1 Introduction

Batch and fed-batch processes [2] play a critical role in today's fast and constantly changing market demand of critical products. These small scale processes are characterized by their ability to manufacture on-spec product where quality is of vital importance, like in pharmaceutical, polymer, biochemicals etc. Open loop policies were implemented in the past where it was thought that reproducing a successful recipe for a particular batch would lead to similar results. However, such policies are rarely successful as each batch experiences a unique set of variations induced by raw material (impurity levels, variation in weights etc.), operational factors and unaccounted disturbances during the batch cycle, which lead to significant variations in the final quality. Automation for tight monitoring and control of such processes is of utmost importance in order to achieve desired targets and model based monitoring and feedback control strategies have been greatly successful in that regard, and thus significant effort is constantly invested to enhance the quality of such models.

Soft sensors or inferential models come to the aid where online and offline measurement of certain variables are difficult, infrequent and expensive to obtain. The estimation of such variables are typically done on the basis of their relation with other measured variables using process models. The availability of these additional variables has been harnessed to build more robust and reliable process monitoring and control schemes [29, 25, 19]. Traditionally, first principle or physics based models were used for such purposes. These models invoke fundamental governing laws of the process and are described in the form of non-linear/linear partial differential equations (PDE), ordinary differential equations (ODE), algebraic equations etc. and can also be in the form of detailed industrial simulators. However, developing such models can be a highly arduous task and needs great deal of process knowledge and expertise. Estimating model parameters to calibrate such models to process data typically involves a highly complex non-convex optimization which is quite difficult to solve. Although these high fidelity models can predict variable trajectories with relatively good accuracy, their online implementation is restricted due to their intricate nature leading to high computation times. Moreover, for online process monitoring applications [15], such models have to be implemented utilizing observers such as Extended Kalman Filters (EKF), Moving Horizon Estimators (MHE) etc., since all the model states (such as raw material variations and other disturbance states) are typically not available for measurement (and initialization of the model) [13, 18, 30].

Recent advancements in data acquisition and storage facilities have paved the way for building data-based models for monitoring, control, and optimization of complex process. These approaches are fairly easy to model and have a simple model structure, which make them well suited for online deployment. Some of the existing techniques include Auto-Regressive Exogenous (ARX), Artificial Neural Networks (ANN), latent variable methods (Principal Component Analysis (PCA), Partial Least Squares (PLS)) and subspace identification, where data from past batches is used to identify the parameters of the model.

ARX models are linear time invariant dynamic models and it is assumed that the process output at any particular sampling instant is a linear combination of past inputs and outputs. The model coefficients are determined by simple regression techniques using historical data. However, co-linearity between variables (a common happening in batch processes) can lead to improper model estimates affecting the predictive ability of the model. ANN's are static models which are adapted to model dynamic processes and use non-linear functions to relate process inputs and outputs. In the past, they have been used to model and monitor batch processes [35] but their efficacy relies on the availability of significant amount of historical data relevant to the application, and often struggle with the issues of overfitting.

Subspace identification methods [22, 26, 27] are linear algebra based approaches which identify a dynamic state-space model. In one such identification method [22], the subspace states are first identified and then the model matrices are determined using regression techniques. Traditionally built for continuous processes, this technique has been adapted to accommodate batches of different lengths [5] in the identification step. In the recent past, they have been applied to a variety of batch case studies and the results have demonstrated their efficacy in batch modeling and control.

PLS is a statistical latent variable modeling method which reduces the dimension of the original problem, and facilitates computations and visualizations in that lower dimensional space. This attribute makes them very well suited for processes associated with large number of correlated variables. In order to capture the time varying dynamics of batch processes, the inherently static models are appropriately transformed to effectively time varying linear models, called Multiway Partial Least Squares (MPLS) through batch wise unfolding of the historical batch data. Alignment methods have been developed to accommodate the non-uniformity that arises from the termination of batches at different time. This Multiway PCA/PLS approach has found tremendous usage in modeling of batch processes along with online applications like monitoring and control of such processes [25, 11, 12, 24, 17].

A good review on data based soft sensors has been provided in [16]. Auto-regressive moving average exogenous (ARMAX) models were used in [4] to determine the particle size measurement in a grinding circuit. In [38], the product composition of a distillation column was estimated using latent variable methods and ANN's. Emission monitoring of industrial gases was facilitated using soft sensors in [8, 29]. [20] used a dynamic partial least squares (DPLS) based soft sensor for the online estimation of variables related to product quality in a cement kiln system. In one example in the polymer sector [31], the use of PLS for such an application was presented where it was successfully used to estimate quality parameters for a LDPE plant. Further, in batch polymerization cases [10, 21], successful applications of PLS as a soft sensor for real time estimation were presented. For process monitoring applications, the usefulness of MPCA has been well documented [24, 17]. In [25], on-line monitoring of the qualities of final product using MPLS for a polymerization case was presented.

Grey-box or hybrid modeling schemes exist which make use of two modeling techniques to create better and efficient models. They are broadly classified into series [28, 34] and parallel approaches [33, 9]. [37, 1] present a good review of both the approaches. In the series approach, data based models like ANN [34], PLS etc. were placed in series with the first principles models and used to find the parameters of the mechanistic model. Parallel approaches use the data driven model to act as residual model to correct the error in prediction of the first principles model. In [33], ANN's were used for such purposes where the model was trained with historical error data (difference between the process and the first principles model). In [23], a series hybrid model (using PLS as the data driven unit) was embedded in an EKF to monitor the progress of key process variables in a mammalian cell culture case study. In another example [3], a synergistic usage of data and mechanistic model was presented for an online monitoring case study where an EKF was used to estimate states using predictions of a mechanistic model and measurement calculations provided by a PLS model. In a recent application [7], an extended kalman filter (EKF) was used to estimate variable trajectory data of unmeasured variables which were later appended with measured variables to create PLS models with enhanced estimation ability used in process monitoring for early detection of faults. In summary, while various hybrid modeling approaches exist, a hybrid modeling approach that combines the powerful PLS models with first principles knowledge for the purpose of estimating quality variables presently does not exist.

The present manuscript derives the motivation from the above consideration, and presents a hybrid PLS approach. The proposed approach involves supplementing the trajectories of recorded variables with additional data (coming from mechanistic models) in a MPLS modeling approach. The availability of a simplified mechanistic model is leveraged and used to generate trajectory data of variables using initial conditions and recorded measurement of process variables at each sampling instant. This simple yet powerful approach is referred to as Hybrid or Augmented PLS and it obviates the need of using and maintaining an EKF or any other state estimation technique pertaining to first principles models. An extremely important aspect of this hybrid modeling approach is that the fundamental model does not have to be calibrated to process data, and hence, can have significant biases between the true and the calculated variable trajectories. The only important point to be considered is that the predicted variations, although biased, should evolve in the direction of the true process. A motivating example of seeded batch crystallizer is used to illustrate the efficacy of the proposed approach. The quality variable - the crystal size distribution (CSD) at batch end is estimated using the proposed methodology and is compared with a standard PLS (trained only with measurement data) and subspace based quality model. The remainder of the manuscript is structured as follows: Section 5.2 describes the batch crystallizer and its modelling equations, followed by a brief introduction to MPLS and subspace based quality models. The proposed methodology is presented in Section 4.3 where the identification and validation of the model are described. Section 4.4 shows the application of the proposed method to the case considered, and the simulation results are presented where the final quality is estimated and compared with other methodologies. The conclusions are presented in Section 5.4.

4.2 Preliminaries

This section first presents two mechanistic models that describe the dynamics of batch crystallization process. The first model is only for the purpose of generating the data, i.e., it is used instead of real data coming from a process, while the second is the 'model' available to the practitioner. Next, a brief overview of two system identification techniques: MPLS and subspace identification based quality models is presented.

4.2.1 Motivating example: Seeded Batch Crystallization process

In this work, a seeded batch crystallization process is considered as the test bed for demonstrating the utility of the proposed approach. Batch crystallization is a process where typically an initial seed (crystals of a particular distribution) is thrown in the reactor containing the solution to initiate the nucleation process. It is expected that the crystals grow from the seeds over the course of crystallization to a desired crystal size distribution. The process of initial seeding restricts, to a degree, the formation of crystals due to unwanted nucleation, and is vital in obtaining a desired CSD at batch termination. This is also affected by the reactor temperature, which in turn affects the growth (of all crystals) and nucleation rate of new crystals. The reactor temperature is controlled by manipulating the jacket temperature. The PDE model used in [32] is utilized in this work where the population balance equation describes the dynamic time evolution of the CSD ($n(r,t)$), while the two associated ODEs dictate the time progression of the solution concentration (C) and reaction temperature (T) contingent on the initial conditions. These equations are presented in Equation (4.1) and are considered to be the 'process'. Gaussian measurement noise is added to the simulated data and the process is initiated with variations in the initial conditions to imitate process conditions. The parameter nomenclature is provided in Table 4.3. The PDE system of equations has the following form:

$$\begin{array}{c}\text{PDE model}\\ \text{(Test bed)}\end{array}\begin{cases} \frac{\partial n(r,t)}{\partial t} + G(t)\frac{\partial n(r,t)}{\partial r} = 0, \quad n(0,t) = \frac{B(t)}{G(t)} \\ \frac{dC}{dt} = -3\rho k_v G(t)\mu_2(t) \\ \frac{dT}{dt} = -\frac{UA}{MC_p}(T - T_j) - \frac{\Delta H}{C_p}3\rho k_v G(t)\mu_2(t) \end{cases} \tag{4.1}$$

$B(t)$ is the nucleation rate and $G(t)$ is the crystal growth rate, and is described as the form given in Equation (4.2):

$$\begin{aligned} B(t) &= k_b e^{-E_b/RT}(\frac{C - C_s(T)}{C_s(T)})^b \mu_3 \\ G(t) &= k_g e^{-E_g/RT}(\frac{C - C_s(T)}{C_s(T)})^g \end{aligned} \tag{4.2}$$

The moments are calculated from the CSD using Equation (4.3)

$$\mu_i = \int_0^\infty r^i n(r,t)dr \qquad\qquad i = 0,1,2,3\cdots \tag{4.3}$$

For the crystals to grow, the solution concentration (C) should be in between the saturation concentration (C_s) and metastable concentration (C_m), i.e., $C_s \leq C \leq C_m$. The saturation and metastable concentration are related to the reaction temperature as follows:

$$\begin{aligned} C_s(T) &= 6.29 * 10^{-2} + 2.46 * 10^{-3}T - 7.14 * 10^{-6}T^2 \\ C_m(T) &= 7.76 * 10^{-2} + 2.46 * 10^{-3}T - 8.10 * 10^{-6}T^2 \end{aligned} \tag{4.4}$$

The moments of CSD generated due to nucleation and seeding are, respectively, denoted by μ_i^n and μ_i^s (i= 0, 1, 2, 3). They are evaluated using Equation (4.5), where r_g is the defined as the threshold radius below which the crystals are considered to be as fines.

$$\begin{aligned} \mu_i^n &= \int_0^{r_g} r^i n(r,t)dr \\ \mu_i^s &= \int_{r_g}^{\infty} r^i n(r,t)dr \qquad\qquad i = 0,1,2,3\cdots \end{aligned} \tag{4.5}$$

The key features of a mechanistic model available to practitioners are that it is relatively simple to develop and maintain, might be structurally different from the 'true' process, and thus may contain process-model mismatch. We next describe such a simpler mechanistic model having the generic form :

$$\begin{aligned} \dot{\mathbf{x}}_{fp} &= f([\mathbf{x}_1, \mathbf{x}_2 \cdots \mathbf{x}_j \cdots \mathbf{x}_n], [\mathbf{u}_1, \mathbf{u}_2 \cdots \mathbf{u}_m]) \\ &= f(\mathbf{x}_{fp}, \mathbf{u}_{fp}) \\ \mathbf{Y}_{fp} &= g(\mathbf{x}_{fp}, \mathbf{u}_{fp}) \end{aligned} \tag{4.6}$$

where $\mathbf{x}_{fp} = \Big[\mathbf{x}_1, \mathbf{x}_2 \cdots \mathbf{x}_j \cdots \mathbf{x}_n\Big]^T$ is the vector of the n first principle model states, $\mathbf{u}_{fp} = \Big[\mathbf{u}_1, \mathbf{u}_2 \cdots \mathbf{u}_m\Big]^T$ is the vector of m inputs, $\mathbf{Y}_{fp}$ is the vector of outputs, f

and g are functions (linear/non-linear) that describe the progression of the states and outputs over the duration of the process.

For the crystallizer application, a moments model [32] with fewer degrees of freedom defined by a set of ODEs is utilized in the present manuscript. This reduced order model doesn't describe the time evolution of the CSD, but provides information about the important aspects of the distribution, and in particular, could be useful in improving purely data driven models. The time marching of the leading moments is described by the set of differential equations given by Equation (4.7):

$$\begin{array}{c}\text{ODE moments model}\\\text{(FP model)}\end{array}\begin{cases}\frac{d\mu_0^n}{dt} = B(t) & \\ \frac{d\mu_i^n}{dt} = iG(t)\mu_{i-1}^n(t) & i = 1,2,3\\ \mu_0^s = k_4 & \\ \frac{d\mu_i^s}{dt} = iG(t)\mu_{i-1}^s(t) & i = 1,2,3\\ \frac{dC}{dt} = -3\rho k_v G(t)(\mu_2^n(t) + \mu_2^s(t)) & \\ \frac{dT}{dt} = -\frac{UA}{MC_p}(T - T_j) - \frac{\Delta H}{C_p}3\rho k_v G(t)(\mu_2^n(t) + \mu_2^s(t)) & \end{cases} \tag{4.7}$$

The vector, x_{fp}, defining the model states is given by : $x_{fp} = \begin{bmatrix}\mu_0^n & \mu_i^n & \mu_0^s & \mu_i^s & C & T\end{bmatrix}^T$, $i = 1, 2, 3$ and the input by $u_{fp} = T_j$. The function f mentioned in Equation (4.6) describes the evolution of these states and are presented in Equation (4.7). The outputs in this study are considered the same as the first principles states, and thus $g(x_{fp}, u_{fp}) = x_{fp}$. In this manuscript, this simpler model is assumed to be available and considered as the first principle (FP) model.

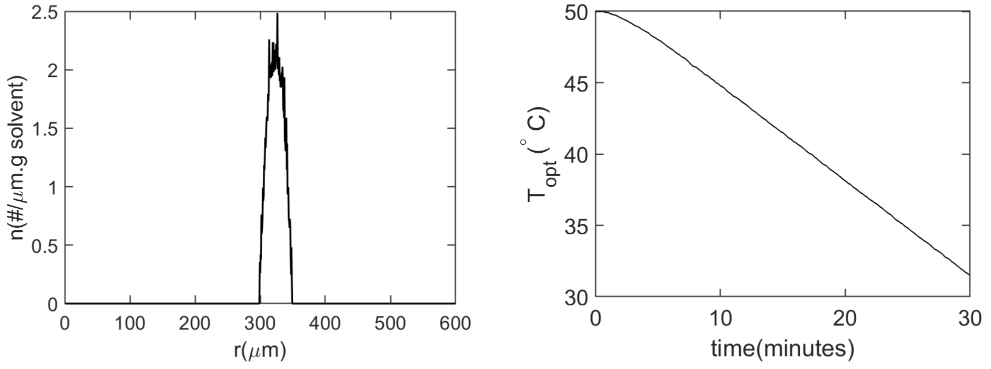
An extra condition given by Algorithm 2 is imposed on both the models which makes sure that no crystals are generated in the event the solution concentration falls below the saturation concentration during the batch run. This additional constraint is specifically important in the context of simulating data using the models as the fractional value of the exponents b and g (for the case $C \leq C_s$) in Equation (4.2) generate complex values of nucleation and growth rate. The value for growth rate (G) while implementing this particular condition for the PDE model is chosen to be a small number rather than 0 to avoid division by zeroes during the calculation of CSD as per Equation (4.1).

Algorithm 2 Condition on growth and nucleation rate dependence on solution concentration

```
if $C \leq C_s$ then
    $B = 0$
    $G = \begin{cases} 1e-6 & \text{PDE model} \\ 0 & \text{Moments model} \end{cases}$
else
    Use the values generated by Equation (4.2)
end if
```

2 variables are assumed to be available as process measurements represented by $y = \begin{bmatrix} C & T \end{bmatrix}^T$, and are generated in response to the manipulated trajectory where the input vector is given by $u = T_j$. Batches are initialized with variations in initial conditions and operated under closed loop using a Proportional Integral (PI) controller which tracks an optimum set-point trajectory (T_{opt}) of the reaction temperature (T). The manipulated input (jacket temperature T_j) is kept within the bounds $[50\ 30]°C$ using an input saturation constraint in the PI algorithm. The initial seed distribution (given by Equation (4.8)) for each batch is considered to be a parabola [32] spread out between radius 300 to 350 μm and having the maximum number of crystals (2/μm g solvent) at a radius of 325μm. A schematic of the initial CSD for a candidate batch and the set-point trajectory profile is shown in Figure 4.1.

$$n(r,0) = \begin{cases} 0 & r < 300\mu m \\ 0.0032(350-r)(r-300) & 300\mu m \leq r \leq 350\mu m \\ 0 & r > 350\mu m \end{cases} \tag{4.8}$$



(a)

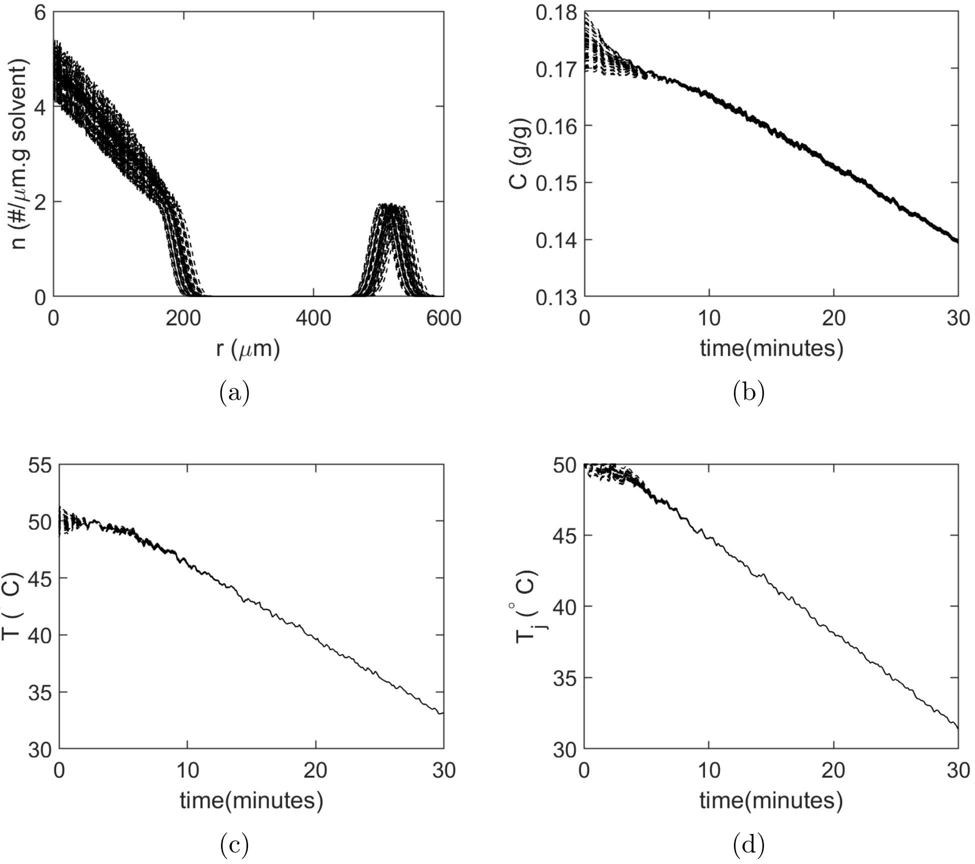
(b)

Figure 4.1: Figure (a) shows the nominal seed distribution at batch initialization and (b) shows the set-point trajectory profile of the reactor temperature

Data generated from $N_T = 50$ training batches are used to train the data-based models and validated on $N_V = 10$ validation batches. Variations in the initial seeds, solution concentration and temperature are introduced in each batch and measurement noise added to the simulated data at each sampling instant to reflect noisy sensor readings. Batches differ due to the presence of varying levels of impurities in raw materials and to reflect such occurrences in the simulation, the value of exponent g in Equation (4.2) is sampled from a normal distribution $\mathcal{N}(\mu(1.5), \sigma(0.001))$ to introduce differences among batches.

A moving average (MA) filter is used in order to mitigate the effect of measurement noise on the process outputs before using the data for feedback calculations. The window size of the filter is kept as 5. All the batches are run for a duration of 30 minutes, and the measurements are recorded at a sampling interval of dt. The parameter values and additional information needed for the simulation are provided in Tables 4.4, 4.6 and 4.7. Figure 4.2 shows the process variable trajectories of the outputs and inputs for all the historical batches to be used in training. It also presents the terminal crystal size distribution of those training batches.

The quality described by the CSD is measured only at batch end. The specific quality



(a)

(b)

(c)

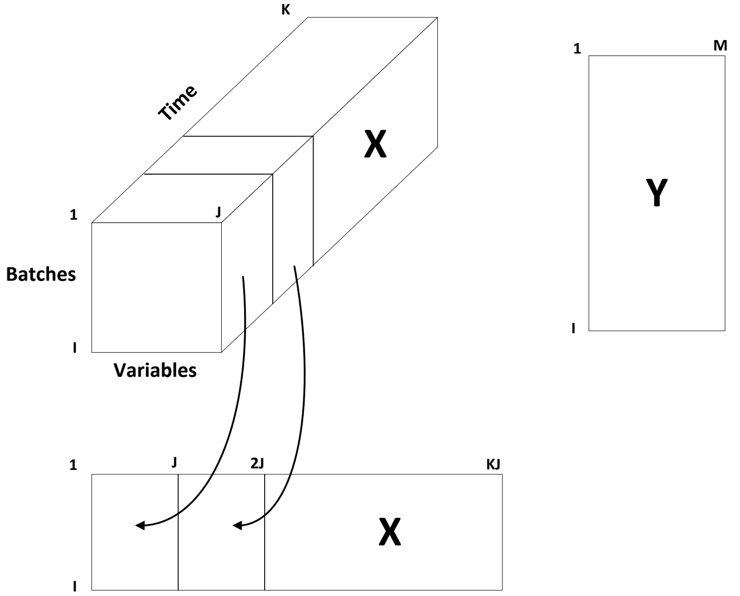
(d)

Figure 4.2: Figures present some of the variable profiles generated using the PDE model for all 50 training batches. Figure (a) presents the CSD at batch termination; (b), (c), and (d) show the solution concentration, temperature and jacket temperature trajectories respectively

problem that the present manuscript focuses on is the estimation of the terminal CSD by using continuous process measurements (such as temperature and concentration) for the entire batch.

4.2.2 Multiway Partial Least Squares

Evolution of the key process variables with time in each batch along with the final quality measurement is generally recorded and stored for future analysis. Batch data is typically organized in a 3-D array where the batches ($i = 1, 2, \cdots I$) run in rows, the measured variables ($j = 1, 2, \cdots J$) are positioned in columns and time ($k = 1, 2, \cdots K$) runs across the third dimension of the block. This input block denoted by $\mathbf{X}$ has the dimension of $I \times J \times K$ and defines the array of input/predictor variables. The quality variables ($m = 1, 2, \cdots M$) pertaining to each batch are stored in the rows of the Y block. These variables are typically measured at the end of each batch, and the block Y has a dimension of $I \times M$. MPLS is a technique to handle such block arrangement of batch data and aptly converts the 3-D input array to a 2-D array which can then be readily used to identify the parameters. There are multiple ways of unfolding the array, however, the most meaningful way of embedding the time varying nature of the process variables is to unfold it batch wise as shown in Figure 4.3. The 3-D array is first sliced up in K parts along the time dimension, each slice having a dimension of $I \times J$. These slices are then concatenated horizontally (to reflect the time evolution of the variables) to form a 2-D matrix having the dimension of $I \times JK$.



(a)

Figure 4.3: Batchwise Unfolding approach

The regular PLS algorithm is then applied on the input (unfolded $\mathbf{X}$) and output ($\mathbf{Y}$) data matrices following mean centering and scaling of the variables. Mean centering is done with respect to the average variable trajectories of the batches considered in the training data set and thus, the entries in the input block reflect the variation of the variables (at each sampling instant) about their respective mean trajectories. The variables are also scaled to unit variance to remove the effect of units which resolves improper assignation of weights to certain variables. This approach of unfolding data to build PLS models results in a model with time indexed parameters which appropriately captures the time varying nature of the process variables, and makes use of the variation in the process measurements till the end of the batch in the X space to predict the variation of the quality variables about their mean values in the Y space. Considering A components are used to fit the model, **X** and **Y** data matrices can be written down in terms of model estimates ($\hat{\mathbf{X}}$ and $\hat{\mathbf{Y}}$) and residual matrices (**E** and **F**) as :

$$\begin{aligned}\mathbf{X} &= \hat{\mathbf{X}} + \mathbf{E} = \mathbf{T}\mathbf{P}^{\mathbf{T}} + \mathbf{E} \\ \mathbf{Y} &= \hat{\mathbf{Y}} + \mathbf{F} = \mathbf{T}\mathbf{Q}^{\mathbf{T}} + \mathbf{F}\end{aligned} \tag{4.9}$$

where the matrix T ($I \times A$) is the matrix of scores, P ($JK \times A$) and Q ($M \times A$) are the loading matrices.

4.2.3 Subspace Identification

Subspace identification methods [27, 26, 22, 36], also commonly referred as SIMs, are system identification techniques which make rigorous use of concepts of linear algebra to identify a discrete linear time invariant (LTI) state-space model from historical input-output (I/O) measurement data, and is written down in the form given by Equation (4.10). In this study, the approach used in [22] is implemented where the states are first identified during the identification stage, and the state space system matrices are then evaluated using linear regression. Subspace algorithms are numerically robust and well suited for handling large data set. The identified LTI model is written down as follows:

$$\begin{aligned} \mathbf{x}[k+1] &= \mathbf{A}\mathbf{x}[k] + \mathbf{B}\mathbf{u}[k] \\ \mathbf{y}[k] &= \mathbf{C}\mathbf{x}[k] + \mathbf{D}\mathbf{u}[k] \end{aligned} \tag{4.10}$$

The subspace states at any time instant k are denoted by $\mathbf{x}[k] \in \mathbb{R}^{n\times 1}$ (where n is the identified order of the model). The outputs and inputs are represented by $\mathbf{y}[k] \in \mathbb{R}^{l\times 1}$ and $\mathbf{u}[k] \in \mathbb{R}^{m\times 1}$ (l and m being the number of measured outputs and inputs respectively). The time invariant system matrices identified by the algorithm are denoted by $\mathbf{A} \in \mathbb{R}^{\mathbf{n}\times\mathbf{n}}$, $\mathbf{B} \in \mathbb{R}^{\mathbf{n}\times\mathbf{m}}$, $\mathbf{C} \in \mathbb{R}^{\mathbf{l}\times\mathbf{n}}$, $\mathbf{D} \in \mathbb{R}^{\mathbf{l}\times\mathbf{m}}$.

The subspace based identification procedure being a dynamic model identification method, predicts the time evolution of the outputs and as such does not provide the estimation of the final batch end quality (a variable not involved in the identification). The notion of state variables dictates that the quality variables can only be a function of the process states and to estimate such variables, subspace based quality models have been built [6] where the quality at batch end is modeled as a linear function of the terminal subspace states. Any regression approach having the following form is used to determine the coefficients.

$$\mathbf{Q} = \mathbf{P} * \mathbf{x}[t_k] + \mathbf{G} \tag{4.11}$$

where $\mathbf{Q}$ is the chosen quality variable, $\mathbf{P}$ are the coefficients identified from regression, $\mathbf{x}[t_k]$ are the terminal subspace states and $\mathbf{G}$ is the residual not captured by the model.

Any state estimator can be employed to generate the states of the training batches using the identified system matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ and the known process measurements $(\mathbf{y}, \mathbf{u})$. For a validation batch '**b**', the state estimator can be run online and the states can be generated as process measurements become available. Once the terminal state $\mathbf{x}^{\mathbf{b}}[t_k]$ of that $\mathbf{b}^{\mathbf{th}}$ batch is determined, the final quality values can be obtained using

Equation (4.11).

Remark 17. *In a recent application [14], the use of subspace models to correct the biases of a mechanistic model has been addressed. The subspace model is used to model the residuals (error between the real measurements and the first principles model's prediction) and used in conjunction with a mechanistic model (in a parallel fashion) to create hybrid models with better predictive ability. This approach builds a dynamic model which can be initialized to predict in the future and finds application in model predictive control schemes. In this work, the case of estimating the final quality with all the measurements up to the end of batch is considered and as such, the applicability of the proposed model as a soft sensor is demonstrated. The utility of the approach as a predictive tool will be explored in future studies.*

4.3 Proposed hybrid modeling approach

A mechanistic model is considered to be available along with recorded measurements of variable trajectories of multiple historical batches. In this section, a modification applied to the first principles model is first presented followed by the proposed model structure describing the steps of identifying the linear model. The validation procedure on new batch data is discussed in the final subsection.

4.3.1 Modified first principles model

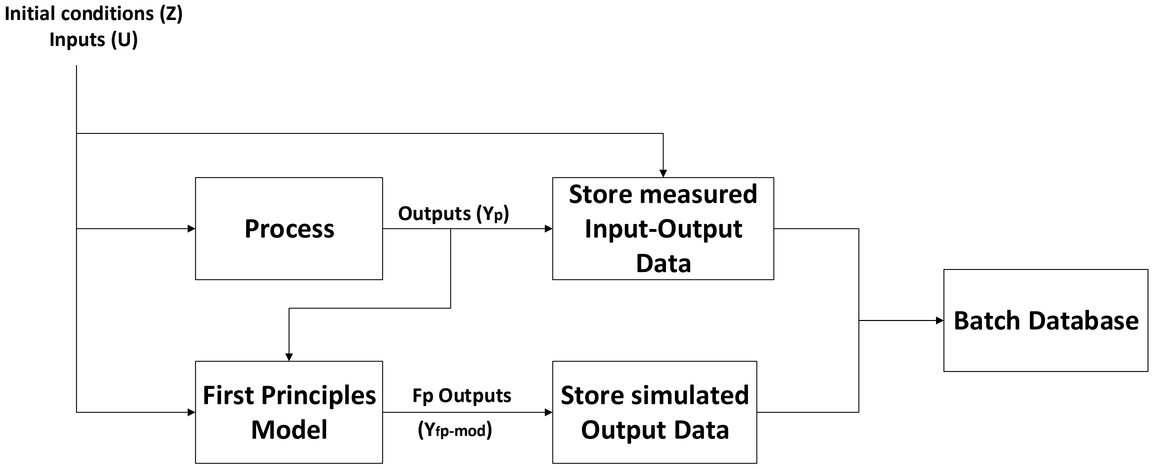
The available mechanistic model is modified so that the simulated trajectories are reflective of the true process occurrences and evolve with the right trends. The measured variables in batches are often the states of the first principles model and as such this knowledge from historical batches can be incorporated to generate variable trajectories of the unmeasured states. Considering the discrete time measurement of the states $\mathbf{x}_j, \cdots, \mathbf{x}_n$ is known to us from historical batch data, the trajectories of the unmeasured ones for those batches can be simulated using these states as additional inputs given by Equation (4.12). The number of states of the modified first principles is $\mathbf{x}_{fp-mod} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{j-1}]^T$ and the new input vector is given by $\mathbf{u}_{fp-mod} = [\mathbf{u}_1, \mathbf{u}_2 \cdots \mathbf{u}_m, \mathbf{x}_j, \cdots, \mathbf{x}_n]^T$. p, q represent a subset of the vector functions f, g in Equation (4.6), and some states now act as known inputs in the these functions. The general equation of the modified first principles is given by Equation (4.12).

$$\begin{aligned}
\dot{\mathbf{x}}_{fp-mod} &= p([\mathbf{x}_1, \mathbf{x}_2 \cdots \mathbf{x}_{j-1}], [\mathbf{u}_1, \mathbf{u}_2 \cdots \mathbf{u}_m, \mathbf{x}_j, \cdots, \mathbf{x}_n]) \\
&= p(\mathbf{x}_{fp-mod}, \mathbf{u}_{fp-mod}) \\
\mathbf{Y}_{fp-mod} &= q(\mathbf{x}_{fp-mod}, \mathbf{u}_{fp-mod})
\end{aligned} \tag{4.12}$$

4.3.2 Model identification

Traditionally, MPLS models are built with measurements coming from real process plants and are often limited by the number of sensors being used. Availability of a first principles model allows us to generate data of other important variables which provides additional information that can be leveraged for a variety of purposes. This proposed approach utilises that extra information in one particular fashion to build better and more efficient MPLS models.

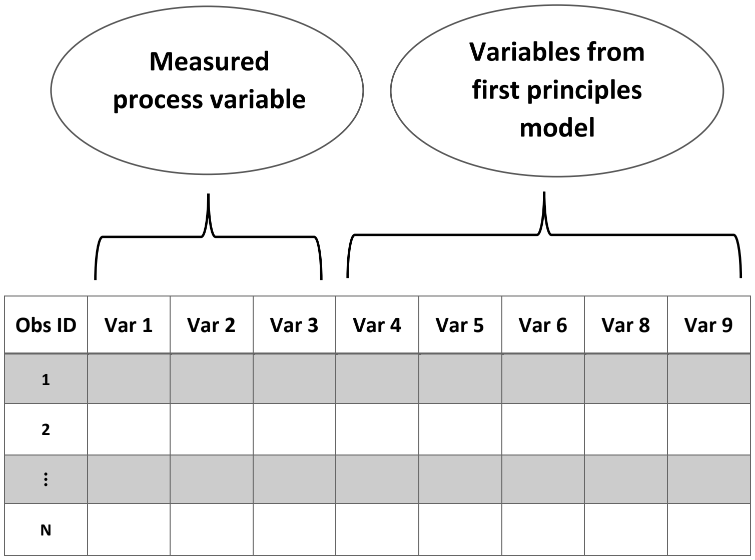
The schematic of the proposed hybrid approach is presented in Figure 4.4. The initial conditions ($\mathbf{Z}$) of historical batches along the time series data of different measured process variables ($\mathbf{Y_p}, \mathbf{U}$) are assumed to be known to us. $\mathbf{U}$'s denote the process (manipulated) inputs which are used to steer the batch towards a desired quality, and $\mathbf{Y_p}$'s are the output variable trajectories measured at certain sampling intervals over the course of the batch. The initial condition of the unmeasured states (over batch duration) of the first principles are also assumed to be known. In order to generate output data ($\mathbf{Y_{fp-mod}}$) of the modified first principles model, it is initialized with the vector ($\mathbf{Z}$) which comprises of the initial condition of the process variables. The inputs to the model are considered to be both $\mathbf{Y_p}$ and $\mathbf{U}$ such that at each sampling point we have information of the process states of modified first principles model. For one such batch, this additional information of inputs help guide the simulated outputs to evolve in the direction of that batch. This simulated data generated by the first principles model is then stored and appended with the recorded measurements to create the augmented batch database. A MPLS model is then built with this augmented $\mathbf{X}$ block, $\mathbf{X_{aug}} = \begin{bmatrix}\mathbf{Y_p} & \mathbf{U} & \mathbf{Y_{fp-mod}}\end{bmatrix}$ and the $\mathbf{Y}$ block containing the batch end quality data $\mathbf{Y} = \begin{bmatrix}\mathbf{Q}\end{bmatrix}$.



(a)

Figure 4.4: Schematic showing the methodology followed to generate simulated outputs to augment the database of historical batches

The model identification for this augmented case is done the same as a standard MPLS which uses only measurement data. A schematic of the unfolded $\mathbf{X}$ block of the Augmented/Hybrid PLS method is presented in Figure 4.5. A cross-validation methodology is adopted to fit the best number of components.


Measured process variable
Variables from first principles model

Obs ID	Var 1	Var 2	Var 3	Var 4	Var 5	Var 6	Var 8	Var 9
1								
2								
⋮								
N								

(a)

Figure 4.5: X block of Augmented/Hybrid PLS

4.3.3 Model validation

In this work, the problem of estimating the batch end quality is considered where all the measurements and generated outputs up to batch termination are considered to be available. When implementing the method online for a new batch, the first principles model is run in real-time having the same conditions as the process, and the measured variables are fed as inputs to the model to generate simulated outputs at each time instant. Therein lies the utility of using simple reduced order mechanistic models which reduce the computational burden, and the model calculations can be performed easily within the sampling interval. Once the entire observation row is obtained, the quality variable is estimated using the coefficients identified during the model building stage.

Remark 18. *This methodology can very well be used to generate information of variables in real time for process monitoring purposes. In such an application, at any time instant, a new row of observation contains information of the measured and simulated* *variables up to that instant in the batch, and the future values of the variables can be predicted using missing data algorithms. These algorithms make use of the correlation structure of the historical batches to predict the missing values. Once these values are predicted the final quality can also be monitored in real time.*

4.4 Application to the batch crystallizer case study

This section presents the application of the proposed approach to the seeded batch crystallization process. The development of the augmented batch database for the Hybrid PLS is first explained in Section 4.4.1 followed by the simulation results on validation batches in Section 4.4.2 where it is compared with a standard PLS and subspace based quality model.

4.4.1 Augmented PLS modeling of the seeded batch crystallizer

The ODE moments model described in Section 4.2.1 is considered as the first principles for this study. The hybrid modeling strategy proposed in this work is unique in its way of utilizing the first principles model to aid a data model as compared to the traditional grey box models. The objective here is to build better MPLS models by providing it with additional information of process variable trajectories during the model building stage, which otherwise is not known. In order to achieve that, the ODE model is fed with real process measurements as inputs, $u_{fp-mod} = \begin{bmatrix} C & T & T_j \end{bmatrix}$ at every time instant k to generate the value of model states $x_{fp-mod} = \begin{bmatrix} \mu_0^n & \mu_0^s & \mu_i^n & \mu_i^s \end{bmatrix}^T$, $i = 1, 2, 3$ at the next sampling instant. Values of some additional variables C_s, C_m are also generated during this process and are utilized later during the data model building stage. In this case study, the function p, stated in Equation (4.12), is a vector comprising of functions that describe the evolution of these states and is given by the algebraic equations and ODEs in Equations (4.13) to (4.15). The outputs are described by the function q, given by $q(x_{fp-mod}, u_{fp-mod}) = \begin{bmatrix} x_{fp-mod} & C_s & C_m \end{bmatrix}$. It is important to note that in this modified first principles, the solution concentration (C) and reaction temperature (T) are not model states anymore, and are introduced as known inputs available from historical batch trajectory data.

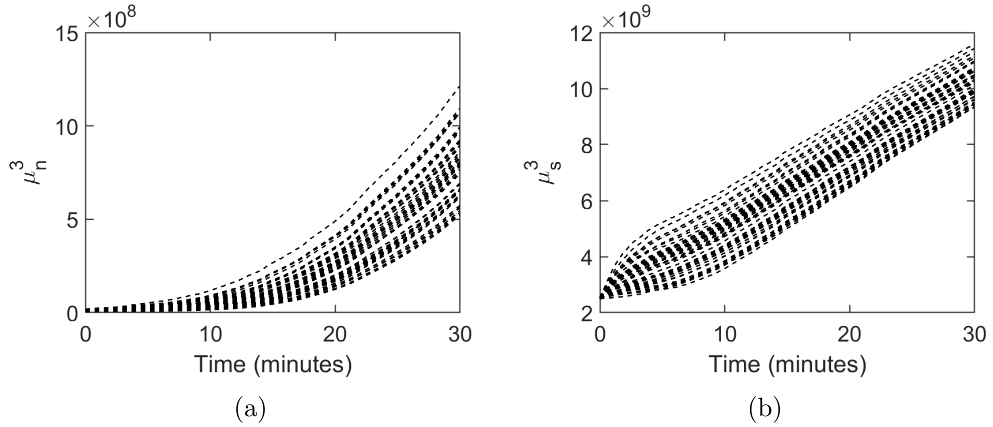
$$\begin{aligned} C_s(T) &= 6.29 * 10^{-2} + 2.46 * 10^{-3}T - 7.14 * 10^{-6}T^2 \\ C_m(T) &= 7.76 * 10^{-2} + 2.46 * 10^{-3}T - 8.10 * 10^{-6}T^2 \end{aligned} \tag{4.13}$$

$$\begin{aligned} B(t) &= k_b e^{-E_b/RT}(\frac{C - C_s(T)}{C_s(T)})^b \mu_3 \\ G(t) &= k_g e^{-E_g/RT}(\frac{C - C_s(T)}{C_s(T)})^g \end{aligned} \tag{4.14}$$

$$\begin{aligned} \frac{d\mu_0^n}{dt} &= B(t) \\ \frac{d\mu_i^n}{dt} &= iG(t)\mu_{i-1}^n(t) \qquad\qquad i = 1, 2, 3 \\ \mu_0^s &= k_4 \\ \frac{d\mu_i^s}{dt} &= iG(t)\mu_{i-1}^s(t) \qquad\qquad i = 1, 2, 3 \end{aligned} \tag{4.15}$$

The database previously containing the measurement data of $N_T = 50$ training batches is supplemented with data simulated from the first principles model. The batch trajectories of two such variables are presented in Figure 4.6. A Hybrid/Augmented MPLS model is then built with the information provided below :

1. **Initial conditions (Z) :**
   - T_0, initial reaction temperature
   - C_0, initial solute concentration
   - $\mu^n_{0,0}$, $\mu^s_{0,0}$, $\mu^n_{i,0}$, $\mu^s_{i,0}$, $i = 1, 2, 3$, the initial value of moments



(a) (b)

Figure 4.6: Figures present the trajectories of some variables generated by the modified first principles model for the 50 training batches. Figure (a) presents the 3rd moment due to nucleation; and (b) 3rd moment due to seeding respectively

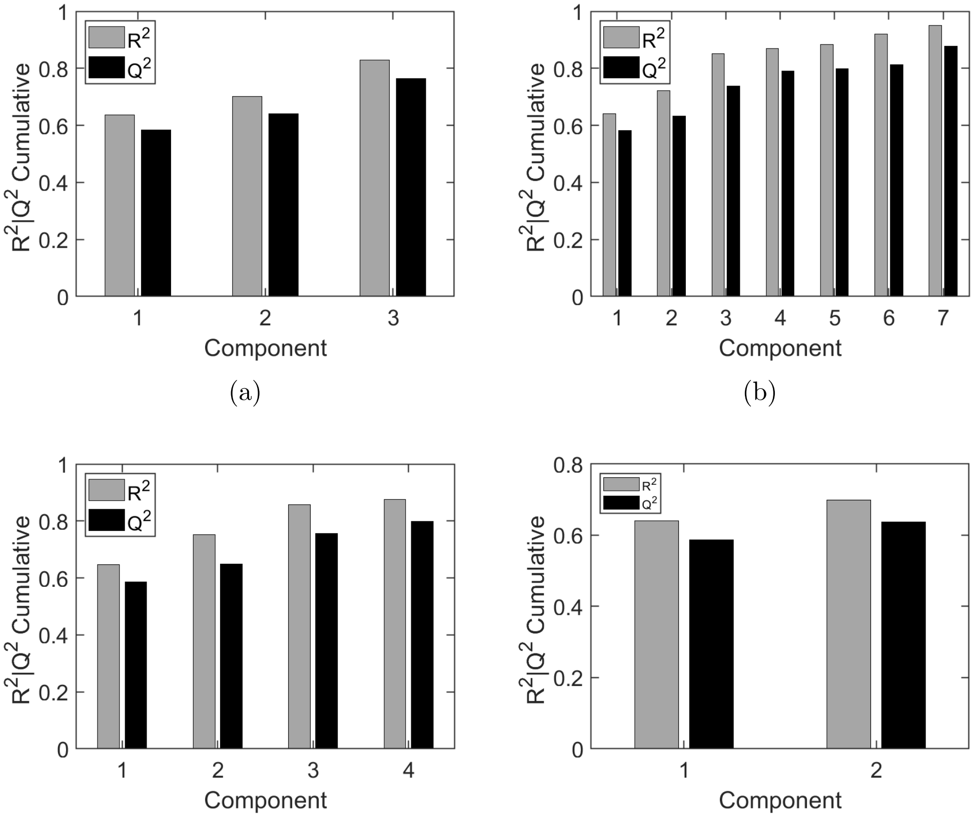
2. **Trajectory data measured during the process :**
   - $\begin{bmatrix}\mathbf{Y_p} & \mathbf{U}\end{bmatrix} = \begin{bmatrix}T & C & T_j\end{bmatrix}$
3. **Trajectory data from the first principles model given $\mathbf{Z}, \mathbf{Y_p}, \mathbf{U}$:**
   - $\begin{bmatrix}\mathbf{Y_{fp-mod}}\end{bmatrix} = \begin{bmatrix}\mu_0^n & \mu_i^n & \mu_0^s & \mu_i^s & Cs & Cm\end{bmatrix}$, $i = 1, 2, 3$
4. **Augmented X block :**
   - $\mathbf{X_{aug}} = \begin{bmatrix}\mathbf{Y_p} & \mathbf{U} & \mathbf{Y_{fp-mod}}\end{bmatrix}$
5. **Crystallization quality data :**
   - Crystal size distribution at batch end, $\mathbf{Y} = \begin{bmatrix}n\end{bmatrix}$.

Remark 19. *The case study in this manuscript considers and represents cases where the states of the first principles model, while not available through measurement for the duration of the batch, are readily available initially, or known by virtue of initialization. Thus, in the present case, the seed distribution is known through measurement, and it is also known that the moments of the newly nucleated crystals are zero, and this can be utilized to initialize the first principles model. Note that online measurement of these* *moments over the batch duration is considered unavailable. The situation where there are errors in the measured initial conditions is already incorporated into the hybrid PLS model since the model is based on historical data that has these measurement errors in the data. The only effect of increased errors in the initial conditions will be to make both the hybrid model somewhat less predictive. If many of the initial conditions are unknown then this approach may not be appropriate and instead some form of state and initial condition estimator approach may be required. However, those state estimation approaches are much more complex than the simple approach presented here. Certain practical issues such as missing data will also be addressed in the application of the proposed framework.*

4.4.2 Simulation results and analysis

Two simulation case studies are performed to illustrate the key observations.

- **Case - 1:** Measurement data from 50 historical batches is used to build both the standard and the Augmented PLS (referred as AUG-PLS-modfp). The trajectory data of additional variables Y_{fp-mod}, as mentioned in Section 4.4.1, generated from the modified first principles model are also introduced into the X-block of the Hybrid PLS. This X block of the Augmented/Hybrid PLS model is presented in Figure 4.11. 3 and 7 components are used, respectively, to fit the standard PLS and the Augmented PLS model using the auto-fit option in Aspen ProMV. The auto-fit option uses the cross-validation metric Q^2 for this purpose. In order to evaluate the efficacy of the models, they are validated on 10 new batches to predict the final CSD. Another hybrid PLS model is built using measurement data along with additional trajectory data generated from the standard first principles (initialized with the initial conditions from the process, the unmeasured states, and without the addition of measured variable information as inputs). It is referred to as to as AUG-PLS-stdfp. 4 components are used to build the model using the auto-fit option and then validated on the same 10 fresh batches.



(a)

(b)

(c)

(d)

Figure 4.7: Figures (a), (b), (c) and (d) respectively, present the R^2 and Q^2 metrics for the Y-space using the Pure-PLS, Aug-PLS-modfp, Aug-PLS-stdfp and Sub-PLS-Kalman strategies

The predictions are further compared with a subspace based quality model explained in Section 5.2.4. A model order of 3 is chosen for the subspace model. The terminal subspace states of the 50 training batches are generated by employing a kalman filter, following which the PLS models were built relating those states with the final CSD. The PLS model is referred to as Sub-PLS-Kalman. The kalman tuning parameters are provided in Table 4.8. 2 components are used to fit the model and validated on the 10 fresh batches. The R^2 and Q^2 value for all the methodologies used are shown in Figure 4.7. The root mean square (RMSE) values using the different modeling techniques are presented in Table 4.1. The predicted values of Aug-PLS-modfp and Aug-PLS-stdfp are compared with the true process values, and the CSD for two such candidate batches is shown in Figure 4.8. The Sub-PLS-Kalman estimation of CSD is not shown due to higher RMSE with respect to Pure-PLS, and to avoid clutter in the plots. It can be seen from the results that the AUG-PLS-modfp outperforms the general PLS approach and the other methodologies considered in this study.

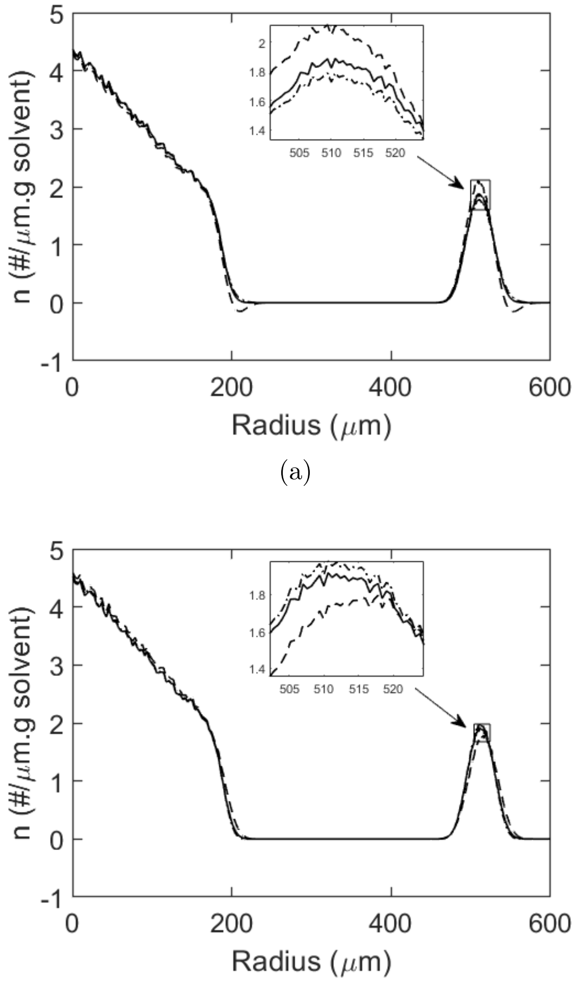
Model	Cumulative RMSE (10 batches)
AUG-PLS-modfp	0.21081
AUG-PLS-stdfp	0.28006
Pure-PLS	0.39247
Sub-PLS-Kalman	0.64174

Table 4.1: The RMSE values of 10 validation batches using different modeling techniques for Case-1

However, while the the augmenting of the X-block with additional trajectory data is valuable, the augmentation need not always be ($\mathbf{Y_{fp-mod}}$) generated using the modified first principles model and another simulation study is done is to illustrate this point.

- **Case - 2:** In this study, the same measurement data is considered as the last time. However, data generated from the first principles model data is different as another set of mismatch is considered in the parameter values of the first principles model. The mismatch in parameters are reported in Table 4.5. 7 and 8 components are used to fit AUG-PLS-modfp and AUG-PLS-stdfp, respectively, using the auto-fit option in ASPEN ProMV. The two models are compared for this case and the RMSE values are reported in Table 4.2.

  This analysis shows that there can be cases where the PLS model built using measurement data along with additional data from the standard first principles can



(a)

(b)

Figure 4.8: Figures (a),(b), respectively, present the CSD at batch termination of two candidate batches. The Process/measured CSD is given by the $-$ line, prediction of Pure-PLS by $--$ and the prediction of Augmented PLS by $-.$ line

Model	Cumulative RMSE (10 batches)
Aug-PLS-modfp	0.20979
Aug-PLS-stdfp	0.11343

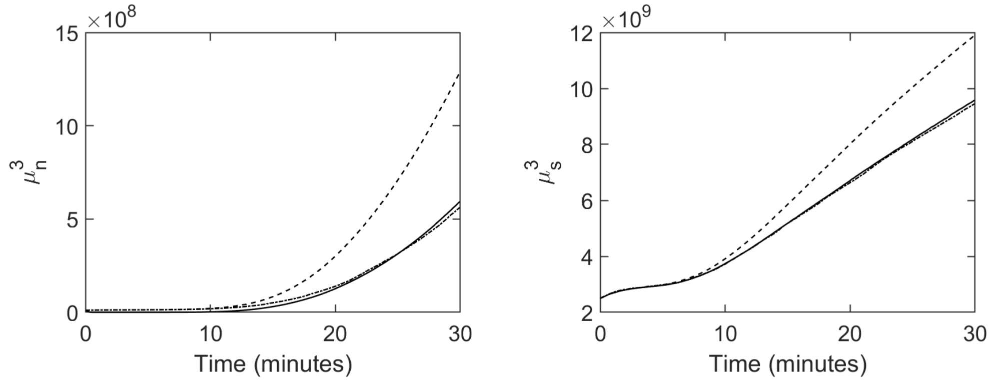
Table 4.2: The RMSE values of 10 validation batches using different modeling techniques for Case-2

outperform the addition from modified first principles model.

In order to understand the disparity in results of the two hybrid PLS models in the two cases, the PDE process data is plotted with the data from the standard first principles and the modified first principles. It should be noted that this analysis is possible only because the PDE model (considered as process) provides us the additional information of first principle states along with those considered as measured outputs. This additional information of true states for all practical cases won't be available as the historical data is based on sensor readings and not simulations.

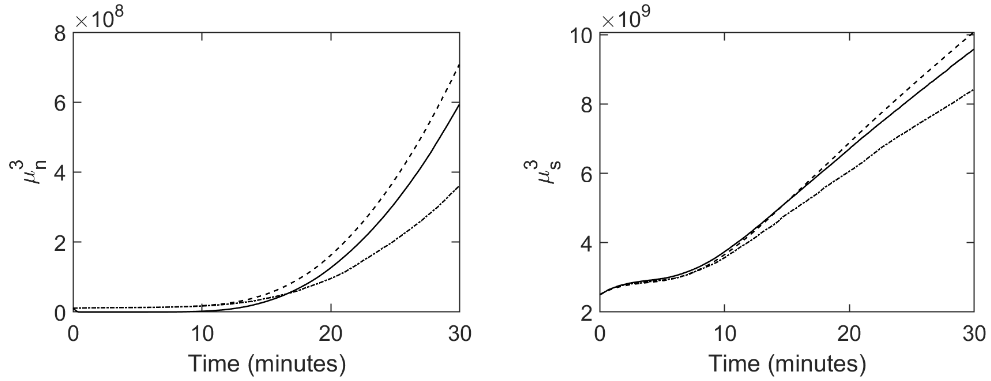
It can be observed from Figure 4.9 that in Case-1, the addition of measured variables as input to the ODE model (termed as modified first principles) shows a great improvement in its predictions over the standard ODE model, as this modified strategy drives the state trajectories closer to the true process states. This improvement is also reflected in the RMSE value (Table 4.1) of the PLS model built with the state trajectories of the modified first principles model. However, in Case-2, with a different set of parameter mismatch values, the trend appears to be different and is shown in Figure 4.10. The standard first principle predictions are closer to the process than the modified one and the RMSE values (Table 4.2) of the PLS models built with such data attests to that fact.

In practice, although, the knowledge of the true process states may not be available, an offline analysis involving building the two hybrid PLS models and validating them on historical batch data can easily be done. This will reveal the best choice of the hybrid model to be used for predicting the CSD of fresh batches.



(a) (b)

Figure 4.9: Figures compares the third moment due to nucleation (a) and due to seeding (b) predicted by the PDE model (−), Modified first principles model (−.) and the standard first principles model (−−) for Case-1



(a) (b)

Figure 4.10: Figures compares the third moment due to nucleation (a) and due to seeding (b) predicted by the PDE model (−), Modified first principles model (−.) and the standard first principles model (−−) for Case-2

4.5 Conclusions

A novel strategy of incorporating measurement data into first principles models to generate state trajectory information has been proposed. This additional information is incorporated to improve the predictive ability of MPLS models and the proposed scheme is validated using simulation case studies for the estimation of final quality of a batch crystallization process. This simple yet effective Augmented PLS strategy has been shown to perform better than the standard PLS and subspace based quality models. Two approaches to generate the augmented data are shown and case studies that illustrate the suitability of one particular approach versus the other are presented.

Supporting Information

Seeded batch crystallizer parameters; Parameter values of the batch crystallizer and simulation settings; Percentage error in the parameters of the first principles model; Batch initial conditions and noise parameters; PI controller settings used in the identification and validation batches; Tuning parameters of Kalman filter used in the estimation of terminal subspace states; X block of Augmented/Hybrid PLS for the crystallizer case study. This information is available free of charge via the Internet at http://pubs.acs.org/.

4.6 Acknowledgments

Financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC) through the Engage Grant, and the McMaster Advanced Control Consortium (MACC) is gratefully acknowledged.

Bibliography

[1] Bhutani, N., Rangaiah, G. P., and Ray, A. K. (2006). First-principles, data-based, and hybrid modeling and optimization of an industrial hydrocracking unit. *Industrial & Engineering Chemistry Research*, 45(23):7807–7816.

[2] Bonvin, D., Srinivasan, B., and Hunkeler, D. (2006). Control and optimization of batch processes. *IEEE Control Systems Magazine*, 26(6):34–45.

[3] Cabaneros Lopez, P., Udugama, I. A., Thomsen, S. T., Roslander, C., Junicke, H., Iglesias, M. M., and Gernaey, K. V. (2021). Transforming data to information: A parallel hybrid model for real-time state estimation in lignocellulosic ethanol fermentation. *Biotechnology and Bioengineering*, 118(2):579–591.

[4] Casali, A., Gonzalez, G., Torres, F., Vallebuona, G., Castelli, L., and Gimenez, P. (1998). Particle size distribution soft-sensor for a grinding circuit. *Powder Technology*, 99(1):15 – 21.

[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] Destro, F., Facco, P., García Muñoz, S., Bezzo, F., and Barolo, M. (2020). A hybrid framework for process monitoring: Enhancing data-driven methodologies with state and parameter estimation. *Journal of Process Control*, 92:333 – 351.

[8] Dong Dong, McAvoy, T. J., and Chang, L. J. (1995). Emission monitoring using multivariate soft sensors. In *Proceedings of 1995 American Control Conference - ACC'95*, volume 1, pages 761–765 vol.1.

[9] Doyle, F. J., Harrison, C. A., and Crowley, T. J. (2003). Hybrid model-based approach to batch-to-batch control of particle size distribution in emulsion polymerization. *Computers & Chemical Engineering*, 27(8):1153 – 1163. 2nd Pan American Workshop in Process Systems Engineering.

[10] Facco, P., Doplicher, F., Bezzo, F., and Barolo, M. (2009). Moving average pls soft sensor for online product quality estimation in an industrial batch polymerization process. *Journal of Process Control*, 19(3):520 – 529.

[11] Flores-Cerrillo, J. and MacGregor, J. F. (2002). Control of particle size distributions in emulsion semibatch polymerization using mid-course correction policies. *Industrial & Engineering Chemistry Research*, 41(7):1805–1814.

[12] 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.

[13] Gagnon, L. and Macgregor, J. F. (1991). State estimation for continuous emulsion polymerization. *The Canadian Journal of Chemical Engineering*, 69(3):648–656.

[14] 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.

[15] Isermann, R. (1984). Process fault detection based on modeling and estimation methods—a survey. *Automatica*, 20(4):387 – 404.

[16] Kadlec, P., Gabrys, B., and Strandt, S. (2009). Data-driven soft sensors in the process industry. *Computers Chemical Engineering*, 33(4):795 – 814.

[17] Kourti, T., Nomikos, P., and MacGregor, J. F. (1995). Analysis, monitoring and fault diagnosis of batch processes using multiblock and multiway pls. *Journal of Process Control*, 5(4):277–284. IFAC Symposium: Advanced Control of Chemical Processes.

[18] Kozub, D. J. and MacGregor, J. F. (1992). State estimation for semi-batch polymerization reactors. *Chemical Engineering Science*, 47(5):1047 – 1062.

[19] Kresta, J., Marlin, T., and MacGregor, J. (1994). Development of inferential process models using pls. *Computers Chemical Engineering*, 18(7):597 – 611. An International Journal of Computer Applications in Chemical Engineering.

[20] Lin, B., Recke, B., Knudsen, J. K., and Jørgensen, S. B. (2007). A systematic approach for soft sensor development. *Computers Chemical Engineering*, 31(5):419 – 425. ESCAPE-15.

[21] Marjanovic, O., Lennox, B., Sandoz, D., Smith, K., and Crofts, M. (2006). Real-time monitoring of an industrial batch process. *Computers Chemical Engineering*, 30(10):1476 – 1481. Papers form Chemical Process Control VII.

[22] Moonen M, Demoor B, V. L. V. J. (1989). Online and off-line identication of linear state-space models international journal of control. *International Journal of Control*, pages 49:219–232.

[23] Narayanan, H., Behle, L., Luna, M. F., Sokolov, M., Guillén-Gosálbez, G., Morbidelli, M., and Butté, A. (2020). Hybrid-ekf: Hybrid model coupled with extended kalman filter for real-time monitoring and control of mammalian cell culture. *Biotechnology and Bioengineering*, 117(9):2703–2714.

[24] Nomikos, P. and MacGregor, J. F. (1994). Monitoring batch processes using multiway principal component analysis. *AIChE Journal*, 40(8):1361–1375.

[25] Nomikos, P. and MacGregor, J. F. (1995). Multi-way partial least squares in monitoring batch processes. *Chemometrics and Intelligent Laboratory Systems*, 30(1):97 – 108. InCINC '94 Selected papers from the First International Chemometrics Internet Conference.

[26] Overschee, P. V. and Moor, B. D. (1992). Two subspace algorithms for the identification of combined deterministic-stochastic systems. In *[1992] Proceedings of the 31st IEEE Conference on Decision and Control*, pages 511–516 vol.1.

[27] Peter van Overschee, B. d. M. (1996). *Subspace Identification for Linear Systems*. Springer US.

[28] Psichogios, D. C. and Ungar, L. H. (1992). A hybrid neural network-first principles approach to process modeling. *AIChE Journal*, 38(10):1499–1511.

[29] Qin, S. J., Yue, H., and Dunia, R. (1997). Self-validating inferential sensors with application to air emission monitoring. *Industrial & Engineering Chemistry Research*, 36(5):1675–1685.

[30] Schuler, H. and Schmidt, C.-U. (1992). Calorimetric-state estimators for chemical reactor diagnosis and control: review of methods and applications. *Chemical Engineering Science*, 47(4):899 – 913.

[31] Sharmin, R., Sundararaj, U., Shah, S., Vande Griend, L., and Sun, Y.-J. (2006). Inferential sensors for estimation of polymer quality parameters: Industrial application of a pls-based soft sensor for a ldpe plant. *Chemical Engineering Science*, 61(19):6372 – 6384.

[32] 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. Advances in population balance modelling.

[33] Su, H.-T., Bhat, N., Minderman, P., and McAvoy, T. (1992). Integrating neural networks with first principles models for dynamic modeling. *IFAC Proceedings Volumes*, 25(5):327 – 332. 3rd IFAC Symposium on Dynamics and Control of Chemical Reactors, Distillation Columns and Batch Processes (DYCORD+ '92), Maryland, USA, 26-29 April.

[34] Tsen, A. Y.-D., Jang, S. S., Wong, D. S. H., and Joseph, B. (1996). Predictive control of quality in batch polymerization using hybrid ann models. *AIChE Journal*, 42(2):455–465.

[35] Venkatasubramanian, V. and Chan, K. (1989). A neural network methodology for process fault diagnosis. *AIChE Journal*, 35(12):1993–2002.

[36] Verhaegen, M. and DeWilde, P. (1992). Subspace model identification part 1. the output-error state-space model identification class of algorithms. *International Journal of Control*, 56(5):1187–1210.

[37] von Stosch, M., Oliveira, R., Peres, J., and de Azevedo, S. F. (2014). Hybrid semi-parametric modeling in process systems engineering: Past, present and future. *Computers & Chemical Engineering*, 60:86 – 101.

[38] Zamprogna, E., Barolo, M., and Seborg, D. E. (2005). Optimal selection of soft sensor inputs for batch distillation columns using principal component analysis. *Journal of Process Control*, 15(1):39 – 52.

4.7 Supporting Information for Publication

Table 4.3: Seeded batch crystallizer parameters

Parameter nomenclature	
C: Solute concentration	T: Reaction temperature
C_m: Metastable concentration	C_s: Saturation concentration
μ_i: ith moment of the particle size distribution	T_{jk}: Jacket temperature
μ_i^n: ith moment corresponding to nucleation	ρ : Density of crystals
μ_i^s: ith moment corresponding to seed/desired	k_v: Volumetric shape factor
U: Overall heat transfer co-efficient	A: Total heat transfer surface area
M: Mass of solvent in crystallizer	C_p: Heat capacity of solution
ΔH: Heat of reaction	E_b: Nucleation activation energy
E_g: Growth activation energy	
b: exponent relating nucleation rate to supersaturation	
g: exponent relating growth rate to supersaturation	

Table 4.4: Parameter values of the batch crystallizer and simulation settings

Parameters	Nominal Values
b	1.45
g	1.5
k_b	285 $s^{-1}\mu m^{-3}$
k_g	1.44e8 $\mu m/s$
ρ	2.66e-12 g/m^3
UA	0.8 KJ s K
k_v	1.5
C_p	3.8 KJ/(Kg K)
ΔH	44.5 KJ/Kg
E_b/R	7517 K
E_g/R	4859 K
M	27
dt	6.01 seconds
$T(0)$	50°C
$C(0)$	0.1742 g/g

Table 4.5: Percentage error in the parameters of the first principles model (original values of these parameters are listed in Table 4.4).

Parameter	% Mismatch	
	Case-1	Case-2
E_b/R	0.8	0.8
E_g/R	0.8	0.8
g	-1	1
b	-1	1
k_b	5	5
k_g	5	5
k_v	-30	-10
ΔH	-30	-10

Table 4.6: Batch initial conditions and noise parameters

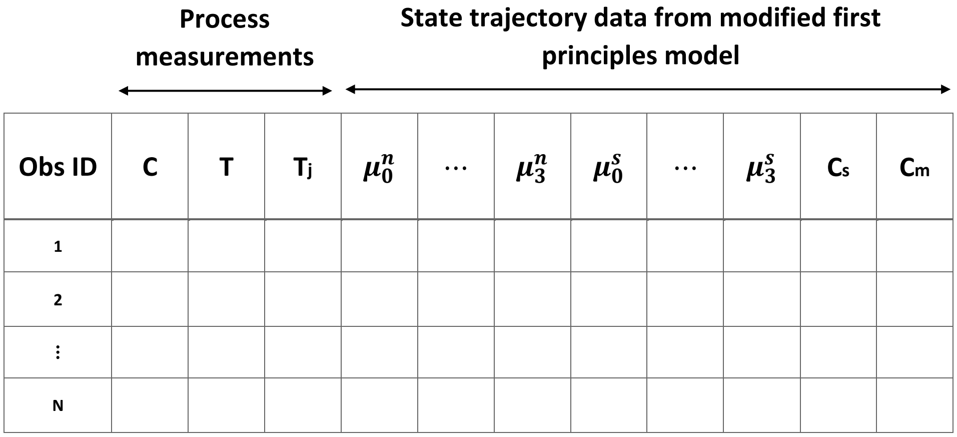
Variable	$\sigma_{batchwise}$	σ_{noise}
n	0.2	0.05
C	0.0028	0.008
T	0.52	0.003
$\mu_i^n, \mu_i^s (i = 1, 2, 3)$	-	0.001

Table 4.7: PI controller settings used in the identification and validation batches

Input policy	Number of Batches
PI trajectory tracking ($K_C = 0.85$, $T_I = 38$)	50

Table 4.8: Tuning parameters of Kalman filter used in the estimation of terminal subspace states

Model	Process noise (σ^2)	Measurement noise (σ^2)	
Subspace	0.001	0.0004	0.0004



	Process measurements			State trajectory data from modified first principles model							
Obs ID	C	T	Tj	μ_0^n	$\cdots$	μ_3^n	μ_0^s	$\cdots$	μ_3^s	Cs	Cm
1											
2											
⋮											
N											

Figure 4.11: X block of Augmented/Hybrid PLS for the crystallizer case study

Chapter 5

Application of data-driven modeling approaches to industrial hydroprocessing units

The contents of this chapter have been published in *Chemical Engineering Research and Design* Journal.

- Debanjan Ghosh, Jesús Moreira, Prashant Mhaskar, Application of data-driven modeling approaches to industrial hydroprocessing units, *Chemical Engineering Research and Design*, Volume 177, 2022, Pages 123-135, ISSN 0263-8762, https://doi.org/10.1016/j.cherd.2021.10.023.

Abstract

Hydroprocessing units in petroleum refineries comprise of several complex interconnected network of unit operations, and perform the function of removing impurities from the crude, and cracking it to lighter products for subsequent operations. Modeling these units play a pivotal role in predicting future values of important variables, improving the control and optimization of the plant for efficient operation among several other applications. This paper presents the development and implementation of data-based models in estimating product qualities and other key monitoring variables in the hydroprocessing unit of an industrial refinery. Real industrial data from two different units was used and appropriate data-driven modeling strategies were formulated in order to address this problem. In one instance, the usefulness of Dynamic-Partial Least Squares (DPLS) over Partial Least Squares (PLS) in the estimation of important variables of the unit is demonstrated. In the other instance, subspace identification methodology is found to yield a superior model. The methods used in this study can also elegantly handle the missing data problem associated with real data sets, and thus demonstrate the ability, and the importance of using the right data driven technique for specific problems in the context of Hydroprocessing refinery.

5.1 Introduction

Hydroprocessing is an indispensable unit in petroleum refining and it's efficient operation is vital to regulate environmental standards and qualities of final product. Catalyst activity in several downstream units following the hydroprocessing unit is sensitive to the incoming feed, and thus it is imperative to maintain the products processed in the hydroprocessing unit at specification. The unit comprises two major operations: a) hydrotreating, where the feed is treated with hydrogen to remove impurities like sulfur, nitrogen, metallic compounds etc., and b) hydrocracking, where the heavier feed is cracked to form lighter products by catalytic cracking and hydrogenation.

In today's highly competitive environment, extensive study and modeling of such units are gaining importance more than ever. Models play a critical role in effectively optimizing and controlling such large scale processes to achieve production targets, increase profits, reduce accidents and faults, train manpower, amongst many other applications. Development of fundamental or first principles models [3, 13, 17, 16, 21, 34] for such intricate processes is a highly arduous task and needs in-depth knowledge and expertise in this field. These physics based models are often written down as linear/ non-linear ordinary and partial differential equations, algebraic equations, etc. which describe the evolution of different process variables over a period of time. In the past, researchers have modelled the individual units involved in the process, and they typically involve a series of kinetic and correlation equations of high complexity. Estimating and calibrating the parameters of such models also involves a highly non-convex optimization which adds an additional layer of difficulty, and is undoubtedly one of the hardest steps to solve. While the utility of a good first principles model is undeniable, the online implementation of such models for monitoring and control purposes remains limited due to their intricate nature leading to long computation times and occasional failure to convergence, and more importantly, due to the difficulty in maintaining such models.

With the large amount of data being recorded in recent times and the rise in computation power of machines, purely data based models have gained a lot of attention of process modellers. Unlike the first principles models, they only make use of historical data to build relationships between the predictor and response variables. Many such approaches exist in literature where a pre-defined model structure is first assumed, and the parameters are then estimated using the database of recorded measurements.

They are fairly simple to build compared to mechanistic models but are only valid in the region of their training. Support Vector Machines (SVM) [33, 32], Artificial neural networks (ANN) [4, 12, 36], Sparse Identification of Non-linear Dynamics (SINDy) [7, 18] are some of the non-linear techniques, and have been used in the past to model highly non-linear processes. A SVM based soft sensor was presented in [32] to detect the sulfur content in the product of a Hydrodesulfurization (HDS) unit. [4] used different ANN architectures to predict the sulfur content in a similar unit. [12, 36] showed the usefulness of feed-forward neural (FNN) networks and convolutional neural networks (CNN) in predicting the outputs of a hydrocracking process. These methods however face overfitting issues when available data is noisy and limited, and strategies to avoid such problems are being actively researched.

On the same spectrum of purely data-driven techniques, linear approaches of model building like Principal Component Analysis (PCA) [28, 5], Partial Least Squares (PLS) [19, 31], Subspace identification methods (SIM) [25, 10], Koopman Operator approaches [26] etc. have been well researched, and these methods have demonstrated their suitability in modelling various continuous and batch processes. PCA, PLS are dimensionality reduction techniques and can handle correlated variables (a common occurrence in industrial processes) quite elegantly. They are inherently static models and are appropriately adapted to model batch processes. SIMs on the other hand, build Linear Time Invariant (LTI) dynamic state-space models making use of concepts of linear algebra, and are excellent for control oriented applications. They can be appropriately used for monitoring and soft sensing purposes by casting them in any state estimator. Both of these methods can handle the problem of over-fitting efficiently by suitably specifying the number of a single tuning parameter: components (PCA/PLS) and states (SIM). Practical concerns of missing data [27, 30] are also appropriately dealt with during the model building stage, and their computationally friendly structure makes them well suited for real-time implementations.

Dynamic Principal Component Analysis (DPCA) and Dynamic Partial Least Squares (DPLS) are extensions of PCA and PLS strategies, and make use of lagged time measurements to incorporate dynamic features in the model. DPCA [20] has been successfully implemented for monitoring purposes in both continuous [8] and batch processes [9]. Other methodologies of building a dynamic PCA and PLS exist in literature which focus on building dynamic inner relationships [22, 24] between the input and output scores. A Dynamic Inner Partial Least Squares (DIPLS) [11] was recently proposed which builds an outer model (between true variables and scores) derived from a dynamic inner model (between input and output scores), and its superiority over static PLS models in modeling dynamic systems was shown.

This study addresses the problem of modeling two hydroprocessing units and demonstrates the need to utilize the right tools for the process and data availability in question. The models are built with data collected over a period of time from real industrial units. The data sets for the two units have different sampling frequencies and appropriate models are chosen for the cases. Two linear models, namely, DPLS and Subspace identification are considered to model the dynamic input-output (I/O) relationship, both equipped to efficiently handle the problem missing data problem related with real data sets. In this study, the problem of building and implementing subspace models in the presence of missing data is further generalized to handle the missing data in inputs. The manuscript is structured as follows: Section 5.2 describes the hydrorprocessing unit in detail, followed by a brief introduction to PLS, DPLS, and Subspace based estimation models. In Section 5.3, the application of these models to the hydroprocessing unit is presented. The training and validation methodology of these different data based models including the proposed approach of adapting the subspace model to handle missing data in inputs are first presented and the results comparing the different methodologies are later shown. The conclusion and future work are presented in Section 5.4.

5.2 Preliminaries

This section first presents a brief overview of the hydroprocessing process, the associated units and the important variables that are either measured or estimated. Next, a summary of the data driven methods used in this study namely: PLS, DPLS and Subspace identification based estimation method are presented.

5.2.1 Hydroprocessing Unit

Crude oil has a lot of contaminants which can have detrimental effects in catalysts (poisoning, deactivation) and equipment (corrosion, fouling). Finished products coming out from the refinery's gates need to be cleaned so that they meet environmental regulations imposed by the government to avoid polluting the environment. For example, sulfur, if not removed from gasoline and diesel, can lead to acid rain [23].

Any refining stream containing C6 and heavier hydrocarbons are very likely to contain sulfur and nitrogen hidden in various organic compounds, even if these molecules come from sweet crudes. Removal of these contaminants needs to be performed chemically as they are embedded within the hydrocarbon molecules. Hydrotreating is a catalytic process mainly used to reduce undesirable contaminants from various hydrocarbon streams. This process is achieved by selectively reacting the contaminant species with hydrogen in a reactor at elevated temperatures and moderate to high pressure [1]. Hydrotreating catalysts are in general high surface area materials consisting of an active site and a promoter, which are uniformly dispersed on a support. The active component is a dispersed metal, including molybdenum sulfide with cobalt (CoMo) and nickel (NiMo) as promoters. The role of the promoter is to substantially increase the activity of the metal site. The catalyst support is normally a gamma alumina (γ-Al2O3). There are a lot of commercially available hydrotreating catalysts, with various metal and promoter content, depending on any particular application. However, most catalysts contain up to 25 wt % promoter and up to 25 wt % active components as oxides. If a catalyst contains an acid function, zeolite for example, this catalyst will contain cracking properties as well. The process of breaking down bonds in the presence of hydrogen and a catalyst is referred to as hydrocracking. NiMo catalysts are mostly used in the hydro-desulfurization of diesel fuels to produce ultra-low sulfur diesel (ULSD with $<$ 10 wppm sulfur). Hydrotreating units are amongst the most recurrent units in a refinery. These units are not only used to get fuels, such as gasoline, Jet and diesel, to meet environmental regulations, but they are also used to clean up most of the intermediate streams used to make other products, such as solvents, alkenes, alcohols, aromatics, etc.

In a refinery, hydroprocessing units are typically optimized and operated with models derived from fundamental governing laws. These models, however, tend to be time consuming to develop and maintain and difficult to handle in an optimization problem. Note also that the run-length cycle of a typical hydroprocessing unit varies from 1 to 4 years. Every new cycle requires updating the fundamental models, which becomes a time consuming and challenging exercise for process engineers that operate these units. In addition, given that the historical operation is recorded in the refinery data base (or in a cloud for more modern ones), it is worth exploring if data driven models can also be utilized. The idea is to quickly leverage historical process data from hydroprocessing units and build data driven models. With time, there will be more operational data available to refine the data driven models. Modeling with data driven approaches allows for re-tuning model parameters once more operating data becomes available. It is desirable then to develop these models in such a way that model updates are done automatically.

In this work, data driven modeling using DPLS and Subspace Identification is explored for two hydroprocessing units, Unit 1 and 2 in Figures 5.1 and 5.2, respectively.

Unit-1 is a conventional diesel hydrotreating unit that uses three types of feeds, one virgin and two cracked ones. The objective of this unit is to maximize diesel production and minimize hydrogen utilization while ensuring the catalyst is protected from deactivation. This unit is comprised of the Feed, Furnace, and Reaction and Separation sections, as described in Figure 5.1. Models are developed for prediction purposes, the output variables selected for modeling are chosen based on economic incentive, with product Diesel, product qualities, and hydrogen consumption being the main ones. For Unit-1, the input variables were selected according to fundamental understanding of the process. The intent is to capture the unit operation with as many process variables as possible while minimizing the number of inputs to avoid the risk of overfitting. Unit-2 presented in Figure 5.2, is a Diesel/Jet unit that is based on both hydrotreating and hydrocracking. This unit processes a total of five different feeds, three of them have boiling points in the Diesel and Jet range while two of them are too heavy and thus need cracking. The reaction section in Unit-2 is a combination of both hydrotreating and hydrocracking catalysts. The hydrotreating portion ensures contaminant removal (sulfur and nitrogen) while the hydrocracking one converts the heavy molecules into lighter ones that are in the Diesel and Jet range. Mode of operation depends on market conditions, with reactor temperatures and feed distributions being the main operational parameters used to define what product is maximized. Similar to Unit-1, input variables in Unit-2 were selected based on economic incentives. More details about these two units are not provided for proprietary and contractual reasons.

5.2.2 Partial Least Squares

PLS builds a predictive model involving a predictor data matrix $\mathbf{X}$ and a response variable matrix $\mathbf{Y}$ using a historical database. It is very well suited for multivariate analysis and possesses several advantages over Multiple Linear Regression (MLR),

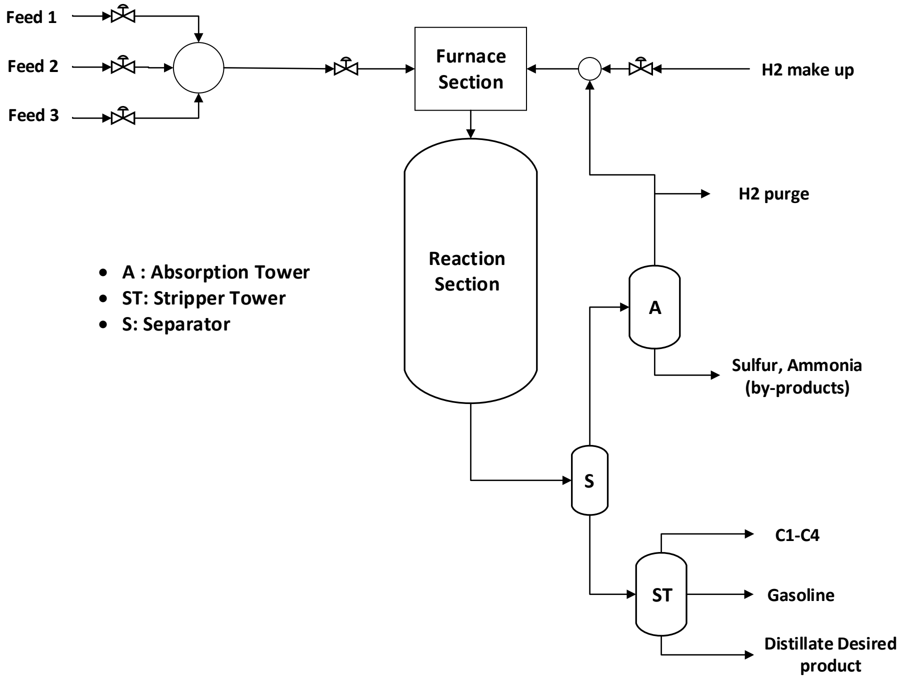


Figure 5.1: Simplified Hydrotreating Unit-1

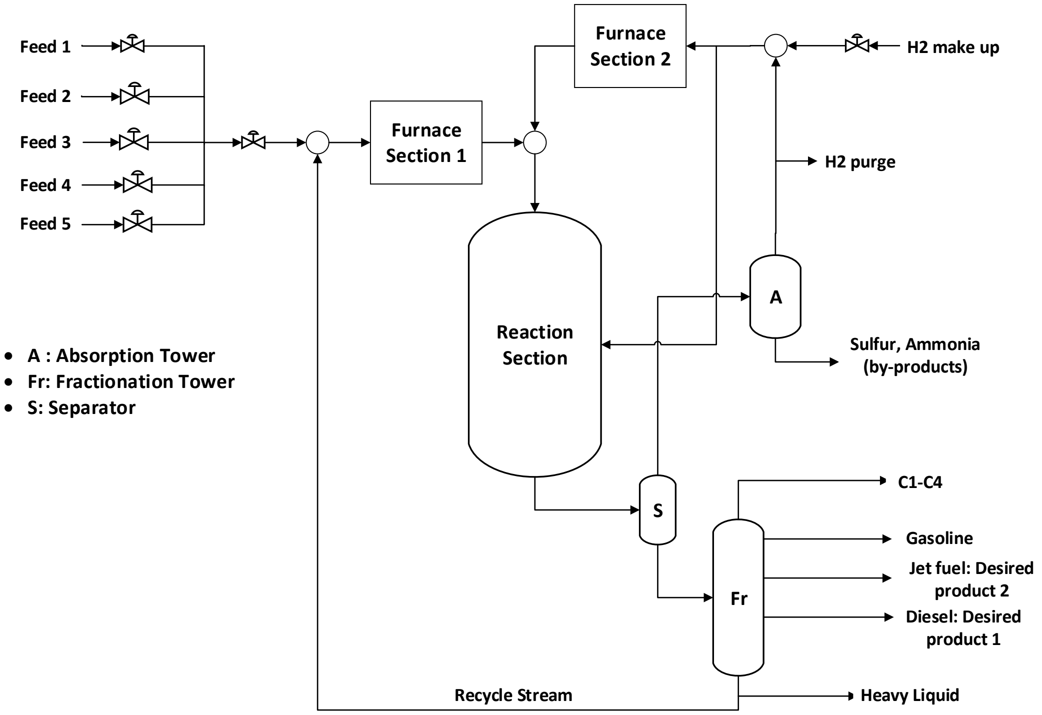


Figure 5.2: Simplified Hydrotreating Unit-2

which typically faces a rank deficiency problem due to collinearity of variables among many other practical issues. Unlike MLR, it also builds an input space error model which helps in quantifying deviations from nominal behaviour, and thus, explains the suitability of an observation in the input domain to be used for prediction. The modelling objective of PLS is 3 fold : a) best explain input or X-space b) best explain the output or Y-space c) maximize the correlation between the input and output spaces. Higher dimensional data in the original variable space is converted to latent or hidden variables (scores) having a much lower dimension, which enables much easier visualization and, at the same time, computations are less intensive. Non-linear Iterative Partial Least Squares (NIPALS) [15] is one of the algorithms that builds PLS models in a very computationally efficient manner, and is capable of handling the missing data problem. Components or the loading vectors are the link between the higher and lower dimensional space, and are fit judiciously using a cross-validation technique (on the basis of a Q^2 metric) to avoid possibilities of overfitting.

Considering the two data matrices $\mathbf{X}$ $(N \times K)$ and $\mathbf{Y}$ $(N \times M)$, the variables are first mean centered and scaled (typically to unit variance). A components are assumed to fit the model using the NIPALS algorithm. The model estimates ($\hat{\mathbf{X}}$ and $\hat{\mathbf{Y}}$) and residual matrices ($\mathbf{E}$ and $\mathbf{F}$) together comprise the original data matrices $\mathbf{X}$ and $\mathbf{Y}$ and are given by Equation (5.1). A pictorial representation of the different components in a PLS model is shown in Figure 5.3.

$$\begin{aligned} \mathbf{X} &= \hat{\mathbf{X}} + \mathbf{E} = \mathbf{T}\mathbf{P}^{\mathbf{T}} + \mathbf{E} \\ \mathbf{Y} &= \hat{\mathbf{Y}} + \mathbf{F} = \mathbf{T}\mathbf{C}^{\mathbf{T}} + \mathbf{F} \end{aligned} \tag{5.1}$$

where the matrix of scores of the X and Y space are, respectively, denoted by T $(N \times A)$ and U $(N \times A)$; the associated weight matrices are denoted by W $(K \times A)$ and C $(M \times A)$, respectively. The loading matrix used for deflation is given by P $(K \times A)$.

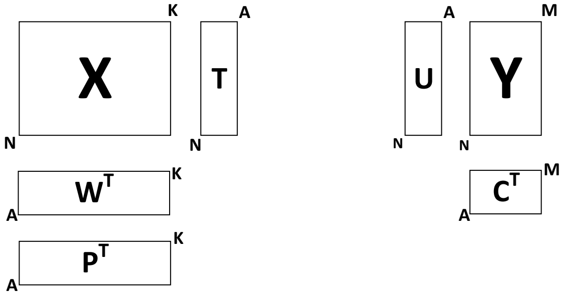


Figure 5.3: PLS model components

5.2.3 Dynamic Partial Least Squares

Standard PLS builds a static relationship between the inputs and outputs and does not explicitly account for the dynamic behaviour of the data. While dealing with dynamic data, this may lead to several inaccuracies and may not be the right tool for estimation, prediction and monitoring purposes as pointed out in [20, 9]. Auto Regressive Exogenous (ARX) models are dynamic time series models where time lagged measurements are introduced to model the autocorrelation among variables. A simple technique of incorporating the logic of ARX models into latent variable modeling to build Dynamic PCA models was proposed in [20], and later both DPCA, DPLS were implemented on batch processes in [9]. Some other techniques of incorporating dynamic information into PCA and PLS exist, however, in this study, the simple yet powerful approach of introducing lagged variables in the predictor data matrix **X** is implemented. Considering input variables $x_1, x_2, \cdots, x_K$ and output variables $y_1, y_2, \cdots, y_M$ with N observations, a typical PLS X and Y block is given by

$$\mathbf{X} = \begin{bmatrix} \mathbf{x_{1,1}} & \mathbf{x_{1,2}} & \cdots & \mathbf{x_{1,K}} \\ \mathbf{x_{2,1}} & \mathbf{x_{2,2}} & \cdots & \mathbf{x_{2,K}} \\ \vdots & \vdots & \cdots & \vdots \\ \mathbf{x_{N,1}} & \mathbf{x_{N,2}} & \cdots & \mathbf{x_{N,K}} \end{bmatrix}, \mathbf{Y} = \begin{bmatrix} \mathbf{y_{1,1}} & \mathbf{y_{1,2}} & \cdots & \mathbf{y_{1,M}} \\ \mathbf{y_{2,1}} & \mathbf{y_{2,2}} & \cdots & \mathbf{y_{2,M}} \\ \vdots & \vdots & \cdots & \vdots \\ \mathbf{y_{N,1}} & \mathbf{y_{N,2}} & \cdots & \mathbf{y_{N,M}} \end{bmatrix}$$

$\mathbf{X(i)}$, $\mathbf{Y(i)}$ are chosen to be an entire row of observation at any i^{th} instant in time and given by $\mathbf{X(i)} = \begin{bmatrix}\mathbf{x_{i,1}} & \mathbf{x_{i,2}} & \cdots & \mathbf{x_{i,K}}\end{bmatrix}$, $\mathbf{Y(i)} = \begin{bmatrix}\mathbf{y_{i,1}} & \mathbf{y_{i,2}} & \cdots & \mathbf{y_{i,M}}\end{bmatrix}$. The entries in the matrices $\mathbf{X}$, $\mathbf{Y}$ have two subscripts where the first one represents the observation number and the second one is the variable number. Considering $\mathbf{\Theta}$ lags or past measurements of both input and output variables are to be used in building the predictor data matrix of the DPLS model, an i^{th} observation of such a matrix is represented by the vector $\mathbf{X_{DPLS}(i)}$ and given as follows :

$$\mathbf{X_{DPLS}(i)} = \begin{bmatrix}\mathbf{X(i)} & \mathbf{X(i-1)} & \cdots & \mathbf{X(i-\Theta)} & \mathbf{Y(i-1)} & \mathbf{Y(i-2)} & \cdots & \mathbf{Y(i-\Theta)}\end{bmatrix}$$

Note that the above vector does not contain $\mathbf{Y(i)}$, and the reason being the output information at the i^{th} time instant won't be available while implementing this model. The input and output data matrix for the DPLS is given by:

$$\mathbf{X_{DPLS}} = \begin{bmatrix}\mathbf{X_{DPLS}(\Theta+1)} \\ \mathbf{X_{DPLS}(\Theta+2)} \\ \vdots \\ \mathbf{X_{DPLS}(N)}\end{bmatrix}, \mathbf{Y_{DPLS}} = \begin{bmatrix}\mathbf{Y(\Theta+1)} \\ \mathbf{Y(\Theta+2)} \\ \vdots \\ \mathbf{Y(N)}\end{bmatrix}$$

Once the two matrices are created, the same NIPALS algorithm is used and all the computations follow the same principle like the standard PLS.

5.2.4 Subspace Identification

Subspace identification [25, 29, 37] is a system identification method which uses past process input-output measurement data to identify a discrete linear time invariant (LTI) state-space model. It typically distinguishes the manipulated inputs from the controlled variables and it's structure is very well suited for handling and controlling Multi-Input Multi-Output (MIMO) systems. Unlike another class of system identification technique called Prediction Error Minimization (PEM), SIMs avoid any non-convex optimization problems to estimate the parameters. It invokes concepts of linear algebra and makes rigorous use of computationally efficient and robust matrix computations to identify the parameters. The identified state-space structure is given by the form presented in Equations (5.2) and (5.3).

$$\mathbf{z}[i+1] = \mathbf{A}\mathbf{z}[i] + \mathbf{B}\mathbf{u}_{\mathbf{SIM}}[i] \tag{5.2}$$

$$\mathbf{y}_{\mathbf{SIM}}[i] = \mathbf{C}\mathbf{z}[i] + \mathbf{D}\mathbf{u}_{\mathbf{SIM}}[i] \tag{5.3}$$

where $\mathbf{z}[i] \in \mathbb{R}^{n\times 1}$ denotes the vector of subspace states at the i^{th} instant, and n represents the number of states which is also the identified order of the model. The inputs to the model are represented by the vector $\mathbf{u}_{\mathbf{SIM}}[i] \in \mathbb{R}^{\mathbf{K}\times 1}$, whereas the outputs are given by $\mathbf{y}_{\mathbf{SIM}}[i] \in \mathbb{R}^{\mathbf{M}\times 1}$. The number of measured inputs and outputs are, respectively, given by $\mathbf{K}$ and $\mathbf{M}$. The dimensions of the identified system matrices are as follows: $\mathbf{A} \in \mathbb{R}^{\mathbf{n}\times\mathbf{n}}$, $\mathbf{B} \in \mathbb{R}^{\mathbf{n}\times\mathbf{K}}$, $\mathbf{C} \in \mathbb{R}^{\mathbf{M}\times\mathbf{n}}$, $\mathbf{D} \in \mathbb{R}^{\mathbf{M}\times\mathbf{K}}$. Note that in the present description of subspace modeling, standard notations are not used to maintain consistency and avoiding overlap with notations/variable names with the DPLS model.

An excellent approach [25] makes use of Singular Value Decomposition (SVD) of Hankel matrices (appropriately constructed using input and output data) to evaluate the parameters of model. In that approach, the training process first involves identifying the subspace state trajectory for the training data set, and the state-space system matrices are then determined using a least square error minimization computation. An adaptation of this technique [30] is, however, implemented in the present application as it is robust to missing data which makes it well suited for handling industrial data sets. It follows the same sequence of steps as the original approach [25], but the computations are carried out in the latent variable domain which provides an ideal platform to counter issues involving missing data. For the sake of brevity, details regarding the approach are not included in the present manuscript, and the readers are referred to the original sources [25, 30].

Identification involves first finding a valid state sequence from the I/O Hankel matrices using linear algebra based projection algorithms and algebraic manipulations. The algorithm to find the state sequence is succinctly described below, for details on the algorithm and the construction of the Hankel matrices, please see [25, 30].

The first part of the algorithm involves a PCA step using the NIPALS algorithm on the future input Hankel matrix followed by a deflation step which removes all the variance in the past inputs, outputs and future outputs that can be explained by the future inputs. This results in a form where the future outputs is influenced by the current states without the influence of inputs. Note that in traditional subspace identification, the above step is accomplished by projecting the future outputs onto a space perpendicular to the future inputs. Presence of missing data in the outputs limits the implementation in the conventional way whereas NIPALS can handle this step quite elegantly. The next crucial step involves identifying the relationship between the past inputs, outputs and future outputs which paves the way for finding a valid state sequence. A NIPALS PLS is used in this algorithm to relate the past inputs, outputs with the future outputs. The loadings and weights obtained from the PLS step is used to obtain an expression which is the product of the extended observability matrix and a state trajectory matrix. This expression is further decomposed by a NIPALS PCA step to determine a valid realization of the state trajectory. Again, in conventional subspace identification, a SVD is typically employed to perform this decomposition, however, missing data in the outputs renders it unimplementable. Once the state trajectory has been obtained, the final step is to determine the system matrices A,B,C and D, typically computed using linear regression.

The generic I/O state-space relationship can be represented by Equation (5.4) as :

$$\begin{bmatrix} \bar{\mathbf{Z}}_{r+1} \\ \bar{\mathbf{Y}}_r \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{Z}}_r \\ \bar{\mathbf{U}}_r \end{bmatrix} \tag{5.4}$$

where $\bar{\mathbf{Z}}_r = \begin{bmatrix} \mathbf{z}[r] & \mathbf{z}[r+1] & \cdots & \mathbf{z}[r+j-2] \end{bmatrix}$ is the state trajectory matrix obtained from an earlier step in the identification procedure, $\bar{\mathbf{Z}}_{r+1} = \begin{bmatrix} \mathbf{z}[r+1] & \mathbf{z}[r+2] & \cdots & \mathbf{z}[r+j-1] \end{bmatrix}$ is the one-step shifted state trajectory matrix. $\bar{\mathbf{Y}}_r = \begin{bmatrix} \mathbf{y}_{\mathbf{SIM}}[r] & \mathbf{y}_{\mathbf{SIM}}[r+1] & \cdots & \mathbf{y}_{\mathbf{SIM}}[r+j-2] \end{bmatrix}$ and $\bar{\mathbf{U}}_i = \begin{bmatrix} \mathbf{u}_{\mathbf{SIM}}[r] & \mathbf{u}_{\mathbf{SIM}}[r+1] & \cdots & \mathbf{u}_{\mathbf{SIM}}[r+j-2] \end{bmatrix}$ are, respectively, the output and input trajectory matrix. r and j, respectively, denote the number of rows and columns of the Hankel matrices [25, 30] and are user defined parameters. The regressor matrix $\begin{bmatrix} \bar{\mathbf{Z}}_r \\ \bar{\mathbf{U}}_r \end{bmatrix}$ and the regressand matrix $\begin{bmatrix} \bar{\mathbf{Z}}_{r+1} \\ \bar{\mathbf{Y}}_r \end{bmatrix}$ in eq. (5.4) are both known quantities, and the coefficient matrix $\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}$ can be easily identified using least-squares regression.

SIM models are very well suited for control and optimization problems where the inputs are typically manipulated inputs, and the values are either computed or known in advance to the practitioner. One important condition in performing the regression using Equation (5.4) is that the regressor matrix should be devoid of any correlated inputs and missing values. For applications (inluding the ones in the present manuscript) where there exists correlation and missing data in the inputs, the linear regression step has to be appropriately modified to be able to handle such bottlenecks. The present manuscript addresses this problem by resorting to PLS using the NIPALS algorithm [15]. The ability of PLS modeling strategy using the well established NIPALS algorithm to effectively compute the coefficients in the presence of correlation and missing data makes it a great choice for such situations.

In order to use these SIM models for soft-sensing and one step head prediction purposes, the subspace identified system matrices can be appropriately embedded in any state estimator routine to generate the estimated outputs. In this work, a Kalman filter [35] is chosen for estimating the states, but any other standard state estimator can be readily used for this purpose as well. Process, measurement noise variances and an initial estimate of the state co-variance matrix is needed to initialize the algorithm. The mechanism follows the set of equations provided in Equations (5.5) to (5.7) to predict the one step ahead states, outputs and state co-variance matrix. The Kalman gain is calculated using Equation (5.8). Equation (5.9) is used to correct the states once the output measurements are available. The state co-variance matrix is updated using Equation (5.10).

$$\hat{\mathbf{z}}^{\mathbf{pred}}[i+1] = \mathbf{A}\hat{\mathbf{z}}^{\mathbf{corr}}[i] + \mathbf{B}\mathbf{u}_{\mathbf{SIM}}[i] \tag{5.5}$$

$$\hat{\mathbf{y}}_{\mathbf{SIM}}[i+1] = \mathbf{C}\hat{\mathbf{z}}^{\mathbf{pred}}[i+1] + \mathbf{D}\mathbf{u}_{\mathbf{SIM}}[i+1] \tag{5.6}$$

$$\mathbf{G}^{\mathbf{pred}}[i+1] = \mathbf{A}\mathbf{G}^{\mathbf{corr}}[i]\mathbf{A}^T + \mathbf{I} \tag{5.7}$$

$$\mathbf{L}[i+1] = \mathbf{G}^{\mathbf{pred}}[i+1]\mathbf{C}^T\mathbf{A}^T.(\mathbf{C}\mathbf{G}^{\mathbf{pred}}[i+1]\mathbf{C}^T + \mathbf{J})^{-1} \tag{5.8}$$

$$\hat{\mathbf{z}}^{\mathbf{corr}}[i+1] = \hat{\mathbf{z}}^{\mathbf{pred}}[i+1] + \mathbf{L}[i+1](\mathbf{y}_{\mathbf{SIM}}[i+1] - \hat{\mathbf{y}}_{\mathbf{SIM}}[i+1]) \tag{5.9}$$

$$\mathbf{G}^{\mathbf{corr}}[i+1] = \mathbf{G}^{\mathbf{pred}}[i+1] - \mathbf{L}[i+1]\mathbf{C}\mathbf{G}^{\mathbf{pred}}[i+1] \tag{5.10}$$

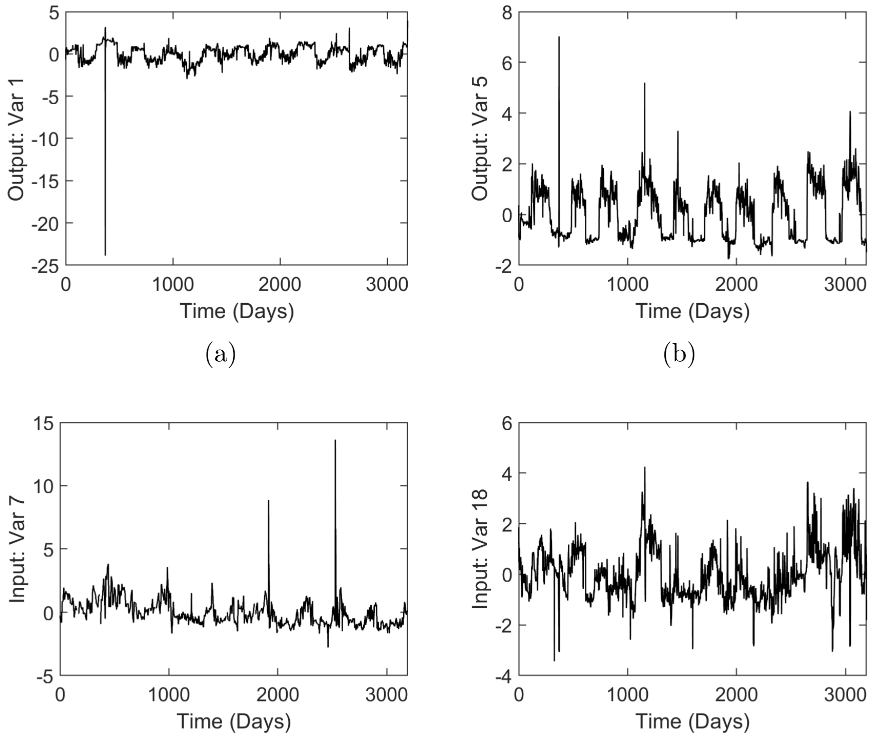
where $\hat{\mathbf{z}}^{\mathbf{pred}}[i+1]$, $\hat{\mathbf{y}}_{\mathbf{SIM}}[i+1]$ and $\mathbf{G}^{\mathbf{pred}}[i+1]$ are the one step ahead estimated states, outputs and the state co-variance matrix. $\hat{\mathbf{z}}^{\mathbf{corr}}[i+1]$ refer to the corrected states, $\mathbf{G}^{\mathbf{corr}}[i+1]$ is the updated state co-variance matrix, and $\mathbf{L}[i+1]$ is the Kalman gain. They are calculated using the information of system matrices and the variances of the process and measurement uncertainties (user defined parameters). $\mathbf{I}$ and $\mathbf{J}$ are diagonal matrices, respectively, having the process and measurement noise variances along the diagonal elements.

5.3 Application to hydroprocessing unit

The data sets pertaining to both the units contain measured data values of variables gathered from different equipment and locations in the unit. These variables are distinguished as inputs and outputs typically based on their role in the units, ease of availability and their future use in control and optimization schemes. There were a total of 18 input variables and 11 output variables for Unit-1, and 30 input variables and 18 output variables for Unit-2. They broadly constitute the characteristics and operating profiles in the feed, reaction and product section. Var 1 (mentioned in Table 5.2) of Unit-1 is a desirable product, and Var 7, Var 8, Var 9 are some of the product qualities. In Unit-2, Var 8 and Var 9 (mentioned in Table 5.3) are our desirable fuels, Var 11 refers to the volume increase, and Var 12 is the hydrogen requirement. Unit-1 data set has variable data recorded daily whereas Unit-2 has an hourly measurement frequency. In the following subsections, the training and validation methodology of the different data-driven techniques employed are explained and the simulation results are alongside presented. ASPEN ProMV and Matlab R2020a were used to perform all the simulations and other calculations. Some representative input and output variable profiles (mean centered and scaled) of Unit-1 are shown in Figure 5.4.

5.3.1 Data preprocessing

The data sets were first preprocessed to remove dubious observations. Certain variable values which could easily be identified as outliers were removed by looking at the variable time series data. In the case when most of the variables have arbitrary values at any time instant, entire rows pertaining to that instant were completely removed. It should, however, be highlighted that in an ideal case when relevant information regarding the happenings during that period can be gathered (information from plant



(a)

(b)

(c)

(d)

Figure 5.4: Figures (a),(b) represent time series profiles of two candidate outputs of Unit-1. (c),(d) similarly represent two such input profiles of Unit-1

operators or other sources), proper judgement calls can be made to either include or omit such observations. A PLS model is first built using this trimmed data set. The Hoteling T^2, Squared Prediction Error (SPE) statistic and contribution plots in PLS models were also used to further remove outliers. The final data set for Unit-1 to be used for modeling has missing variable values at certain instances due to either unavailable sensor readings or trimming of outliers. Unit-2 does not contain any missing variables.

5.3.2 Model Identification

Unit-1: The cleaned data set was used for model building. A PLS model initially built with the input and output data did not yield good results, and hence, is not presented for the sake of brevity. The reason can be attributed to the fact that, although measurement frequency of the variables in this data set is not frequent, the dynamics do not seem to die down and can be modelled with lagged information. Another way to interpret this is that there is a presence of time lags in the system such that inputs take some time to impact the process outputs.

To enable accounting for the effect of past values of the variables on the present output, a DPLS modeling strategy was then considered and trained with the same cleaned data set. Different models were built with each having a distinct number of lags for all the variables. A DPLS model of lag 1 was finally chosen on the basis of two reasons: 1) lower RMSE values from the validation data set 2) parsimony in the number of predictor variables (an important consideration for end use of the models by practitioners). The various identification aspects related to DPLS with lag 1 (referred to as DPLS1 from here onward) is presented in Table 5.1.

The DPLS1 model considered in this study has the form :

$$\text{Response matrix} = \text{Predictor matrix} * \text{Co-efficient matrix}$$

$$\begin{bmatrix} \mathbf{Y(2)} \\ \mathbf{Y(3)} \\ \vdots \\ \mathbf{Y(i)} \\ \vdots \\ \mathbf{Y(N)} \end{bmatrix} = \begin{bmatrix} \mathbf{X_{DPLS}(2)} \\ \mathbf{X_{DPLS}(3)} \\ \vdots \\ \mathbf{X_{DPLS}(i)} \\ \vdots \\ \mathbf{X_{DPLS}(N)} \end{bmatrix} * \beta, \text{ where * refers to matrix multiplication}$$

Identification Specifications: Unit-1	
Number of Input variables (K)	18
Number of Output variables (M)	11
Number of Observations (N) for training	2561
Number of chosen lags (Θ)	1
Number of Predictor variables	47 (18 Input variables + 18 Input variables with lag 1 + 11 Output variables with lag 1)
Number of Response variables	11
Identification Specifications: Unit-2	
Number of Input variables (K)	30
Number of Output variables (M)	18
Number of Observations (N) for training	11876
Number of chosen lags (Θ)	1
Number of Predictor variables	78 (30 Input variables + 30 Input variables with lag 1 + 18 Output variables with lag 1)
Number of Response variables	18

Table 5.1: Variable information used in modeling DPLS1 for Unit-1 and Unit-2

The Response matrix, Predictor matrix, and the Co-efficient matrix has, respectively, dimensions of $\mathbf{N} \times \mathbf{M}$, $\mathbf{N} \times (\mathbf{K} + \boldsymbol{\Theta}\mathbf{K} + \boldsymbol{\Theta}\mathbf{M})$, $(\mathbf{K} + \boldsymbol{\Theta}\mathbf{K} + \boldsymbol{\Theta}\mathbf{M}) \times \mathbf{M}$. A particular row in the Predictor ($\mathbf{X_{DPLS}(i)}$) and Response ($\mathbf{Y(i)}$) data matrices is given by

$$\mathbf{X_{DPLS}(i)} = \begin{bmatrix} \mathbf{X(i)} & \mathbf{X(i-1)} & \mathbf{Y(i-1)} \end{bmatrix}, \text{ where } \mathbf{X(i)} = \begin{bmatrix} \mathbf{x_{i,1}} & \mathbf{x_{i,2}} & \cdots & \mathbf{x_{i,K}} \end{bmatrix}$$

$$\mathbf{Y(i)} = \begin{bmatrix} \mathbf{y_{i,1}} & \mathbf{y_{i,2}} & \cdots & \mathbf{y_{i,M}} \end{bmatrix}$$

β, the matrix of co-efficients is given by

$$\beta = \begin{bmatrix} \mathbf{B}^1_{\mathbf{X}} & \mathbf{B}^2_{\mathbf{X}} & \cdots & \mathbf{B}^j_{\mathbf{X}} & \cdots & \mathbf{B}^M_{\mathbf{Xlag}} \\ \mathbf{B}^1_{\mathbf{Xlag}} & \mathbf{B}^2_{\mathbf{Xlag}} & \cdots & \mathbf{B}^j_{\mathbf{Xlag}} & \cdots & \mathbf{B}^M_{\mathbf{Xlag}} \\ \mathbf{B}^1_{\mathbf{Ylag}} & \mathbf{B}^2_{\mathbf{Ylag}} & \cdots & \mathbf{B}^j_{\mathbf{Ylag}} & \cdots & \mathbf{B}^M_{\mathbf{Ylag}} \end{bmatrix}$$

where $\mathbf{B}^j_{\mathbf{X}} = \begin{bmatrix} \mathbf{b}^j_{\mathbf{x,1}} & \mathbf{b}^j_{\mathbf{x,2}} & \cdots & \mathbf{b}^j_{\mathbf{x,K}} \end{bmatrix}^T$, $\mathbf{B}^j_{\mathbf{Xlag}} = \begin{bmatrix} \mathbf{b}^j_{\mathbf{xlag,1}} & \mathbf{b}^j_{\mathbf{xlag,2}} & \cdots & \mathbf{b}^j_{\mathbf{xlag,K}} \end{bmatrix}^T$
and $\mathbf{B}^j_{\mathbf{Ylag}} = \begin{bmatrix} \mathbf{b}^j_{\mathbf{ylag,1}} & \mathbf{b}^j_{\mathbf{ylag,2}} & \cdots & \mathbf{b}^j_{\mathbf{ylag,M}} \end{bmatrix}^T$

A total of 6 components is used to fit the model using the Autofit option (based on cross-validation) in ASPEN ProMV and the β matrix is identified. In general, the R^2 value corresponding to the a^{th} component for the Y-space is calculated using Equation (5.11).

$$R^2_a = 1 - \frac{\text{Variance(Outputs - Model Predicted Outputs)}}{\text{Variance(Original output data matrix)}} \tag{5.11}$$

Q^2 is a metric similar to R^2 which is evaluated during the cross-validation stage. It is an indicator of the predictive ability of each component and is considered a metric to choose the best number of components. The training data is divided into G number of groups during cross validation, and $F_1, F_2, \cdots F_G$, respectively, represent the residuals (difference between the output values considered and the predicted ones) corresponding to each of the testing groups used in this procedure. For the a^{th} component, it is calculated by Equation (5.12).

$$Q^2_a = 1 - \frac{\text{PRESS}}{\text{Variance(Original output data matrix)}} \tag{5.12}$$

where the Prediction Error Sum of Squares (PRESS) is given by

$$\text{PRESS} = \text{sum of square}(F_1) + \text{sum of square}(F_2) + \cdots + \text{sum of square}(F_G)$$

The R^2 and Q^2 values of the output space for the DPLS1 model is shown in Figure 5.5. The Q^2 value increases with addition of components till the 6^{th} one, and the Autofit option in ASPEN ProMV renders this value as the best number of latent variables to fit the model.

Subspace identification is the second method that was considered to capture the dy-

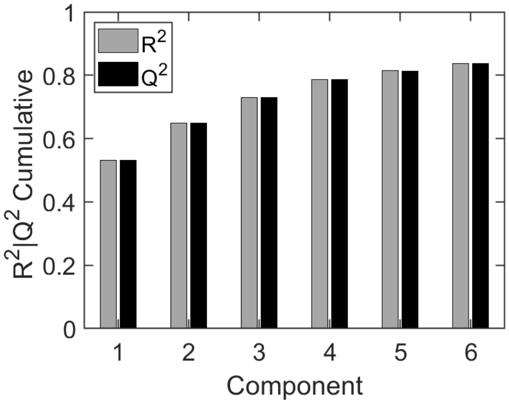


Figure 5.5: The R^2 and Q^2 metrics for the Y-space of Unit-1 using the DPLS1 model

namics of the process. The model was built with the same trimmed training data set as used in the DPLS1 model. The input and output variables were also considered the same. Note that the inputs here are the original variables and not the predictor variables used in the DPLS1 model. A total of 6 states were chosen to build the model using the methodology proposed in [30] and the modification proposed in Section 5.2.4. A PLS regression using the NIPALS algorithm with 6 components was used to identify the system matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ due to the presence of missing data in the input profiles.

Unit-2: The data set for Unit-2 contains hourly measured variable data with 30 input variables and 18 output variables. The data set was pre-processed in a similar fashion as Unit-1. A DPLS lag1 was built for this case as well and some information related to variables used for identification is presented in Table 5.1. A DPLS lag1 model having 9 components was fit using the Autofit option in ASPEN ProMV.

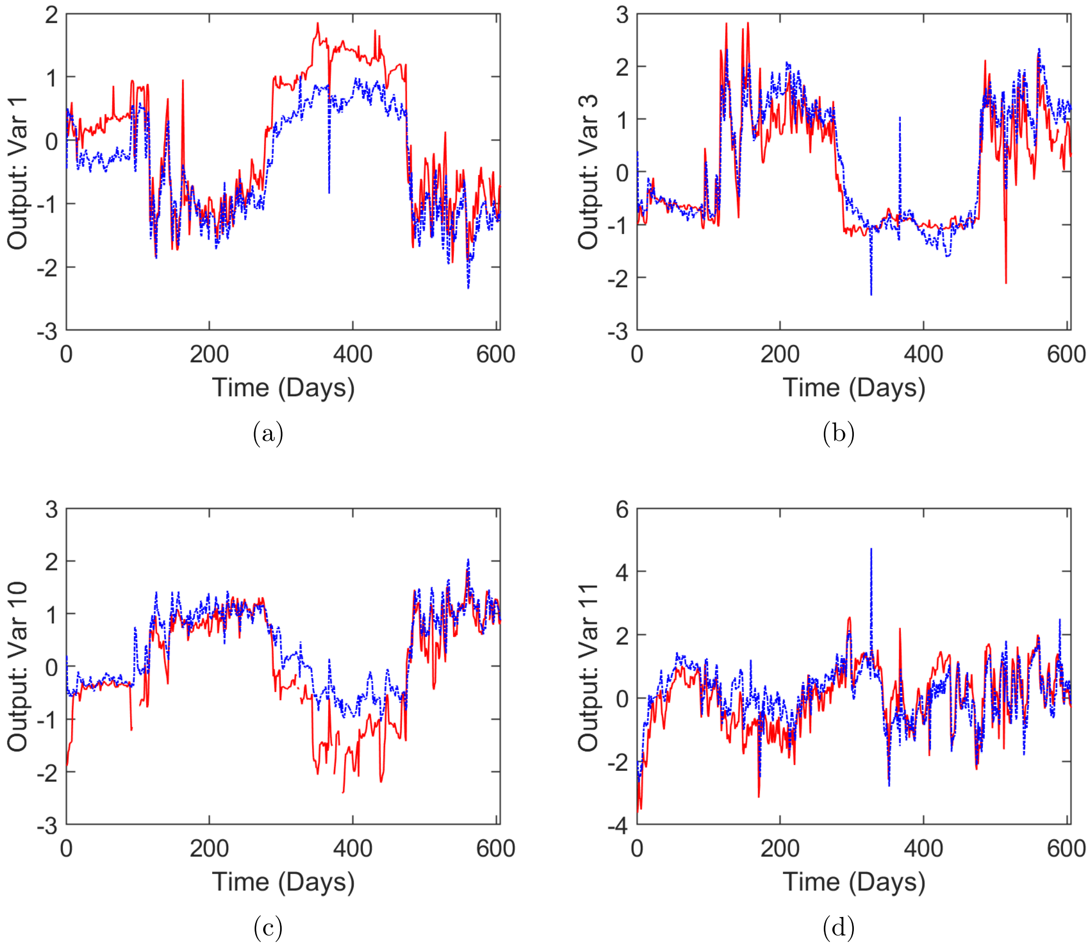
Subspace identification is also considered to model Unit-2, and is expected to perform well owing to the fact that the measurements are recorded with a high frequency. The same pre-processed training data set used in the DPLS1 model was used in the identification of the model matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$, and a total of 8 states were chosen to build them. Unlike the Unit-1 data set, Unit-2 data set was devoid of any missing values and hence, the PLS step proposed in Section 5.2.4 was not necessary. The traditional way of using MLR to generate the matrices was used in this step.

5.3.3 Model validation results

In this study, the problem of estimating the outputs is considered when the input data is readily available. The models identified for both the units were validated on their respective testing data sets. The validation approaches are described and the modeling results for each of the units are presented below.

Unit-1: DPLS1 model of Unit-1 is validated on 606 observations. A check for X-space data for outlier detection is also done using the Hoteling T^2 and SPE statistic. At any time instant, the entire row of observation pertaining to X-space predictors is considered to be available and the identified coefficients from training stage were used to generate the outputs. The validation results of some important variables, from the fractionation product and reaction region, are presented in Figure 5.6. All the outputs are mean centered and scaled prior to plotting. The Root Mean Square Error (RMSE) values for all the outputs are presented in Table 5.2. In general, the trend of the predicted outputs by the DPLS model in Figure 5.6 suggests that the model was able to capture the dynamics relatively well. There are mismatch at certain times with the observed values and this could be ascribed to the low measurement frequency of variables, highly non-linear behaviour or unavailability of certain predictor variables.

The subspace model is validated on the same observations as the DPLS1 model. The missing data in the input measurements for the validation data set, however, needs to be tackled first prior to using these values for prediction of outputs using the state-space model. This problem was not addressed in [30], which assumed that the data for validation did not include missing input values.



(a)

(b)

(c)

(d)

Figure 5.6: The validation results for 4 outputs of Unit-1 are presented in figures (a), (b), (c) and (d), respectively. The Process/observed values are given by the Red $-$ line and the model prediction of DPLS1 by Blue $-.$ line

The problem of model validation in the presence of missing input values is addressed in the present manuscript as follows: A PCA step using the NIPALS [15] algorithm is first performed on the training input data matrix to get the loading matrix. The generic step is presented in Equation (5.13).

$$\mathbf{X_{train}} = \mathbf{\hat{X}_{PCA}} + \mathbf{E_{PCA}} = \mathbf{T_{PCA}} \mathbf{P_{PCA}^{T}} + \mathbf{E_{PCA}} \tag{5.13}$$

$\mathbf{X_{train}}$ represents the mean centered and scaled input training data matrix; $\mathbf{\hat{X}_{PCA}}$

represents the PCA model estimate of the same where $\mathbf{T_{PCA}}$ and $\mathbf{P_{PCA}}$ are the associated scores and loading matrices; $\mathbf{E_{PCA}}$ represents the residual not explained by the model.

The main aim of resorting to PCA is to generate an estimate of the validation observations. A new validation observation is first mean centered and scaled using the values used in the training step of the PCA model. For any i^{th} observation or row of the validation data matrix $\mathbf{X_{valid}}$, the score $\mathbf{t^i_{valid}}$ using the first loading vector is calculated by the Equation (5.14) using the Single Component Projection (SCP) algorithm [27].

$$\mathbf{t^i_{valid}} = (\mathbf{\tilde{X}^i_{valid}\tilde{p_1}})/\mathbf{\tilde{p_1}}^T\mathbf{\tilde{p_1}} \tag{5.14}$$

$\mathbf{\tilde{X}^i_{valid}}$ is the i^{th} observation without the missing values and $\mathbf{\tilde{p_1}}$ is the first loading vector from the matrix $\mathbf{P_{PCA}}$ having the elements corresponding to the non-missing values of the observation.

The observation is then deflated to remove any variance explained by the first component, and the next element in the score vector of the i^{th} new observation is calculated using the second loading vector from $\mathbf{P_{PCA}}$ in a similar fashion. Once the entire score matrix $\mathbf{T_{valid}}$ for all the observations is generated, the estimate of the validation input matrix is constructed using the Equation (5.15).

$$\mathbf{\hat{X}_{Valid}} = \mathbf{T_{valid}P^T_{PCA}} \tag{5.15}$$

The estimated input data matrix $\mathbf{\hat{X}_{Valid}}$ from the PCA step is used in the state-space model equations to determine the outputs. The number of components for the PCA step in the present implementation were chosen to be 2 using trial and error as it provided the lowest RMSE values when used in the subspace model for validation. In this case, the estimated values of only the missing values were retained, and rest of the values were kept the same as the original variables. One can, however, depending on the problem at hand, use this PCA step efficiently to reduce noise in the input data matrix and use the estimated values for the non-missing inputs as well. Other efficient ways of handling the problem of missing data in the inputs of state-space models while validating remains the subject of future work.

A Kalman filter was chosen in this study and the identified model was cast into the Kalman framework to estimate the outputs. The algorithm is initialized with the aid of the following information :

- Identified model matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$
- An initial estimated state vector guess : $\hat{\mathbf{z}}^{init}$ is considered to be a vector of zeros of size n×1, where n=6 is the identified model order. $\hat{\mathbf{z}}^{corr}[1] = \hat{\mathbf{z}}^{init}$ at initialization
- Entire row of initial value of the inputs comprising of original variables and the estimated values from the PCA step in case of any missing data
- An identity matrix of size (n × n) is chosen as the initial estimate of the state co-variance matrix
- Kalman tuning parameters chosen as follows: Process noise $(\sigma^2) = 0.006 \times$ ones(n × 1); Measurement noise $(\sigma^2) = 0.001 \times$ ones($\mathbf{M} \times 1$). The values were chosen by trial and error with the objective of reducing the estimation error, time needed for convergence while avoiding large initial jumps in the state values

The states at for any instant i are predicted using the corrected states $(\hat{\mathbf{z}}^{\mathbf{corr}}[i-1])$ and inputs $(\mathbf{u_{SIM}}[\mathbf{i}-\mathbf{1}])$ by the following relation

$$\hat{\mathbf{z}}^{\mathbf{pred}}[i] = \mathbf{A}\hat{\mathbf{z}}^{\mathbf{corr}}[i-1] + \mathbf{B}\mathbf{u_{SIM}}[i-1] \tag{5.16}$$

Once the inputs at the instant i are available, the estimated outputs during the validation run are computed as follows

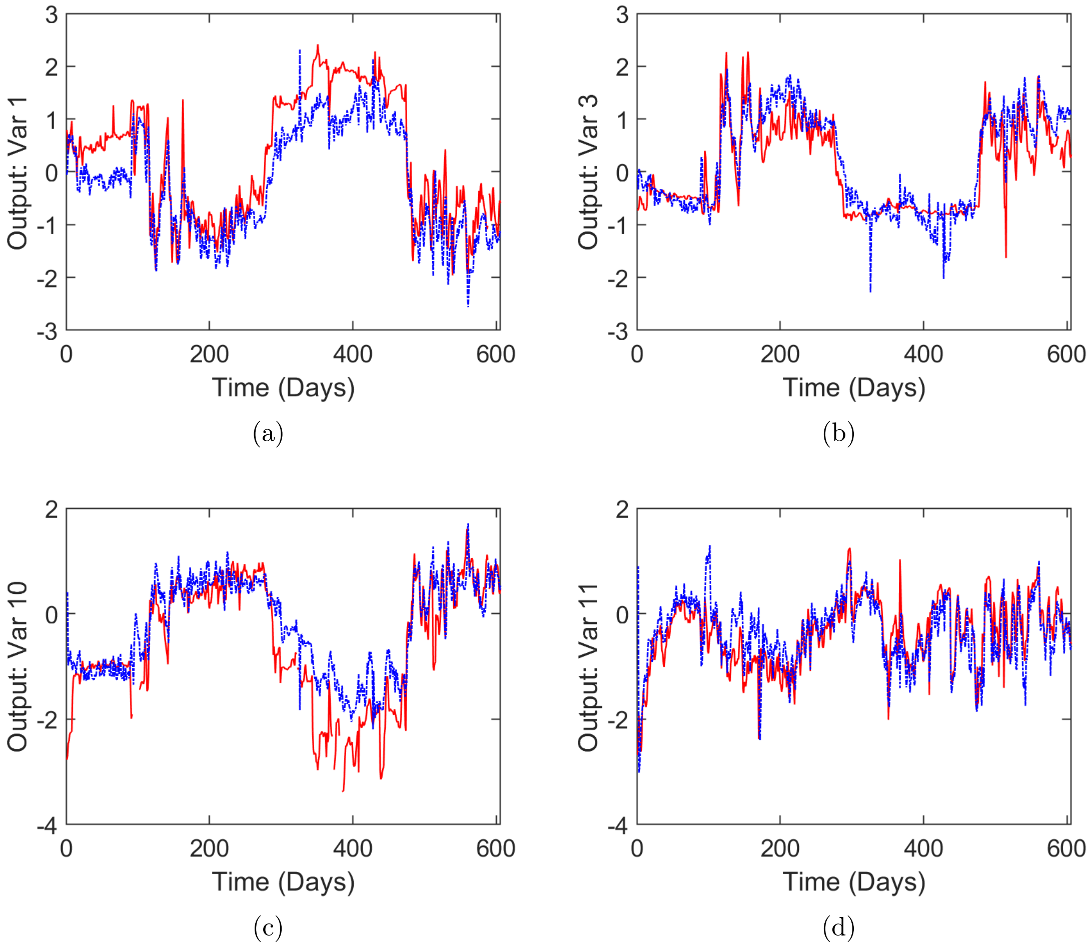
$$\hat{\mathbf{y}}_{\mathbf{SIM}}[i] = \mathbf{C}\hat{\mathbf{z}}^{\mathbf{pred}}[i] + \mathbf{D}\mathbf{u}_{\mathbf{SIM}}[i] \tag{5.17}$$

where $\mathbf{u}_{\mathbf{SIM}}[\mathbf{i}] = \begin{bmatrix} \mathbf{x}_{\mathbf{i},\mathbf{1}} & \mathbf{x}_{\mathbf{i},\mathbf{2}} & \cdots & \mathbf{x}_{\mathbf{i},\mathbf{K}} \end{bmatrix}^T$ is the vector of inputs comprising of original and estimated values (computed using the PCA in case of any missing data) at the i^{th} instant. The states are updated when the new output measurements ($\mathbf{y}[i]$) are registered and given by

$$\hat{\mathbf{z}}^{\mathbf{corr}}[i] = \hat{\mathbf{z}}^{\mathbf{pred}}[i] + \mathbf{L}(\mathbf{y}_{\mathbf{SIM}}[i] - \hat{\mathbf{y}}_{\mathbf{SIM}}[i]) \tag{5.18}$$

where $\mathbf{y}_{\mathbf{SIM}}[\mathbf{i}] = \begin{bmatrix} \mathbf{y}_{\mathbf{i},\mathbf{1}} & \mathbf{y}_{\mathbf{i},\mathbf{2}} & \cdots & \mathbf{y}_{\mathbf{i},\mathbf{M}} \end{bmatrix}^T$ is the vector of measured outputs. The role of the output measurements is just to correct the states, which then can be used to predict the states at the next instant. The equations involving the calculation of the Kalman gain matrix along with the propagation state co-variance matrix for the case considered is not shown for the sake of brevity. The outputs estimated by the algorithm ($\hat{\mathbf{y}}[i]$) are compared with the true values ($\mathbf{y}[i]$), and profiles of some of the outputs are shown in Figure 5.7. It should be highlighted that for cases, when there is missing data in the output measurements, only the state prediction step (shown in Equation (5.16)) is used to compute the estimated outputs and the update step (shown in Equation (5.18)) is skipped. The RMSE values of all the outputs for the entire validation test data is presented in Table 5.2. Similar to the DPLS model, the Subspace estimation was also able to predict the trends in the output variables, most of the time, with a relatively good accuracy.

Unit-2: The models chosen for Unit-2 are validated on 793 observations. The DPLS1 model is validated in a similar fashion as the one in Unit-1. The validation results for some outputs are presented in Figure 5.8.



(a) (b) (c) (d)

Figure 5.7: The validation results for 4 outputs of Unit-1 are presented in figures (a), (b), (c) and (d), respectively. The Process/observed values are given by the Red − line and the model prediction of Subspace estimation by Blue −. line

The same pre-processed validation data set used for the DPLS1 model was also chosen for the Subspace identification based estimation method. The filter is initiated in a similar fashion as in Unit-1. The initial value of the state vector, the state-covariance matrix are also kept the same as in Unit-1. The dimensions of these vectors and matrices are, however, different and in accordance to the information provided in identification section and Table 5.1 for Unit-2. The Kalman tuning parameter values for the process noise and measurement noise variances are, respectively, Process noise $(\sigma^2) = 0.00022 \times \text{ones}(\text{n} \times 1)$; Measurement noise $(\sigma^2) = 0.001 \times \text{ones}(\mathbf{M} \times 1)$.

	Output	DPLS1	Subspace estimation
Fractionation region	Var 1	0.02105	0.02164
	Var 2	0.01705	0.01649
	Var 3	0.00287	0.00331
	Var 4	0.00042	0.00052
	Var 5	0.00077	0.00096
	Var 6	0.00266	0.00323
	Var 7	0.00515	0.00595
	Var 8	0.54562	0.56415
	Var 9	1.74497	2.09491
Reaction region	Var 10	23.81970	23.35889
	Var 11	11.09420	11.96253

Table 5.2: RMSEP values of different outputs of Unit-1

The profiles of some of the estimated and true outputs are shown in Figure 5.9, and the RMSE values computed for the two modeling strategies are presented in Table 5.3.

Both the DPLS1 and subspace identification methodology were able to estimate the outputs with good accuracy in both the units. The initial PLS models built did not perform well (results not shown in the interest of conciseness), and encourages the use of Dynamic PLS models when handling dynamic data. The Subspace estimation method for unit-2 showed exceptional results and outperformed the DPLS1 model estimation for most of the outputs by a good margin. It can be attributed to the frequent sampling of variables in Unit-2, the inherent nature of Subspace models to capture transients, and the efficient Kalman based algorithm used for estimation.

Remark 20. *An ARX model built with I/O data with a chosen number of lags for inputs and outputs can be transformed into an equivalent Subspace model with a state-space structure. However, such a model is quite different from a model built with Subspace Identification based techniques, and most of the time will have a different set of parameters. Unlike SIMs, ARX models do not have the fundamental notion*

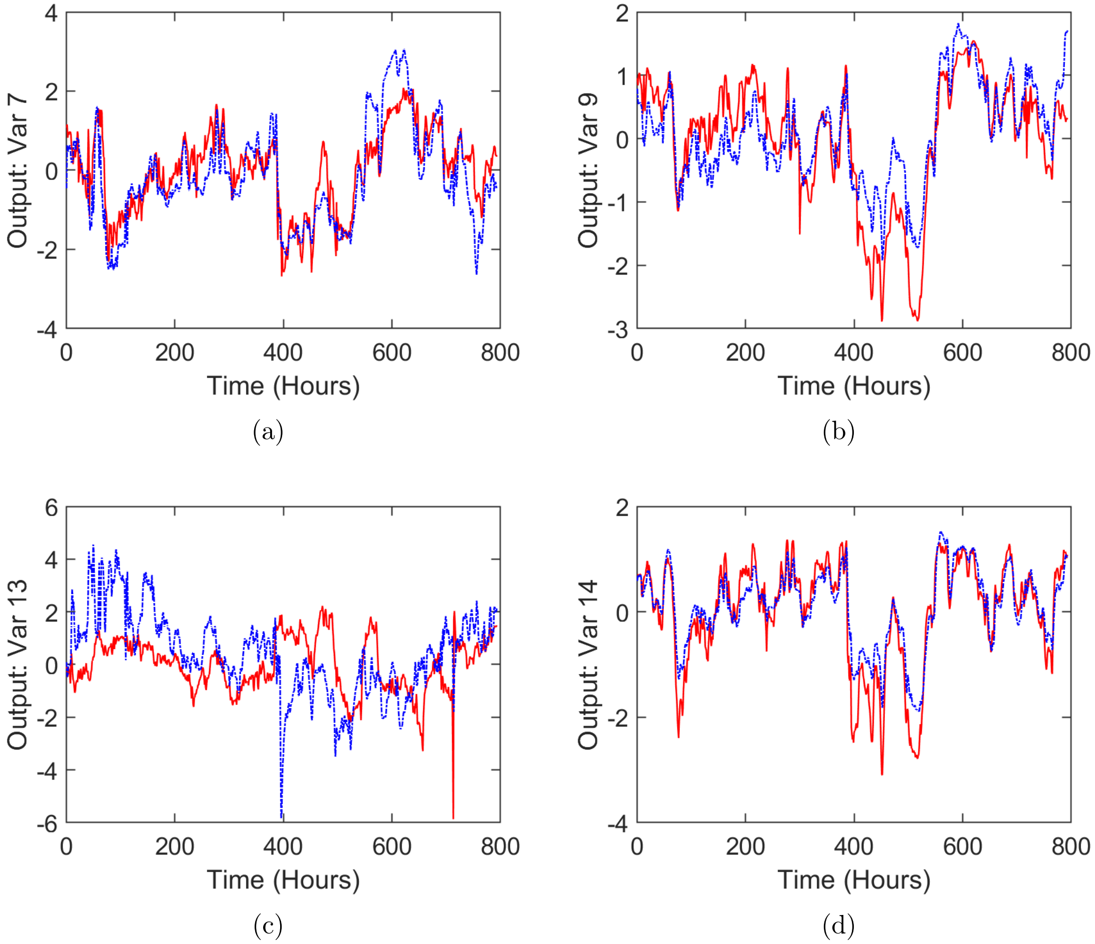
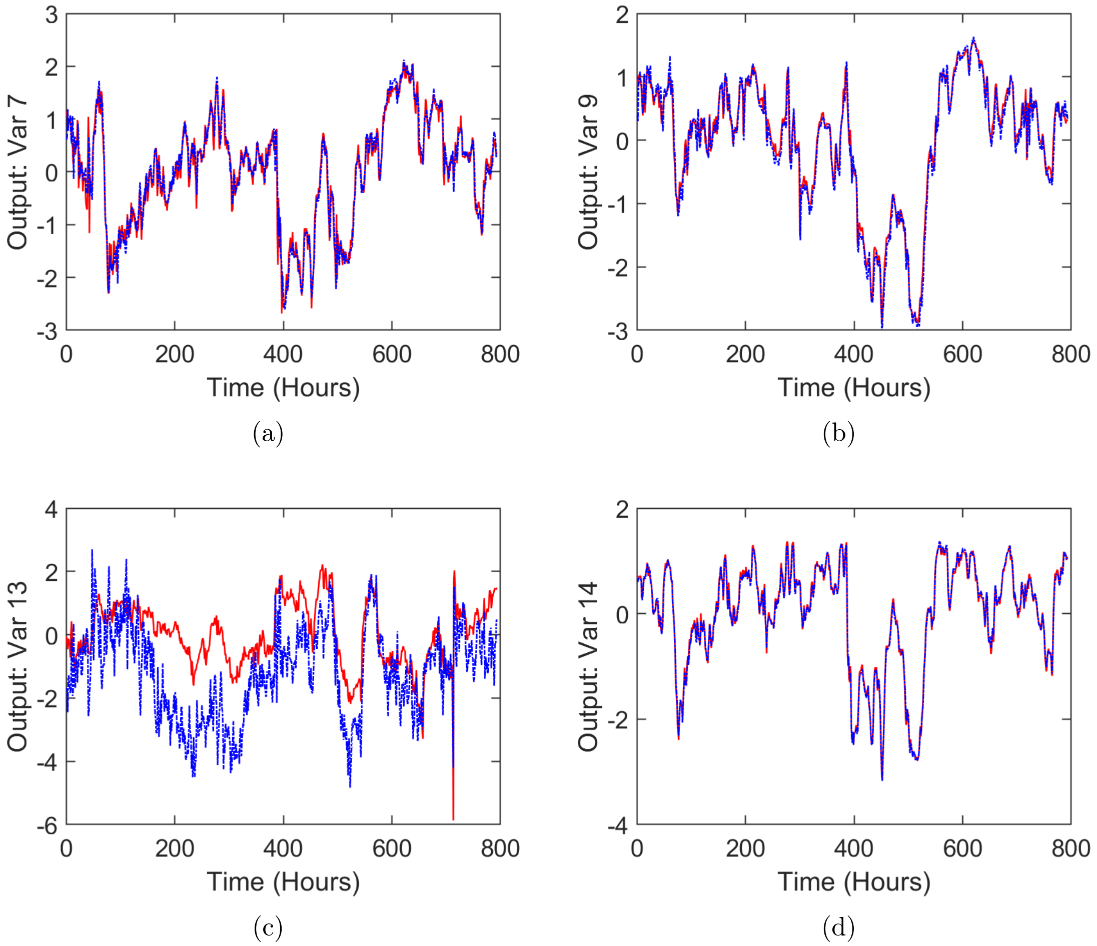


Figure 5.8: The validation results for 4 outputs of Unit-2 are presented in figures (a), (b), (c) and (d), respectively. The Process/observed values are given by the Red − line and the model prediction of DPLS1 by Blue −. line

of states, and depend on an appropriate guess/estimation of the number of lags to be used. The model parameters are then identified from data using regression techniques. DPLS models also have a structure similar to ARX models which involve a predetermined guess of lags, however, the parameters are identified in a different way. In essence, although ARX and DPLS can be converted to have a state-space structure through appropriate transformations, they differ fundamentally from SIMs in the way the parameters are identified.

Remark 21. *One of the important factors in the efficient and smooth operation of a*



(a) (b) (c) (d)

Figure 5.9: The validation results for 4 outputs of Unit-2 are presented in figures (a), (b), (c) and (d), respectively. The Process/observed values are given by the Red − line and the model prediction of Subspace based estimation by Blue −. line

plant in any production unit can be attributed to proper monitoring of process variables for early detection of faults [19, 6, 14]. Preventive maintenance based on such analysis leads to equipment safety, averts accidents, and minimizes major losses in production targets. Both DPLS [9] and Subspace methods [2] have demonstrated excellent ability in detection and diagnosis of faults, and, as such, can be implemented in a straightforward manner in hydroprocessing units as well. Studies involving these models for their potential use in a monitoring framework for such units will be explored in the future.

	Output	DPLS1	Subspace estimation
Fractionation region	Var 1	1.75446	1.26901
	Var 2	1.02073	1.08716
	Var 3	1.33204	1.39715
	Var 4	1.91197	2.20649
	Var 5	1.42332	1.73676
	Var 6	6.67260	7.48992
	Var 7	14.0420	4.56650
	Var 8	28.74670	8.11911
	Var 9	20.32090	5.46378
Reaction region	Var 10	15.6023	10.80576
	Var 11	1.18183	1.44064
	Var 12	0.06769	0.05318
	Var 13	1.38402	1.27661
	Var 14	3.36404	0.52568
	Var 15	3.09445	1.13913
	Var 16	4.66547	2.68285
	Var 17	4.24697	1.56998
	Var 18	1.80654	0.79695

Table 5.3: RMSEP values of different outputs of Unit-2

Remark 22. *The statistical modeling techniques adopted in this work also find potential uses in many control and optimization oriented applications. The state-space structure of Subspace methods are excellent for model based control schemes like Model Predictive Control (MPC), and have been widely used in a variety of problems. Latent variable MPC's also have shown their potential capability in many batch and continuous processes. The calculated input moves generated by inverting such models in an MPC scheme can be used to serve several control objectives in a hydroprocessing unit, and is a subject of future study. It is important to note that one can as well try non-linear modeling approaches like ANN's for such purposes. However, the recurrent issue of overfitting limits usage of these models for control strategies, and motivates* *the use of the presented linear modeling strategies, where these problems can be readily addressed.*

Remark 23. *It should be emphasized that the methodologies opted in this study have presented results which are a significant improvement over the current modeling techniques used in the industry where these units are operated. The main intent and contribution of this study was not to necessarily propose new modeling strategies in the hydroprocessing section, but to suggest to practitioners the usefulness and potential application of these models. The linear models used in this work are simple, readily implementable, and provide convenient statistical measures for visualization, data exploration, analysing irregularities, among various other uses. They are, thus, attractive modeling schemes for a variety of purposes in hydroprocessing units, and a valuable tool for practitioners.*

5.4 Conclusions

In this study, data-driven approaches for modeling two different hydroprocessing units of an industrial refinery is studied. Real historical data from plant operation is put to use to create tractable models which can estimate important variables with good accuracy. The usefulness of linear models like DPLS and Subspace based estimation algorithms for such units is shown. DPLS is considered as a good choice for the unit when measurement frequency is low, while the Subspace based estimation method is ideal for the unit where process variables are recorded at a faster rate. In the future, the utility of these linear models along with other non-linear schemes in an optimization framework for offline and online optimization purposes will be explored.

Acknowledgments

Financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC) through the Engage Grant, and the McMaster Advanced Control Consortium (MACC) is gratefully acknowledged.

Bibliography

[1] (2017). *Diesel Hydrotreating Process*, chapter 3, pages 23–49. John Wiley Sons, Ltd.

[2] Alcala, C. F., Dunia, R., and Qin, S. J. (2012). Monitoring of dynamic processes with subspace identification and principal component analysis. *IFAC Proceedings Volumes*, 45(20):684–689. 8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes.

[3] Ancheyta, J., Sánchez, S., and Rodríguez, M. A. (2005). Kinetic modeling of hydrocracking of heavy oil fractions: A review. *Catalysis Today*, 109(1):76–92. Hydroprocessing of Heavy Oil Fractions.

[4] Arce-Medina, E. and Paz-Paredes, J. I. (2009). Artificial neural network modeling techniques applied to the hydrodesulfurization process. *Mathematical and Computer Modelling*, 49(1):207–214.

[5] Bezergianni, S. and Kalogianni, A. (2008). Application of principal component analysis for monitoring and disturbance detection of a hydrotreating process. *Industrial & Engineering Chemistry Research*, 47(18):6972–6982.

[6] Bhadriraju, B., Kwon, J. S.-I., and Khan, F. (2021). Risk-based fault prediction of

chemical processes using operable adaptive sparse identification of systems (oasis). *Computers Chemical Engineering*, 152:107378.

[7] Brunton, S. L., Proctor, J. L., and Kutz, J. N. (2016). Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937.

[8] Chen, G. and McAvoy, T. J. (1998). Predictive on-line monitoring of continuous processes. *Journal of Process Control*, 8(5):409–420. ADCHEM '97 IFAC Symposium: Advanced Control of Chemical Processes.

[9] Chen, J. and Liu, K.-C. (2002). On-line batch process monitoring using dynamic pca and dynamic pls models. *Chemical Engineering Science*, 57(1):63–75.

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

[11] Dong, Y. and Qin, S. J. (2018). Regression on dynamic pls structures for supervised learning of dynamic data. *Journal of Process Control*, 68:64–72.

[12] Elkamel, A., Al-Ajmi, A., and Fahim, M. (1999). Modeling the hydrocracking process using artificial neural networks. *Petroleum Science and Technology*, 17(9-10):931–954.

[13] Froment, G. F., Depauw, G. A., and Vanrysselberghe, V. (1994). Kinetic modeling and reactor simulation in hydrodesulfurization of oil fractions. *Industrial & Engineering Chemistry Research*, 33(12):2975–2988.

[14] Galagedarage Don, M. and Khan, F. (2019). Process fault prognosis using hidden markov model–bayesian networks hybrid model. *Industrial & Engineering Chemistry Research*, 58(27):12041–12053.

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

[16] Jarullah, A. T., Mujtaba, I. M., and Wood, A. S. (2011). Kinetic parameter estimation and simulation of trickle-bed reactor for hydrodesulfurization of crude oil. *Chemical Engineering Science*, 66(5):859–871.

[17] Jiménez, F., Kafarov, V., and Nuñez, M. (2007). Modeling of industrial reactor for hydrotreating of vacuum gas oils: Simultaneous hydrodesulfurization, hydrodenitrogenation and hydrodearomatization reactions. *Chemical Engineering Journal*, 134(1):200–208. Proceedings of the XVII International Conference on Chemical Reactors CHEMREACTOR-17 and Post-Symposium "Catalytic Processing of Renewable Sources: Fuel, Energy, Chemicals".

[18] Kaheman, K., Kutz, J. N., and Brunton, S. L. (2020). Sindy-pi: a robust algorithm for parallel implicit sparse identification of nonlinear dynamics. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 476(2242):20200279.

[19] Kourti, T., Nomikos, P., and MacGregor, J. F. (1995). Analysis, monitoring and fault diagnosis of batch processes using multiblock and multiway pls. *Journal of Process Control*, 5(4):277–284. IFAC Symposium: Advanced Control of Chemical Processes.

[20] Ku, W., Storer, R. H., and Georgakis, C. (1995). Disturbance detection and isolation by dynamic principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 30(1):179–196. InCINC '94 Selected papers from the First International Chemometrics Internet Conference.

[21] Kumar, H. and Froment, G. F. (2007). Mechanistic kinetic modeling of the hydrocracking of complex feedstocks, such as vacuum gas oils. *Industrial & Engineering Chemistry Research*, 46(18):5881–5897.

[22] Lakshminarayanan, S., Shah, S. L., and Nandakumar, K. (1997). Modeling

and control of multivariable processes: Dynamic pls approach. *AIChE Journal*, 43(9):2307–2322.

[23] Leffler, W. L. (2008). *Petroleum Refining in Nontechnical Language (4th Edition)*. Pennwell Corp, USA.

[24] Li, G., Liu, B., Qin, S. J., and Zhou, D. (2011). Quality relevant data-driven modeling and monitoring of multivariate dynamic processes: The dynamic t-pls approach. *Trans. Neur. Netw.*, 22(12):2262–2271.

[25] Moonen M, Demoor B, V. L. V. J. (1989). Online and off-line identication of linear state-space models international journal of control. *International Journal of Control*, pages 49:219–232.

[26] Narasingam, A. and Kwon, J. S.-I. (2020). Application of koopman operator for model-based control of fracture propagation and proppant transport in hydraulic fracturing operation. *Journal of Process Control*, 91:25 – 36.

[27] 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.

[28] Nomikos, P. and MacGregor, J. F. (1995). Multi-way partial least squares in monitoring batch processes. *Chemometrics and Intelligent Laboratory Systems*, 30(1):97 – 108. InCINC '94 Selected papers from the First International Chemometrics Internet Conference.

[29] Overschee, P. V. and Moor, B. D. (1992). Two subspace algorithms for the identification of combined deterministic-stochastic systems. In *[1992] Proceedings of the 31st IEEE Conference on Decision and Control*, pages 511–516 vol.1.

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

[31] Sharmin, R., Sundararaj, U., Shah, S., Vande Griend, L., and Sun, Y.-J. (2006). Inferential sensors for estimation of polymer quality parameters: Industrial application of a pls-based soft sensor for a ldpe plant. *Chemical Engineering Science*, 61(19):6372–6384.

[32] Shokri, S., Sadeghi, M. T., Marvast, M. A., and Narasimhan, S. (2015). Improvement of the prediction performance of a soft sensor model based on support vector regression for production of ultra-low sulfur diesel. *Petroleum Science*, 12:177–188.

[33] Si, F., Romero, C. E., Yao, Z., Xu, Z., Morey, R. L., and Liebowitz, B. N. (2009). Inferential sensor for on-line monitoring of ammonium bisulfate formation temperature in coal-fired power plants. *Fuel Processing Technology*, 90(1):56–66.

[34] Sildir, H., Arkun, Y., Cakal, B., Gokce, D., and Kuzu, E. (2012). A dynamic non-isothermal model for a hydrocracking reactor: Model development by the method of continuous lumping and application to an industrial unit. *Journal of Process Control*, 22(10):1956–1965.

[35] Simon, D. (2006). *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley-Interscience, USA.

[36] Song, W., Mahalec, V., Long, J., Yang, M., and Qian, F. (2020). Modeling the hydrocracking process with deep neural networks. *Industrial & Engineering Chemistry Research*, 59(7):3077–3090.

[37] Verhaegen, M. and DeWilde, P. (1992). Subspace model identification part 1. the output-error state-space model identification class of algorithms. *International Journal of Control*, 56(5):1187–1210.

Chapter 6

Conclusions and recommendations

The main focus of research in my thesis has been modeling and application of different hybrid and data-driven methodologies. This chapter summarizes the main contributions and key findings of this thesis and proposes direction for future research.

6.1 Conclusions

In Chapter 2, a step by step strategy of building and validating a parallel hybrid modeling framework is presented. A subspace identification based residual model was chosen as the data-driven component which predicts the residuals and tries to correct the biases present in the mechanistic model. The parameters of the subspace model was identified using residual data from different historical batches in a meaningful way. A systematic way of identifying the model order was proposed using a cross validation technique. The framework also provides an easier alternative of updating the model by re-identifying the data-driven model with recent data. This job is much easier in contrast to updating the parameters of the mechanistic model. The efficacy of the proposed approach was demonstrated through simulation studies by predicting the dynamic evolution of three important variables for a batch polymerization PMMA process and compared against the predictions of a mechanistic model and a purely data-based subspace model.

In Chapter 3, the problem of implementing a MPC scheme embedding the parallel hybrid methodology was considered for a batch crystallization problem. A subspace based residual model in conjunction with the mechanistic model was first built and tested to establish the efficacy of the hybrid framework. The non-linear mechanistic component in this framework renders the MPC optimization as a non-convex problem. A central idea in this paper was to maintain the linearity of the MPC to make it tractable for real time computations and in that direction, a linear equivalent of the non-linear parallel model was first built, tested, and then implemented in the

MPC scheme. The goal of the control scheme was to reduce the volume of fines generated due to nucleation during the crystallization process. The hybrid model based MPC was able to not only reduce the volume of fines better for higher quality targets compared to an only data (subspace identification) model based MPC but also limited the spread of the final quality to ensure tighter product specifications.

In Chapter 4, a novel way of augmenting the X block of PLS models with information from mechanistic model to improve the predictive ability of PLS models was presented. Historical data from multiple batches were introduced in the mechanistic model to generate trajectories of unmeasured variables for each of those batches. This additional information was added to the X-block of PLS models to build models with better estimation capability. Measuring the crystal size distribution (CSD) in a crystallization process is a time and cost intensive process that is very often done offline. A motivating example of estimating the final crystal size distribution for a seeded batch crystallization example was considered. The hybrid or augmented PLS model was used as a soft/inferential sensor to estimate the quality or final CSD of the process to obviate the need for experimental testing, and its efficacy was demonstrated by comparing it with a traditional historical data-based PLS model.

In Chapter 5, the problem of building and validating linear dynamic models like DPLS and subspace identification to estimate important variables for two different hydroprocessing units was presented. A data-based approach to modeling was considered in the absence of readily available data from a first principles simulator (thereby limiting the avenues for building hybrid models). Real industrial data was used in building and validating these models. The subspace based identification algorithm was made more general to handle missing data in both inputs and outputs, a common occurrence in industrial data-sets. Simulation results show the efficacy of both the modeling strategies, and the presented results were a substantial improvement over the current methodologies used in the industry, where the real units are operated.

6.2 Future work

In this section, directions for future work are outlined for each of the chapters.

- The proof of concept studies in chapters 2 and 3 demonstrated the usefulness of the parallel hybrid methodology in prediction and control. It will be worth exploring/demonstrating the usefulness of such frameworks using real data or in pilot/industrial studies for a variety of processes. The most effective parallel hybrid model developed was the one where the non-linear mechanistic model was working in synergy with the subspace based data-driven model. Given its ability to predict dynamics so well, it is highly desirable to use a non-linear MPC for best control performance. With the rapid innovations in the field of optimization, it is worth to invest time and energy towards building a tractable strategy suitable for real-time computations for MPC embedding the non-linear hybrid model to compute optimal/near optimal solutions. In another direction, to adapt the model to process drifts, an algorithm to trigger re-identification for the hybrid model can be formulated. The parameters of the data-driven component of the architecture can be recursively updated using recent data to compensate for the bias in the predictions.

- The augmented/hybrid PLS methodology was shown to have better estimation ability compared to an only historical data based PLS. While the role of such a methodology as a static soft-sensor has been explored, its use as a dynamic model for k-step ahead prediction, and subsequent utilization in a MPC scheme is yet to be demonstrated. Missing data algorithms for PLS can be employed in that case which make use of previous correlation structure in the data to fill up the missing values for dynamic prediction. The crystallization case study can very well be extended to predict the moments of the crystal size distribution at each time step for a certain prediction horizon. This information can be exploited in a predictive control strategy to achieve quality objectives like on-spec crystals, reducing the volume of fines during the process etc.

- In chapter 5, historical data from actual operation of hydroprocessing units were used to build data-based models. The mechanistic simulator currently used in that concerned industry for making predictions and estimations often fails to converge, and that precluded the implementation of a hybrid model for this case study. A reduced order/surrogate model for such a high fidelity simulator would be a great way to make the knowledge useful. This will also enable real-time implementation of such complex models. Once that is accomplished, traditional grey box strategies can be employed to make those models even more efficient. The linear dynamic tools used in the study did an impressive job of estimating important variables in the unit, and in the future these models will be used in monitoring, optimization and predictive control strategy based studies for a variety of systems.