The Right Tools for the Job: Design Choices of Parallel First Principle and Data-Driven Hybrid Modelling for Prediction and Control of Batch and Fed-Batch Reactors

Alexander McKay

Friday August 19^{th} 2022

Abstract

This thesis focuses on the creation of new parallel hybrid model designs for prediction and control in batch and fed-batch reactors within Model Predictive Control (MPC) frameworks. In the hybrid model, the first principle (FP) explains the dynamics and the residual Subspace Identification (SID) model explains the error between the FP and the process. Modifications to the structure of the hybrid model are motivated by limitations of MPC frameworks. MPCs need accurate models to explain the system dynamics to make informed control decisions, and mechanistic models can be difficult to implement due to challenges of solving the optimization problem in real time. Two tools are demonstrated to help solve these problems. The first tool, Residual First Principle 0 Hybrid (RFP0H) model, helps to deal with the intractability of a mechanistic model in a hybrid modelling framework. The input for the FP model is kept constant and the SID predicts the error between the first principle and the process. Allowing for the desired output to be subtracted by the predicted FP to create a desired error value. Thus, MPC control only needs to be solved using the linear SID model in a linear or quadratic framework. Making a potentially intractable problem, tractable in MPC. This is demonstrated using a simulated fed-batch crystallization process. The second tool, Scaling Factor First Principle 0 Hybrid (SFFP0H) model, modifies the hybrid model structure to multiple the sub-models' outputs together. The SID data driven model predicts a factor to scale the FP output for the process prediction. The results demonstrate that the SFFP0H model has increased predictive ability and has smaller variability in control compared to the RFP0H model. Helping to solve the problem of needing accurate models within an MPC formulation. This is demonstrated by using a laboratory scale batch polymerization process.

Dedication

Dedicated to my previous self who was very stressed, we're all going to make it, and to my family, friends, and relationships who, over the course of my Master, reminded me that we are all that matter.

Acknowledgements

The first thanks goes to Dr. Mhaskar for all his help and advice through out the course of my Master. My education has advanced tremulously over the last two years, and I am grateful for the time and space to grow not just as an engineer but as a well rounded person. To Dr. Swartz for teaching me the phrase "Can't see the forest for the trees" and everything that he does for the MACC. To everyone that helped publish my first accredited paper, Lu Zhu, Debanjan Ghosh, and Dr. Li Xi. To all the people in the MACC for providing me with knowledge and community support during my time in grad school. To the Reject Stream for giving me a way to exercise and to relieve stress this summer. A shout out to the final member of my committee, Dr. Jacob Nease. And a final thanks to all the people, places, and things that I have gotten to experience in my eight years at McMaster University.

I am lucky to have met the people that I have met and the community that we all create.

Also, to the higher ups at McMaster University for teaching me about societal dynamics.

Contents

List of Figures

Chapter 1

Introduction

1.1 Perspective

Can't see the forest for the trees - English Idiom

Point of View: A particular attitude or way of seeing a matter

To be a researcher is to be busy wondering through the trees trying to understand an aspect of the forest better. To see the forest requires understanding what all other people are doing in the forests, the phenomena of the forest, and how the forest evolves over time.

1.1.1 Abstraction

Abstraction: The process of forming a concept to identify common features among a group of individuals and ignoring unique/outlying aspects [37].

To abstract is look at the conditions of reality and understand conditions, patterns, and relationships that exist between unique elements .

1.1.2 Reality

Reality is the aggregate of all the interactions and actions of mass and energy that exist in the universe [80].

What actually goes on between things.

1.1.3 Observation

The active acquisition of knowledge from direct sources. This occurs through sensory devices either biological or technological [14].

1.1.4 Framework

A web of connected concepts that together provide a comprehensive understanding of phenomena [62].

Question: How do humans understand the world?

Answer: People abstract and internalize from their observations of reality and put them together to posit new ideas.

1.2 System

A set of axion/items/things/people working together as part of a network to achieve a goal, or to create an output, or a comprehensive whole [50].

A framework of elements constructed within reality to achieve a goal or a directive or to understand the interaction of elements.

1.2.1 System Engineering

Using Engineering Principles for the creation, modification, and maintenance of systems [5].

1.3 Society

The aggregation of people together into groups that are organized by larger social structures and systems [19].

A system of people working together within their environments to achieve common goals held collectively consciously or unconsciously as a collective.

1.3.1 Agrarian Society

A society constructed around the maintenance and cultivating of arable land for the production and consumption of food [11].

1.3.2 Industrialized Society

A society that uses technology and machinery for the mass production of goods and services [43].

1.3.3 Industrial Revolution

Industrialization is the process of an economy going from an agrarian one to an industrialized one. There exist four distinct phases of industrialization that has occurred and are occurring on earth [43].

Industry 1.0

Industry 1.0 was the first stage of industrialization that occurred from around 1760 to 1840 predominately in Great Britain, mainland Europe, and the United States. This saw the rise of hand production techniques to machine ones, new chemical and iron production methods, and increased steam and water power. Many people working in agriculture in the countryside went to work in factories as labourers [43].

Industry 2.0

Industry 2.0 was the second stage of industrialization that occurred around 1870 to 1914. This stage was characterized by the mass production, standardization of manufacturing facilities, and the increased production of energies to power these systems. This stage of industrialization brought industry to many new areas of the world, and saw a great enough production of resources that industry could affect and change the lives of large swaths of the world [43].

Industry 3.0

Industry 3.0 otherwise known as the Digital Revolution was the third stage of industrialization that occurred from around 1950 to 2000. This saw the advent of the computer and their ascension to a seminal role in society and industrialized functions. Increases in computer technology allowed for increased data acquisition and information to improve systems, thanks to the computation efficiency of computers. Digitization allows for an acceleration of productive gains and news ways of improving manufacturing systems [23].

Industry 4.0

Industry 4.0 is the ongoing fourth stage of industrialization and is meant to represent the rapid transition of our society and industry in the 21st century to one that is deeply interconnected with the use of smart automation. Advances in industry come from synergizing different domains of knowledge which include artificial intelligence, advanced robotics, and gene editing [58].

1.4 Science and Engineering

How do people make sense of a world of noise and how do people force their will on their environments?

1.4.1 Science

Science is a complex and multifaceted approach to understand the natural phenomena that occur in the universe. Scientific understanding relies on testing of ideas (hypotheses) to create observations that help determine the validity of the idea. Science is a pathway to understand and a way to learn about the natural work, the systems within it, and the evolution of the universe [59].

The foundations of science is found in the ancient works through ancient philosophy, astronomy, and medicine [54]. These fields of knowledge shaped Greek natural philosophy that thought after explanation of events of the physical world by natural phenomena [54]. These ideas were held and contained in the Easter Roman Empire and the Muslim world until the advent of the Renaissance and the Scientific Revolution in Western Europe. This brought out renewed interest in the ancient world of thought, and created the conditions for the emergence of mathematics, physics, astronomy, and biology [54]. Science brought out an evidence based rational for understanding the world, and its affects shape humanity to this day, as humans use the foundations of previous knowledge to refine ideas and frameworks. People see that much further due to what has come before.

1.4.2 Engineering

Engineering uses scientific principles to design and build things. At the heart of engineering are problems that needs to be solved, and an engineer who needs to use their domain of knowledge to solve it. Engineers take the ideas and foundations of mathematics and sciences and use them in practical applied ways for desirable applications [52].

The foundations of engineering comes from civil and military engineering, where devices were needed to be created for the functioning of society and the mechanism of war between and within societies [52]. As the knowledge of science and engineering has increased over the centuries, along with the complexity of modern society, engineering has splintered into many different disciplines of knowledge. Now in modern society, it is hard to find an application or device that has not been shaped by scientific and engineering principles.

1.5 Chemical Engineering

Chemical Engineering is the ability to create, manipulate, and maintain chemical processing systems by using scientific principles to understand, explain, and control the movements of mass and energy [26]. Processing plants are a cornerstone of modern industry and society with oil and gas representing approximately 50% of all energy consumed globally in a year [48]. Some important processes in chemical engineering include, crystallization, food processing, polymer manufacturing, energy production, and

waste water [26]. The advances and luxuries of modern society would not be fully realized without the advances that Chemical Engineers have contributed.

1.5.1 Foundations of Chemical Engineering

Modern chemical engineering found its insemination during the second half of the nineteenth century. Originally most chemical processes were produced in modest craft operations, but due to increase demand from industrialization, safety concerns from the public, and competition from various companies created the incentives for greater efficiencies within processes. Large scaled chemical manufacturing operations occurred and these facilities hired more industrial chemists and chemical technologies. People who had dual domain knowledge of both chemistry and manufacturing [4].

These advances in industry brought about engineering as a profession. George E Davis considered by some as the father of chemical engineering first published Handbook of Chemical Engineering in 1901 [18]. In it, Davis talks in length of various systems and processes found within chemical plants. Much of it is descriptions of the mechanics of different process, reactions, and operations in chemical engineering. Interestingly, the first section of the book goes into details about the increased use of coal for the use as an energy source in Britain in the nineteenth century. Showing that at the heart of chemical engineering is the principle to understand how energy and mass are fundamental to processes.

More advances were made from the observations of earlier engineers, include from Arthur D Little. In 1915, Little proposed the concept of unit operations for chemical operations where chemical processes are divided into their component parts that create physical changes in chemical systems [41]. This is a fundamental component of chemical engineering today, and it would be difficult to discuss the subject without talk of heat transfers, reactor, separators, distillation columns, mixers, and more. Additionally, there was a division of chemical reactions into different classifications including polymerization, esterification, nitrations, etc. These advanced allowed for a more segmented view of chemical engineering, allowing for chemicals engineers to specialized into more specific fields of study.

Since these advancements of chemical engineering, there has been increased study into better understanding the fundamental operations of basic laws of mass transfer, heat transfer, and fluid flow [77]. Often the literature of many papers is to understand the mechanics of unit operations and to create mechanistic models of how these mechanics operate [6, 79, 25]. Digital computers allowed for labourious calculations to be done rapidly and for collecting mass amounts of data, allowing for these trends to accelerate in the industry and even finding new models using techniques like latent variables and Machine Learning [23].

1.5.2 Batch Reactors

Batch reactors are a unit operation found within chemical processes. Batch reactors work in distinct stages of operation, a transient state [67]. This is unlike many reactor operations liked Continuous Stirred Tank Reactors or Packed Bed Reactors which are attempted to be run in a condition of steady state. The steps of a batch reactor are three fold. Firstly, the reactor is fed with the necessary components needed for the reaction. Secondly, the reactor will undergo necessary temperature and pressure changes needed for the reaction to occur. Often the temperature of the reactor is manipulated throughout the process to ensure desirable products at the end of the process. Once the reaction has gone through to completion, the product will be emptied by the reactor for further processing and refinement elsewhere in the process plant [86].

Semi-batch Reactors

Similar to batches, semi-batch/fed-batch are reactors which operate with the same procedures of a batch reactor, but are open systems in which fed is added to the system through out the course of the reaction. Said fed often contains a component of the reaction necessary for the process to go to completion [86]. The fed along with removal or adding of heat to the system are inputs into the reactor system.

1.5.3 Control in Chemical Processes

Process Control uses industrial control systems to achieve a level of consistency, economy, and safety that would not be possible by a human alone. This is done in a control hierarchy that helps to control process functions from the smallest unit operation to the planning and scheduling of the entire operation [70]. The bed rock of most process control operations is the three factor Proportional Integral Derivative (PID) controller which was constructed on intelligent control [4]. A heuristic control which came from observation of human operators, specifically naval helmsmen. It was noticed how during control and operation of a ship, that helmsman would look at the proportionality of the error, the anticipation of a buildup or reduction of error, and the compensation for persistence error [4] to navigate towards the desired path. From initial mechanical limitations, PID controllers have been refined and digitized to make up over 80% of automated feedback control by the 1960's. While, PID can provide quality control of small unit operations, they do have setbacks including difficulty in operating Multi Input Multi Output (MIMO) systems and difficulty in dealing with dead time [46]. To make up for these limitations a more elegant solution would be necessary.

In the 1960', advanced control was any control algorithm that deviated from classical PID control. Examples include minimum variance, smith prediction, Generalized minimized variance, and pole placement [7]. With good tuning these control mechanisms could provide excellent control, but had limited

use in industry. It took several modifications of GMV to truly change the control industry. In 1978, Richalet et al proposed a control technique known as Model Predictive Heuristic Control [76]. The technique uses a dynamics plant model to predict the effects of future process output and uses a cost function to determine input values. This technique had many positives over PID including better performance, handling of constraints, non-minimum phase processes, robust control, and straightforward applicability to MIMO systems, and this technique eventually morphed into Model Predictive Control (MPC).

The popularity of MPCs have exploded in industry and academia since it's insemination. The ability to use dynamic plant models has extreme versatility due in part to the diversity of models that can be constructed of processes. The ability of an MPC to perform its job satisfactorily is highly dependent on the accuracy of the model embedded within the MPC. Different modelling domains have their own positive and negatives in terms of MPC implementation.

Mechanistic Models

Mechanistic models or First Principle Models are models constructed with an understanding of the underlying physics of a system [65]. In chemical processes, these are usually constructed by mass and energy balances which attempt to explain the movement and changes of the energy and mass overtime within a system. Typically, a mechanistic model will take the structure of a System of Ordinary Differential Equations (ODE). Equations 1.1 and 1.2 show the typical structure of a mechanistic model. Where x represents the state of the system, u represents the inputs to the system, and y represents the outputs of the system. $f(x, u)$ represents the system of equations that explain the rate of change of states, and $g(x, u)$ represents the system of equations that explain the outputs of the system.

$$\dot{x} = f(x, u) \tag{1.1}$$

$$y = g(x, u) \tag{1.2}$$

A mechanistic model is an excellent candidate for the MPC model if the model is well constructed. As mechanistic models are mathematically approximations of systems based on physical and observable concepts. Thus with enough information and expertise of the physics of a system, a mechanistic model will closely approximate that physical reality [78, 10]. If a mechanistic model has excellent prediction control, it should be used for MPC, and has been used in a number of MPC applications in industry. While powerful, there are a number of limitations to mechanistic models. To fully understanding the dynamics and physics of a system can be a challenging task that is not trivial for large and complex systems [2, 74]. Often creation of mechanistic models requires simplifying assumptions to be able to fully

conceptualize the physics. Additionally, the parameters values need to be approximated which requires data and experimentation to determine accurately, things that are both time consuming and costly. Finally, due to the non-linear structure of mechanistic models, tractability issues, and uncertainty, it can be difficult to formulate and to compute optimization problems in real time [2, 74]. Meaning that MPC with a mechanistic model of a process will either provide sub-optimal input suggestions or simpler controllers options such as PID (with their own problems) have to be used to control the process. Due to all these difficulties of mechanistic models making simpler model structures is attractive.

Data Driven Models

Data driven models are models constructed with a predetermine structure between the input and output data. Data from the system is used to perform algorithms and regressions using the model structure to determine model parameters [55]. The simplest data drive models are linear, and include Partial Least Squares Regression (PLS) and Subspace Identification (SID) [28, 33, 28, 75]. These linear model structures mean that it is trivial to solve and resolve for the parameters of these models. These type of models have been used to great affect in MPC control for a number of systems. Although the linear structure of these models can limited the impact their usefulness.

Neural Networks (NN) are another Data Driven modelling technique. They consist of layers of nodes that summed inputs together and use linear or non-linear activation functions to simulate the dynamics of a system. Neural Networks take advantage of a computer's processing power to compute large networks of nodes to eventually determine one that approximates the system dynamics [42]. Neural Networks are an extremely popular modelling technique and have also been used in a number of MPC applications [1, 57, 15]. There are limitations with the technique though as NNs have large number of parameters which means over-fitting of the model for data of limited amount or of poor quality.

Hybrid Models

Hybrid models are a modelling structure where multiple models from different modelling domains are combined together to provide a more informative model [84]. There are two modelling structures common in hybrid modelling, parallel and series. In the series technique the output of one sub-model is an input to the other sub-model which predicts the system outputs. In the parallel technique, the outputs of both sub-models are combined together for a more informed prediction of the process output. Hybrid models have been growing in popularity due to the different domains of knowledge that can be used between the different sub-models, including data driven, mechanistic, and heuristic [84, 81, 47, 73]. This combining of information helps realize the full information known about process systems, but provides both the positives and negatives of either sub model. Overall, hybrid models are a popular technique for MPC

due to these reasons, and they have been successfully implemented in a number of systems.

1.6 Tool

An object meant to improve a person's ability to modify components of systems. Often meant for improvement, creation, or maintenance of a system [72].

1.7 Putting it All Together

Since the insemination of humanity, people have had to deal with the material reality that surrounds them. People are both observers and participants of the places that they find themselves. To deal with the conditions that people find themselves, they need to provide frameworks from their observations and interactions to create an understanding of them. Once an understanding of the systems occurs after time and discussion, people can start to contend with the limitations and problems that has arisen in their environments and their social structures. Once people have both an understanding of systems and the problems that they create, they can start to process how changes to these systems can occur; how to get around the limitations. Often these changes to systems are most easily manipulated by tools that extend a person's ability to modify the world around them. These occurred in the earliest stages of human society, during hunter gather days. Using tools like flint for the creation of fire and poles for extra reach. The uses of tools allowed humans to create more complex social structures and increased the complexity of human systems [72]. This continued and snowballed and continues on to this day. Human society is much more complex now, then during pre-agricutural society, but humans still need to understand and interrupt their surroundings. To do that people still create frameworks to understand reality and how it can be manipulated. Their is still plenty of problems today and people continue to look for solutions to them.

Over the millennia, new tools have been created to solve for problems and modify people's surroundings. The scientific method and engineering principles have given rise to new ways to interpret, understand, and interact with reality. Now people have empirical ways of interpreting and testing their realities, and people also have ways of more reliably creating things that work within reality. This has allowed for the advent of new technologies that are capable of changing not just human systems but environmental systems. Steam and water engines are tools for the manipulation of energy to power process at a rate that an individual labourer is incapable of achieving [20]. The advent of these tool during the start of the industrial revolution helped to solve problems of the limitation of human labour capacity and the increase demand of goods.

The creation of a manufacturing society created their own problem including problems with safety,

operability, and reliability. These were new mechanical systems that were just being realized, and there was a need for new engineers who understood these processes for the maintenance and expansion of them [18, 41, 77, 44]. This need for technical expertise included the knowledge of chemical and fluid systems. Steam and water engines required the manipulation of fluids. The need for more obscure/refined chemicals required the knowledge of how to process these materials on an industrial scale. The need for knowledge of the movement of mass and energy was more important than ever. All of which provided the conditions for chemical engineers to become a part of the engineering profession at the start of the twentieth century. Chemical engineers have helped created many chemical processes from the Haber-Bosch process to refinery plants [44].

Chemical processes are highly complex and interconnected systems that are almost always in flux [78, 10, 29]. People need to maintain these systems at all times to ensure that conditions are met for a safe and effective processes that produce on-spec products. The maintenance of chemical process can be time consuming due to the many unit operations that operate under various conditions and range of processing conditions, and this becomes even more strenuous, the more complex the system is [10]. The operation of processes were limited by the need of operators to manually control processes. The number of personal who understand and could control processes were insufficient for the demands of chemical processes [9]. With these problems came the need to automate the control of processes. The first tool for automation included binary switches that would turn on or off depending on operating conditions [4]. While binary switches help for certain processing and safety situations, they are insufficient to deal with the range of conditions that processes needed to operate under.

The second big advent for control of chemical processes comes from the PID controller which has been used extensively within the chemical engineering discipline since the 1920's [4]. The PID controller provides simple control of processes. By providing direct fed-back of a process output and comparing said value to a desired value, PID controllers determine values of inputs that could be automatically manipulated that would likely reduce the error between the values. This was done by the creation of gains that would understand the relation of error within the system. This changed allowed for many simple components of processes to be automated, reducing the need for operators to manually control the processes [4]. Although, PID controllers comes with their own shortcomings. PID controllers explain gains of processes but not the underlying mechanics of systems, making it difficult for PID to be scaled to entire processes due to interconnected relationships of variables, impossible for PIDs to realize system constraints, and it handles dead-time poorly [4, 76, 7]. For these problems to be overcome, a more elaborate control method would be necessary.

Since the advent of the PID controller, more elaborate control techniques have became sought after in the field [45]. The advent of the Model Predictive Control was a monumental shift in process and control

engineering and allowed for greater control and relational understanding of processes. The reason for this is that embedded within an MPC is a dynamical model that explains the relationship between inputs and outputs of a process [45, 4]. Similarly to PID, measurements of process outputs are fed into the controller along with an output set-point, where the controller will calculated input values to attempt to achieve the set-point output by manipulating an input value. By using dynamical models within an MPC, it is possible for the controller to have a more complete understanding of the relationships within the process. Meaning that MPC could be used in Multi-Input Multi-Output systems, could handle constraints, deal with dead-time, and provide control for more complicated control objectives [45]. Like PID, MPCs do have their problems. Namely the need for a dynamical model. PID controller gains required a couple of simple experiments to determine their gains for the controller to function [45, 31, 69]. MPC requires a dynamical model which is not always easy to obtain and may not be consistent with the process reality. Failure to obtain an accurate dynamical model of the process can cause irregularities and failure of the MPC to achieve its objectives [45, 31, 69]. Thus, it is imperative to determine a good dynamical model before use of an MPC.

There are two common model structures that are used within an MPC framework. Mechanistic models and Data Driven models, each with their own benefits and drawbacks. Mechanistic models consist of physical relationship that are based on the fundamental laws of nature [65, 78, 10]. Often in chemical processes, they consist of energy and mass balances that explain how these concepts are manipulated and change over time. The benefit of mechanistic models is that as the model is based on the physical reality of a system, a well constructed model will provide excellent results in achieving process output [17, 56]. The problem with mechanistic models is that they are both time and energy intensive to create and maintain. Plus due to non-linearity, intractability, and uncertainty, it can be difficult to effectively control in real-time [78, 10, 29, 17, 56]. The benefit of data driven models is their ease of creation and maintenance. As Data Driven Models have a pre-constructed model formula, all that needs to be done to create one is to have data, and the model is run through an optimization problem to formulate the parameters of the model [55]. The drawback of Data Driven model is that the model structure limits its ability to completely explain the dynamics of the system. Linear Data Driven models might not fully explain the non-linear dynamics of a system [38, 39, 13]. Non-linear Data Driven models like Artificial Neural Networks might explain noise instead of the dynamics due to too many parameters from too little or noisy data [60]. Both of which mean a sub-optimal operation of the MPC. To solve for these problems, new tools need to be incorporated into the MPC model.

Many engineers, technicians, and scientist are working on solutions to these problems that are ongoing in MPC. Ideas like Bayesian Statistics may ease the burden of mechanistic model maintenance and provide more accurate explanations of process dynamics over time by using statistical methods to change

the parameter values over time [83]. The over fitting problem within Artificial Neural Networks could be solved by using Physics Informed Neural Networks to ensure that the process explains the dynamics of the system and not the noise [8]. All of these concepts are potential tools to solve problems, and collectively, people will determine their true worth. But that is not the focus of this thesis as there is interest in a different modelling schema.

Hybrid Models are model structures that contain multiple model that combine together in some constructive method to generally have greater prediction ability than either model either sub-model is capable of on their own [84]. It takes advantage of multiple sources of information, often both mechanistic and data driven models [81, 47, 73]. These multiples models generally combine different insights of the world, varying ways to understand and conceptualize chemical systems. Mechanistic, from insights into the physics of a system and data driven taking advantage of data and algorithms to explain the variance between input and output. Parallel hybrid models are one form of hybrid model where the outputs of the two models are combined together to predict the output of a process [81, 47, 73]. One form of the parallel hybrid model takes the form of a mechanistic model that explains the overall non-linear dynamics of a process and a Data Driven Subspace Identification Model that explains the error between the first principle and process output. Thus, the SID can help explain variance in the process that the first principle cannot and vice versa. Such a system, makes model maintenance and creation simpler. As the mechanistic does not need to explain the dynamics perfectly, that can be compensated for by the SID model. Additionally, maintenance of the model is simpler as deviations in the process can be explain by re-optimizing the residual SID. A problem that still remains with this hybrid modelling framework is the real time optimization under MPC control. Due to challenges such as intractability, uncertainty, and non-linearity that occur with non-linear mechanistic models, it is still difficult to solve the optimization problem in real time to determine optimum input values. New tools for a hybrid parallel models would be necessary to overcome these draw-backs.

For the problem of of real time optimization of the parallel hybrid model combining SID and a first principle, Ghosh et al created their own tool for easier optimization within the MPC framework. This was done by creating an SID model of the first principle. Thus making the model completely linear in its predictive ability and easier for control applications [36]. Although, much of the non-linear process dynamics was lost due to this change of the model structure which hinders the predictive ability of the model.

People need tools to extend their ability to manipulate systems and surroundings, so as to solve problems. In this thesis, two new tools are created and discussed in detail in service of better Control within MPC frameworks. In Chapter 2, titled, A Novel Linear Hybrid Model Predictive Control Design: Application to a Fed Batch Crystallization Process, the Residual First Principle 0 Hybrid (RFP0H)

model is shown. A modification to the Residual Hybrid model is made where the first principle is fed a constant input through the process. This allows for the first principle to be solved once at the start of the process. This modification means that the desired process output can be subtracted by the first principle model to create a desired error output. As a desired error exists, the MPC optimization only needs to be concerned with the linear SID model meaning that control can be computed under a linear or quadratic formulation. This removes the problem of intractability and non-linearity of the mechanistic model during optimization. This makes optimization easier providing superior control to alternative linear models while maintaining the predictive ability of the non-linear model. This new tool is demonstrated on a simulated system of a fed-batch crystallization reactor. In Chapter 3, titled, A Multiplicative Structure of a Parallel Hybrid Model for Optimal Temperature Model Predictive Control in a Laboratory Scale Batch Reactor, the Scaling Factor First Principle 0 Hybrid (SFFP0H) model is shown. A modification of the output structure of the parallel hybrid model is made where the output of the two sub-models are multiplied together instead of being summed together. Meaning that instead of the SID model predicting the residual error of the process, it predicts a factor to scale the first principle prediction to predict the process output. This structure maintains the first principle model structure from the RFP0H model where the first principle is fed a consistent input profile. Allowing for the first principle to be solved once to allow for linear-control. This helps remove the problem of needing an accurate model for good control within an MPC framework. As the SFFP0H has superior predictive ability to the RFP0H model and reduced variation in control operations compared to the RFP0H model. Helping to indicate that multiplicative hybrid structure predicts error more reliably compared to an additive structure. The SFFP0H model is demonstrated using both a simulated and actual laboratory scaled polymerization batch reactor.

Chapter 2

A Novel Linear Hybrid Model Predictive Control Design: Application to a Fed Batch Crystallization Process

Abstract

This paper addresses the problem of enabling the use of complex first principles model information as part of a linear Model Predictive Control implementation for improved control. This is achieved by building a hybrid model that uses an approximate implementation of a first principle model and a Subspace Identification (SID) State Space model to explain the error (the residual) between the first principle implementation and the process outputs. The key idea is to utilize the first principles model with the initial conditions consistent with a particular batch, but using a constant value of the control action. Thus, even though the first principles model may be intractable from an optimization perspective, the approximate implementation allows the hybrid model to be linear (in the control input), while allowing the nonlinear dependence on the initial conditions to be captured. The proposed hybrid model based MPC is compared against a previous hybrid model with 2 SID models and a single SID model on a fed batch crystallization process. The paper demonstrates the improved performance achievable by the readily implementable proposed approach.

2.1 Introduction

The operation and control of processes is a challenging task due to several issues such as non-linearity, uncertainty, and constraints. One control approach designed to handle such challenges is Model Predictive Control (MPC) and is extensively used though out the process industry. MPCs use models that capture the dynamic relationship between manipulated inputs and process outputs to formulate and solve optimization problems to determine inputs values while achieving appropriate closed-loop performance. The accuracy of the model used within an MPC directly affects how well the controller is able to achieve its objectives. There exists many variations of MPCs including linear, non-linear, economic, robust etc [31]. These controllers cater to a wide range of objectives which typically include set-point tracking, regulatory problems or economic objectives like maximizing or minimizing quality metrics of a product, lower energy usage, increased reliability, and can be adequately adapted with regard to the problem at hand [69].

The model embedded within an MPC has to closely reflect the actual physical reality of the system in order to make accurate predictions. Choice of an appropriate model is, thus, imperative to ensure that the controller picks inputs that manipulate the system in a desirable way. The dynamics of many process systems are generally non-linear and are made up of complex interactions between different mechanisms. Mechanistic models can describe the dynamics of such systems relatively well, and where available, can be utilized in MPC formulations [78, 10, 29, 17, 56]. However, in many situations, figuring out the equations that govern the physical process is a highly arduous task, and often require simplifying assumptions. Optimization procedures to determine the parameters further involve a lot of complexity. Finally, the maintenance of such complex models are costly over time, and their real time implementation in control/optimization schemes impose significant challenges [2, 74] to the practitioners as well, and in many cases, the first principles models may be simply too complex to be implemented as part of an optimization algorithm.

Due to the challenges with development, maintenance and deployment of mechanistic dynamic models, there has been a significant push towards building simpler models, and purely data based model structures offer an attractive alternative. These data driven models use predetermined model structures to find correlations between input and output process data [55]. Partial Least Square Regressions (PLS) and Subspace Identification (SID) are two such linear data based modelling techniques which have shown great efficacy in capturing the dynamics of several continuous and batch processes [28, 27]. PLS finds a linear relationship between the input and output data using Singular Value Decomposition (SVD) or regression based techniques where summary variables are discovered to maximize the relationship between the input and output data [33, 75]. It is an inherently static model and modifications need to be introduced to deal with dynamics of a system, examples include dynamic PLS or time-index dynamic

models [22, 63]. SID is another technique that takes advantage of the numerically efficiency of the SVD algorithm to find a linear time invariant (LTI) state-space model using historical input output data. The algorithm first finds a realization of states, and then performs a linear regression step to determine the parameters of the LTI model [66]. These linear data driven models are considerably easier to build compared to the physics based mechanistic model, and can be easily implemented in real-time control strategies. There exists a significant amount of the literature where both PLS and SID have been used to model complex dynamics, and subsequently implemented into an MPC [38, 39, 13, 32, 68, 51] to achieve desired objectives. These models have proved useful in a wide variety of processes, but they remain potentially limited by the linear dynamics of the model structure.

There exists several non-linear data driven approaches, Artificial Neural Networks (ANN), being one of the most notable and frequently used technique. They assign non-linear functions to build the relationship between the inputs and outputs [42], and their dynamic adaptations have been used in a variety of MPC formulations [85, 87, 34]. Unfortunately, these models involve a large number of parameters and have a tendency of over fitting when data is limited and of low quality. These drawbacks along with their complex non-linear nature limit their utility within MPC formulations [60].

Hybrid techniques are another class of models which try to synergize mechanistic and data-driven approaches to obtain more accurate models. Typically, a hybrid model consists of one data driven model, e.g. PLS, SID, and ANN, and one mechanistic model, although it is possible to have two mechanistic models or two data driven models [84] in a specific combination. There are two main structures for a hybrid model: parallel and series, although, other ways of combining knowledge from different sources are possible. A series model consists of one model running prior to the second model in a series arrangement. The output of the first model is typically the parameters of the second model and is fed as inputs into the second model. A parallel model works by each base model running in tandem. The outputs of each separate model are then combined together to provide the predicted output of the whole architecture [81, 47, 73, 82].

The parallel hybrid model typically operates in a way where the mechanistic model tries to best explain the non-linearity in the process, and the data driven model corrects the error between the mechanistic model and the true process measurements [35]. The use of high fidelity mechanistic models in such schemes can be difficult to implement in real time monitoring and control strategies as they are hard to solve and require large computation times [61, 2]. Simpler yet still accurate models are, therefore, desirable as they can be readily embedded into linear or quadratic programming based MPC schemes to obtain faster solutions. This was recently accomplished by creating a hybrid model with two linear subspace based data models running in parallel [36]. While the approach proposed in [36] works sufficiently well, the MPC formulation does not necessarily utilize the first principles knowledge

available directly.

Motivated by the above considerations, a new hybrid modelling approach is proposed that enables the use of potentially intractable first principles model in a way that is tailored for an MPC implementation. The mechanistic part of the hybrid model uses constant input values, and is simulated using only the initial conditions. The SID based data driven model is trained on the error between the constant input first principle predictions and the true process outputs. This allows for the first principle model to be solved only once, and not as part of the predictive model inside the MPC, while still capturing the nonlinear dependence on the initial conditions. The proposed approach is demonstrated in a crystallization fed batch reactor where the control objective is to reach desired terminal qualities by manipulating the inlet fed concentration. The rest of the paper is organized as follows: Section 2.2.1 provides a brief overview of the crystallization fed batch process, Section 2.2.2 provides an overview of Subspace Identification, Section 2.2.3 goes over hybrid modelling and the recently proposed hybrid modelling approach, Section 2.2.4 reviews a recently proposed MPC formulation that uses a linear hybrid model. Section 2.3 describes the proposed modelling approach with Section 2.3.1 proposing the novel hybrid model that is the focus of this paper, Section 2.3.2 provides detail on model identification, Section 2.3.3 provides the details on model validation, and Section 2.3.4 presents the MPC formulation. Section 2.4 provides the methodology and results of the best models and MPC control with Section 2.4.1 describing the model implementation and Section 2.4.2 describing the MPC implementation. Finally, Section 2.5 gives a brief summary and conclusion to this paper.

2.2 Preliminaries

A fed batch crystallization process is considered as the simulated system and is briefly described in this section. This is followed by an introduction to Subspace Identification, and the state of the art in Hybrid Modelling and Model Predictive Control.

2.2.1 Simulated System: Crystallization

A fed batch crystallization reaction between a base and an injected acid solution is used as the simulated system. The stirred-tank reactor is initially filled with the base solution and the acid solution is fed into the liquid phase at a constant flow rate. For modelling and control purposes, the input considered is the inlet concentration of the acid. The base and acid react to form a salt species insoluble in the solvent. The salt precipitates from the solution through a nucleation and crystallization process. Additional base and acid units continue to react and the salt product can either grow the existing crystal particles or form new nuclei. No additional base is added to the reactor throughout the course of the reaction,

resulting in decreasing base concentration as the reaction proceeds. The fed batch reaction is considered to be complete when the base concentration goes below a certain threshold value, which can no longer sustain the supersaturation of the salt product. At that state, the reactor contains crystals of the final size distribution and excess acid in the solution phase. A visual depiction of this fed batch crystallization process is shown in Figure 2.1.

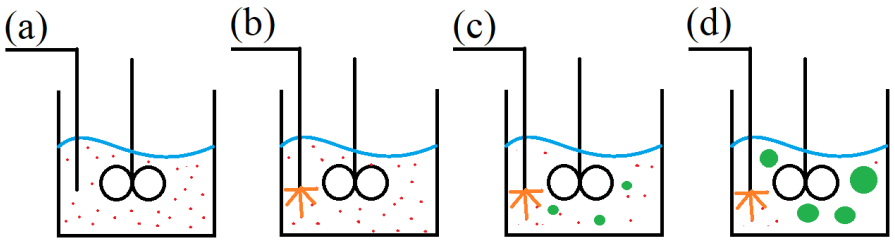

Figure 2.1: Steps of the crystallization process in a fed batch reaction. In (a), the reaction hasn't started, and the reactor contains the base solution, seen as the red dots. In (b), the acid solution is fed into the reactor, seen as the orange lines dispersing from the inlet stream. In (c), the base and acid have started to react together to form solid nuclei, seen as the green spheres in the solution. The amount of base in the solution has decreased. In (d), the reaction has stopped as there is negligible base in the reactor, and the nuclei have grown to their final size and number. Note that the particles and nuclei drawn in the figure are for illustrative purposes only, neither would be resolvable by the naked eye nor would they be the colours shown.

For the illustration of the proposed framework, a mechanistic model is used as the test bed (referred to as the process equations). The same set of equations with parametric errors is used as the first principles model. To describe the dynamics of the process, the method of moments is used [53]. Each of these moments has a physical significance and indicate a certain property associated with the crystals. The zeroth moment represents total crystal number per unit volume of the solution, the first moment represents the total crystal length per unit volume of the solution, the second moment represents the total crystal surface area per unit volume of the solution, and the third moment represents the total crystal volume per unit volume of the solution [49]. These properties are highly important to the quality of crystallization products, and reaching the desired moment values at the end of a batch indicates successful completion of that batch.

The time evolution of the moments are shown in Equations 2.1 and 2.2. The rate of change of the zeroth moment ($m_{0,\alpha}$) is determined by the nucleation rate (J_α) which is shown in Equation 2.1. The nucleation rate is described by Equation 2.3. The rate of change of successive moments ($m_{k,\alpha}\, \alpha = 1, 2, 3, 4$) is determined by the previous moments magnitude and the growth rate (G_α) which is shown in Equation 2.2. The growth rate is determined by Equation 2.4. $A_{f,\alpha}$ and $B_{f,\alpha}$ are parameters that vary depending on the properties of the reactor, where $f = 1, 2$ for A and $f = 1, 2, ..., 6$ for B.

$$\frac{\delta m_{0,\alpha}}{\delta t} = J_\alpha \tag{2.1}$$

$$\frac{\delta m_{k,\alpha}}{\delta t} = KG_p m_{k-1} \tag{2.2}$$

$$J_\alpha = \begin{cases} 0 & \text{if } S_\alpha < 1 \\ B_{1,\alpha}(S_\alpha - 1)^{7/3} exp(\frac{-B_{2,\alpha}}{ln^2 S_\alpha - 1}) + B_{3,\alpha}(S_\alpha - 1)^{B_{4,\alpha}} (\frac{M_\alpha}{M_{solu}})^{B_{5,\alpha}} \epsilon^{B_{6,\alpha}} & \text{if } S_\alpha \geq 1 \end{cases} \tag{2.3}$$

$$G_\alpha = \begin{cases} 0 & \text{if } S_\alpha < 1 \\ A_{1,\alpha}(S_\alpha - 1)^{6/5} exp(\frac{-A_{2,\alpha}}{S_\alpha - 1}) & \text{if } S_\alpha \geq 1 \end{cases} \tag{2.4}$$

Both the nucleation and growth rates are highly dependent on the variable concentration product ratio calculated in Equation 2.5. The concentration product ratio is the square root of the ratio of the product of the base and the acid concentration over a reaction specific parameter, the square of the polymorph solubility ($K_{sp,\alpha}$). If the concentration product ratio has a value below 1, there is no growth or nucleation and the reaction stops. Equation 2.6 shows the calculation of the mean specific output, ϵ is a variable in the nucleation rate, N_p is the power number of the motor in the reactor, d_{imp} is the diameter of the impeller, n_s is the stirring rate, and V_0 is the initial volume of the reactor. Additionally, Equations 2.7 and 2.8 show the mass balances of the acid (n_{acid}) and the base (n_{base}) within the reaction. C_{in} and Q_{in} are, respectively, the inlet concentration and inlet flow rate. ρ is the density of fluid in the reactor, $k_{v,\alpha}$ is a parameter of the crystal shape factor, and $M_{w,init}$ is the molecular weight of the acid. As time progresses, the acid and the base react to form nuclei decreasing their concentrations and the fed batch process stops once the concentration product ratio reaches a value less than one.

$$S_\alpha = (\frac{C_{mon} C_{init}}{K_{sp,\alpha}})^{1/2} \tag{2.5}$$

$$\epsilon = \frac{N_p d_{imp}^5 n_s^3}{V_0} \tag{2.6}$$

$$\frac{\delta n_{acid}}{\delta t} = Q_{in} C_{in} - \frac{\rho k_{v,\alpha}}{M_{w,init}} \frac{\delta m_{3,\alpha}}{\delta t} \tag{2.7}$$

$$\frac{\delta n_{base}}{\delta t} = -\frac{\rho k_{v,\alpha}}{M_{w,init}} \frac{\delta m_{3,\alpha}}{\delta t} \tag{2.8}$$

As mentioned earlier, Equations, 2.1-2.8, are used both as an alternative to the process, and as the first principles model, albeit with parameter differences, see Table 3.1 for the differences. Further, measurements from the process is assumed (and simulated) to be corrupted by measurement noise. The

Parameter	Process Value	FP Value	Percent Change	Unit
$A_{1,\alpha}$	4E-11	3.8E-11	-5%	1/s
$A_{2,\alpha}$	1.0	0.95	-5%	-
$B_{1,\alpha}$	1.46E12-	1.533E12	+5%	1/s
$B_{2,\alpha}$	67.3	70.665	+5%	-
$B_{3,\alpha}$	1E13	1.05E13	+5%	1/s
$B_{4,\alpha}$	0.8	0.84	+5%	-
$B_{5,\alpha}$	50	52.5	+5%	-
$B_{6,\alpha}$	0.1	0105	+5%	-
$k_{sp,\alpha}$	1.63E-5	1.711E-5	+5%	mol^2/m^6
N_p	1000	1000	0%	-
d_{imp}	2.0	2.0	0%	m
n_s	10.0	10.0	0%	1/s
ρ	1000	1000	0%	kg/m^3
$M_{w,init}$	142.04	142.04	0%	g/mol
$M_{w,mon}$	233.38	233.38	0%	g/mol
$K_{v,\alpha}$	0.5235988	0.54978	+5%	mol^2/kg^2

Table 2.1: Parameters values used to solve the ODE system of the method of moments in the fed batch crystallization reaction for both the process outputs and the first principle (FP) model, the percent difference between the two scenarios, and the units of the parameters.

standard deviation value of each process output is first found, and a random value between -5% and 5% of the standard deviation is added to the process output at every time step (see Figure 2.2 illustrating (a) the mechanistic model and (b) the process). Euler's method is used to solve the ODEs with an integration interval of 0.01 minutes. However, measurements are recorded only every 0.5 minutes. The initial conditions of all the moments are zero.

Data Base Generation

75 batches are simulated and used to train and validate all the models presented in this paper. To reflect process variation, initial conditions are varied from batch to batch. Thus, batches have different initial base fractions, acid fractions, initial masses of the reactor, and initial inlet acid fraction concentrations into the reactor. Table 2.2 provides the bounds on these initial conditions, and the values for each batch are determined randomly.

There is a series of step changes for the input, inlet concentration, throughout the batch. Every 12.5 minutes a step change is performed on the inlet concentration, and this change is randomly determined between 0.8 and 1.2 times the magnitude of the previous concentration value. Measurements are assumed to be available every 0.5 minutes. The magnitude of inlet concentration is always kept between the bounds of 0.04285 and $0.4 mol/L$ which is equivalent to an inlet fraction between 0.01 and 0.1.

The length of each batch varies as the initial conditions and input values differ from batch to batch, which influences the rate at which reactants are consumed. The end of a batch is determined when the

Variable	Low End	High End	Units
Initial Base Fraction	0.2	0.3	-
Initial Acid Fraction	0.01	0.1	-
Initial Mass	100	250	kg
Inlet Flow Rate	1E-3	2.5E-3	m^3/s
Inlet Acid Fraction	0.01	0.1	-

Table 2.2: Range of the initial conditions in the fed batch reaction used to generate the database.

moments values plateau.

To generate the training data, a matrix of input data (U) with dimensions ($m \times N$), the parameter values, the System of Ordinary Equations, and the initial output conditions are given to the ODE solver. The variable m represents the number of inputs and N is the total number of time steps of the batch. The value of m is 1 as the only input is the inlet salt fraction into the reactor. The outputs are the values of the moments for the entire batch (Y) with dimensions ($l \times N$). The variable l represents the number of outputs, and the value of l is 5 as the outputs are the zeroth to fourth moments.

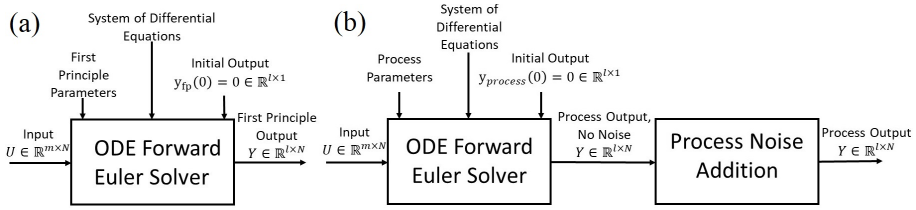

Figure 2.2: Black Box of how the batch data is generated for both the First Principle and the Process. (a) shows the first principle data generation, and (b) shows the process data generation.

2.2.2 Subspace Identification

Subspace Identification uses Single Value Decomposition (SVD), QR Decomposition, or other similar techniques to identify a linear time invariant state space model of the form provided in Equations 2.9 and 2.10. The first step in identification involves determining a state sequence trajectory using SVD by finding the row space intersection of a Hankel matrix containing input-output data. These theoretical states of the batches can then be used with the inputs and outputs to determine the matrices A, B, C, and D through an appropriate linear regression step [66]. The resultant model takes the form:

$$x_{sid}[k+1] = Ax_{sid}[k] + Bu_{sid}[k] \tag{2.9}$$

$$y_{sid}[k] = Cx_{sid}[k] + Du_{sid}[k] \tag{2.10}$$

where $x_{sid}[k]$ is an $(n \times 1)$ vector with n representing the number of states in the system, k indicates the

sampling instance, and $u_{sid}[k]$ and $y_{sid}[k]$ represent the input and output vectors, respectively, and have the dimensions of $(m \times 1)$ and $(l \times 1)$. m and l represent the number of inputs and outputs, respectively. These states are referred to as theoretical states and do not necessarily directly correlate with the states of the mechanistic model. A model structure for a process consisting of a single SID model is referred to as the Process 1-SID Model (P1SID Model) for the remainder of the paper.

2.2.3 Hybrid Modelling

A hybrid modelling strategy uses a combination of models to achieve superior predictions of outputs. Most hybrid models typically use both a mechanistic model and a data driven model (e.g. PLS, ANN, SID) It is, however, possible to use two mechanistic models or two data driven models in combination as well [84]. There are two popular hybrid modelling structures: parallel and series.

In this paper, we focus on the parallel structure. A parallel hybrid model uses two sub-models running in tandem. The outputs of the two sub-models (Y_1 and Y_2) are combined together to produce the predicted output of the process (Y_3). The input to the two sub-models (U_1 and U_2) can be the same set or a different set of data depending on the individual models [81, 47, 73, 82]. Figure 2.3 shows the generic structure of a parallel hybrid model.

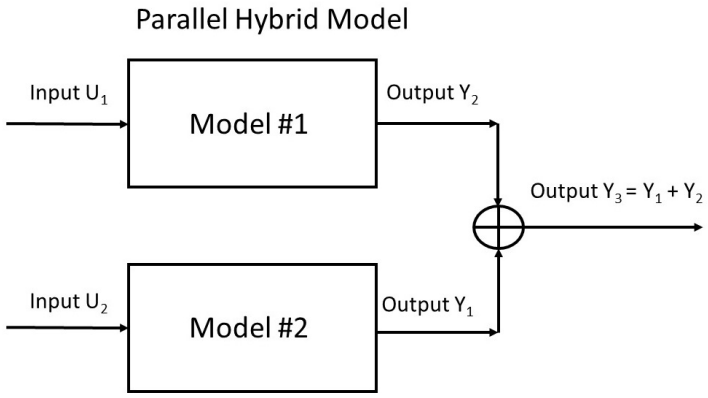

Figure 2.3: Black Box diagram of a typical Parallel Hybrid Model which is showing the model structure, inputs, and outputs.

A parallel hybrid modelling approach was recently proposed which integrates a mechanistic model with a subspace identification model. The first principle explains the overall trend of the evolving trajectories, and the SID model is built to predict the error that exists between the process and the first principle output trajectories. These two models together form the parallel hybrid architecture [35]. This model structure is referred to as the Residual and First Principle Hybrid Model (RFPH Model) for the rest of the paper.

The first principle calculations are shown in the Equations 2.11 and 2.12 which include the states

(x_{fp}) of the process, the inputs (u), and the outputs (y_{fp}). Equations 2.13 and 2.14 show the equations for the error model ($e_{sid}[k]$). The input values are the same for both the first principle and the SID. A, B, C, and D are the matrices determined from SID. Equation 2.15 shows that the output of the first principle and the SID error prediction are combined together to calculate the predicted output at any k^{th} time point ($y_{pred}[k]$).

$$\dot{x}_{fp} = f(x_{fp}, u) \tag{2.11}$$

$$y_{fp} = g(x_{fp}, u) \tag{2.12}$$

$$x_{sid}[k+1] = Ax_{sid}[k] + Bu[k] \tag{2.13}$$

$$e_{sid}[k] = Cx_{sid}[k] + Du[k] \tag{2.14}$$

$$y_{pred}[k] = y_{fp}[k] + e_{sid}[k] \tag{2.15}$$

2.2.4 Linear Hybrid Model for Model Predictive Control

While the first principles model for the simulated system is relatively simple, there are often cases where high fidelity and complex mechanistic models, such as process simulators are available. The problem with these first principles models is that it can be exceedingly difficult to implement these models into control formulations because these models require complex optimization techniques to solve. To address this challenge, a linear hybrid predictive model was formulated in reference [36] and used in an MPC framework. The hybrid model consists of two SID models running in parallel, an SID model that predicts the first principle outputs and an SID model that predicts the error between the process outputs and the first principles SID outputs. The two outputs are then combined to predict the process outputs. The model structure for this linear hybrid model is shown in Figure 2.4.

Equations 2.16 and 2.17 show the first principle prediction SID, Equations 2.18 and 2.19 show the error prediction SID, and 2.20 shows the combining of the two models for process prediction. $x_{fp,sid}$ is the states of the SID first principle, $x_{e,sid}$ is the states of the error SID, $u[k]$ is the vector of input values at time k, $y_{fp,sid}$ is the predicted output of the first principle SID output, e_{sid} is the predicted error between the first principle and the process output, and y_{pred} is the predicted process output. This model structure is refereed to as the Residual and First Principle 2-SID Hybrid Model (RFP2SIDH Model)

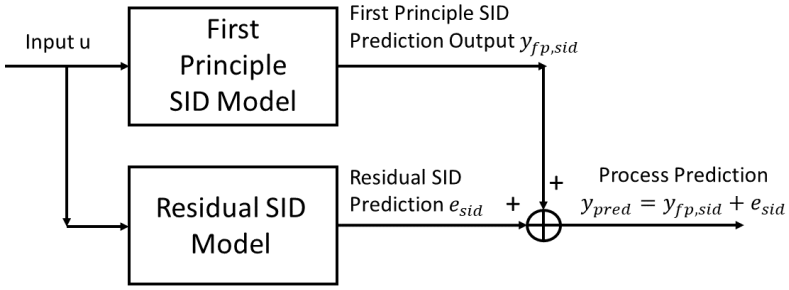

Figure 2.4: Model structure of the RFP2SIDH model.

[36].

$$x_{fp,sid}[k+1] = A_{fp}x_{fp,sid}[k] + B_{fp}u[k] \tag{2.16}$$

$$y_{fp,sid}[k] = C_{fp}x_{fp,sid}[k] + D_{fp}u[k] \tag{2.17}$$

$$x_{e,sid}[k+1] = A_e x_{e,sid}[k] + B_e u[k] \tag{2.18}$$

$$e_{sid}[k] = C_e x_{e,sid}[k] + D_e u[k] \tag{2.19}$$

$$y_{pred}[k] = y_{fp,sid}[k] + e_{sid}[k] \tag{2.20}$$

A schematic of the close loop implementation using the RFP2SIDH Model is shown in Figure 2.5. While the approach in [36] provides a way to embed the hybrid model within the MPC, it possibly compromises on the predictive capability of the first principles model. A hybrid modelling and control structure that could make better use of the first principles model, without increasing the complexity of the resultant optimization problem in the MPC is therefore valuable, and presented in the next section.

2.3 Proposed Modelling Approach

In the present work, a linear hybrid model approach containing a first principle model and a SID model is proposed and implemented into an MPC framework. This is compared to a non-linear hybrid model, a hybrid model with 2 SIDs, and an SID model. The proposed modelling approach is first presented followed by model identification and validation. The MPC control strategy for the proposed model is

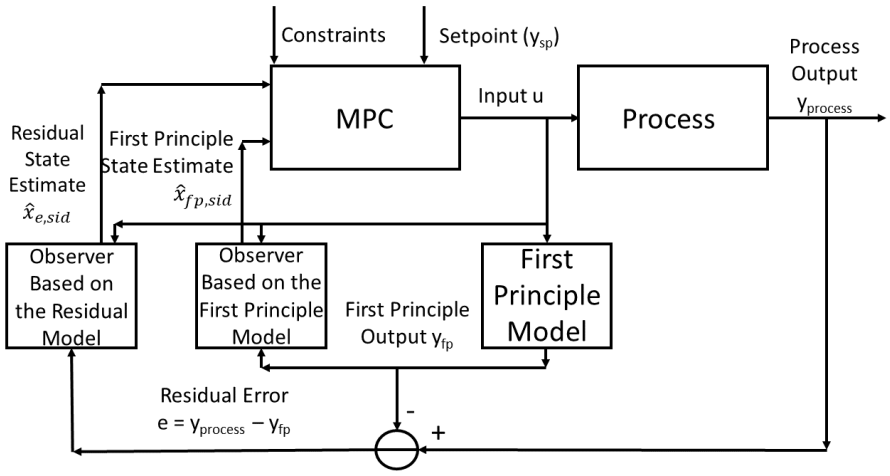

Figure 2.5: A closed loop schematic of the RFP2SIDH Model used in reference [36].

presented in the final subsection.

2.3.1 Hybrid modelling Strategy

A novel change to the standard RFPH Model is proposed where the inputs for the first principle model are held constant at their initial values throughout the extent of the batch. The SID model used to predict the error is modified to predict the error between the first principle outputs (generated using constant inputs) and the true process outputs. The modified first principle equations are shown in Equations 2.21 and 2.22 with a constant input variable (u_0). x_{fp0} represents the first principle states and y_{fp0} is the first principle output. Equations 2.23 and 2.24 represent the equations of the error SID model, notice the variable input ($u[k]$). x_{sid0} is utilized in computing the residual SID states with e_{sid0} being the residual error prediction. The final equation, Equation 2.25, shows the output of the two models being added together to produce the predicted process output (y_{pred}). The idea behind this approach is to enable retaining the quadratic nature of the MPC problem presented in Section 2.3.4, while still incorporating the nonlinear dependence of the initial conditions on the process evolution. This model structure is referred to as the Residual and First Principle 0 Hybrid Model (RFP0H Model). The first principle model with constant inputs on its own is refereed to as the First Principle 0 model (FP0).

$$\dot{x}_{fp0} = f(x_{fp0}, u_0) \tag{2.21}$$

$$y_{fp0} = g(x_{fp0}, u_0) \tag{2.22}$$

$$x_{sid0}[k+1] = Ax_{sid0}[k] + Bu[k] \tag{2.23}$$

$$e_{sid0}[k] = Cx_{sid0}[k] + Du[k] \tag{2.24}$$

$$y_{pred}[k] = y_{fp0}[k] + e_{sid0}[k] \tag{2.25}$$

In any MPC problem, the most accurate model that can be reliably computed and solved in real time is the model that should be used for control. If a good first principles model exists, and is tractable for optimization, utilizing it in an optimization formulation is expected to yield the best performance. The present manuscript's contribution is not to address such scenarios. The manuscript addresses those scenarios specifically where a practitioner puts in significant effort to develop a detailed first principles model that works very well for prediction. The complexity of the resultant model is such that it is simply intractable to include in an optimization framework. This is especially true for models developed in simulators like Aspen Dynamics that can be next to impossible to implement into an optimization equation due to the structure of the software architecture. If one were to completely abandon this detailed model, one could resort to the other extreme which is a linear data driven model. Such a model can be reliably and easily implemented into real time optimization, because the optimization would be linear or quadratic in nature which are guaranteed to converge, and have short computation times.A hybrid model with a residual SID and a first principle with constant inputs presents a trade off between the two modelling extremes. In the proposed technique, the first principle can be solved once at the start to capture the effect of the initial conditions on the process evolution. The effect of the inputs on deviations of this trajectory are what the data driven model computes, resulting in a tractable optimization problem and the approach still contains information about the non-linear dynamics of the system via the first principle model.

2.3.2 Model Identification

The general procedure for building the SID component of the RFP0H Model is described below. To identify the residual model, subspace identification is utilized. The approach is tailored to accommodate for multiple batches by using a pseudo Hankel Matrix [12]; unlike a typical Hankel Matrix, a pseudo Hankel matrix contains data for multiple batches or data with time discontinuities. The first step of SID is to create Hankel matrices in the form of Equation 2.26 using the output data from the b^{th} batch. The dimensions of each matrix are $(il \times j)$. i and j are parameters that need to be determined before creating the matrix, and i needs to remain consistent between batches [66]. Note that the outputs in the Hankel matrix for the RFP0H model is the difference between the process output and the first principle model output with a constant input, because the output of the SID is the error between the first principle

output and the process output. The input is the variable input profile.

$$Y_{H1}^{b} = \begin{vmatrix} y^b[1] & y^b[2] & \dots & y^b[j] \\ y^b[2] & y^b[3] & \dots & y^b[j+1] \\ \vdots & \vdots & \ddots & \vdots \\ y^b[i] & y^b[i+1] & ... & y^b[i+j-1] \end{vmatrix} \tag{2.26}$$

$$Y_{H2}^{b} = \begin{vmatrix} y^b[1+i] & y^b[2+i] & \dots & y^b[j+i] \\ y^b[2+i] & y^b[3+i] & \dots & y^b[j+i+1] \\ \vdots & \vdots & \ddots & \vdots \\ y^b[2i] & y^b[2i+1] & ... & y^b[2i+j-1] \end{vmatrix} \tag{2.27}$$

i is chosen in such a way that it is greater than the order of the model [66]. Additionally, once i has been determined the value of j is determined as j is picked so the matrix contains all data points of the batch. Three other Hankel matrices are created, one where the indexing starts at $i+1$ instead of 1, which is shown in Equation 2.27, and two more matrices that have the same structures of Equations 2.26 and 2.27 but contain inputs instead (U_{H1}^{b} and U_{H2}^{b}). These input Hankel matrices have the dimensions of $(im \times j)$ [66].

Once all Hankel matrices have been created for the b_{th} batch, these are amalgamated into the pseudo Hankel matrix that contains data for all the batches. The formulation of these concatenated matrices for the outputs are shown in Equation 2.28 and 2.29. The inputs also have these amalgamated matrices in the same format and are referred to as U_{H1} and U_{H2} [66].

To factor in the variable lengths of each the batches, the number of columns of each batch, j, are allowed to be different. The number of columns of the concatenated matrices (Y_{H1}, Y_{H2}, U_{H1}, and U_{H2}) is given by M which is the sum of all the different j values for all the batches.

Y_{H1} and U_{H1} are concatenated vertically as are Y_{H2} and U_{H2}, amalgamating the inputs and outputs together in the form shown in Equations 2.30 and 2.31. There is one final concatenation of H_1 and H_2 that creates the final matrix H shown in Equation 2.32 [66].

$$Y_{H1} = \begin{bmatrix} Y_{H1}^{1} & Y_{H1}^{2} & \dots & Y_{H1}^{N_t} \end{bmatrix} \tag{2.28}$$

$$Y_{H2} = \begin{bmatrix} Y_{H2}^{1} & Y_{H2}^{2} & \dots & Y_{H2}^{N_t} \end{bmatrix} \tag{2.29}$$

$$H_1 = \begin{bmatrix} Y_{H1} \\ U_{H1} \end{bmatrix} \tag{2.30}$$

$$H_2 = \begin{bmatrix} Y_{H2} \\ U_{H2} \end{bmatrix} \tag{2.31}$$

$$H = \begin{bmatrix} H_1 \\ H_2 \end{bmatrix} \tag{2.32}$$

The identification stage involves finding the theoretical states of the training batches, and the parameter matrices A, B, C, and D. The theoretical states are found by performing Single Value Decomposition (SVD) on the H matrix of Equation 2.33. There are three resulting matrices from the SVD, namely, U, S, and V. The sub matrices within U are U_{11}, U_{12}, U_{21}, U_{22}, and have dimensions of $(mi+li)\times(2mi+n)$, $(mi+li)\times(2li-n)$, $(mi+li)\times(2mi+n)$ and $(mi+li)\times(2li-n)$, respectively. The sub matrix within S, S_{11}, has the dimensions $(2mi+n)\times(2mi+n)$. The transpose of U_{12}, U_{11}, and S_{11} are multiplied together and SVD is computed on the resulting matrix shown in Equation 2.34. The U_q sub matrix within the U_2 matrix has dimensions of $n\times(2li-n)$. The theoretical states of the SID model are then found by multiplying the transpose of U_q along with U_{12} and H_1 shown in Equation 2.35 [66].

$$H = USV^t = \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix} \begin{bmatrix} S_{11} & 0 \\ 0 & 0 \end{bmatrix} V^t \tag{2.33}$$

$$U_{12}^t U_{11} S_{11} = U_2 S_2 V_t = \begin{bmatrix} U_q & U_q^T \end{bmatrix} \begin{bmatrix} S_q & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_q^t \\ V_q^{t,T} \end{bmatrix} \tag{2.34}$$

$$\widehat{x_{theoretical}} = U_q^t U_{12} H_1 \tag{2.35}$$

Once the states trajectories of the SID are found, the parameter matrices of the model are determined. The indices between $(1:l)\times(1:M-1)$ contain all the output data for all the batches in Y_{H2}, and the indexes between $(1:m)\times(1:M-1)$ contain all the input data for all the batches in U_{H2}, Equations 2.36 and 2.37. With the states, inputs, and outputs data, the parameter matrix (Φ) can be determined using a matrix regression in the form shown in Equation 2.38. The matrices A,B,C and D have dimensions of $n\times n$, $n\times m$, $l\times n$, and $l\times m$, respectively [66]. Once the matrices parameters have been computed the identification of the model is complete.

$$Y_j = Y_{H2}[1:l, 1:M-1] \tag{2.36}$$

$$U_j = U_{H2}[1:m, 1:M-1] \tag{2.37}$$

$$\Phi * \begin{bmatrix} \widehat{x_2}[:, 1:M-1] \\ U_j \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} * \begin{bmatrix} \widehat{x_2}[:, 1:M-1] \\ U_j \end{bmatrix} = \begin{bmatrix} \widehat{x_2}[:, 2:M] \\ Y_j \end{bmatrix} \tag{2.38}$$

2.3.3 Model Validation

The model validation involves evaluating the predictive capability of the model on new batch data not used in the identification process. There are two phases of model validation for each new batch, namely, (initial) state estimation and prediction. As stated in Section 2.2.2, subspace identification uses theoretical states to explain the process. At the start of a new batch the states of the SID are not known as the states cannot be directly observed. To estimate the states, a state observer is utilized. A state observer uses the model and the measured output values to correct the state estimates at each time step, and any state estimation technique like Kalman filter, Luenberger observer, moving Horizon estimator, etc can be used for this purpose. In this work, a Luenberger observer is used to estimate the states [24].

The form of Luenberger observer is provided in Equation 2.39. L is the observer gain and the poles are ensured to be within the unit circle, values less than one. This is done using the place function in Matlab. The initial states estimates are taken to be zero. The Luenberger observer is allowed to run for a limited number of sampling instances, by which instant the output predictions have converged with the actual outputs which indicates the possible convergence of the state estimates. In particular, the Luenberger observer takes the following form:

$$\hat{x}[k+1] = A\hat{x}[k] + Bu[k] + L(\hat{y}[k] - y_{pred}[k]) \tag{2.39}$$

The Mean Absolute Scaled Error (MASE) metric is used to evaluate the efficacy of the model predictions on validation batches. The MASE calculation is shown in Equation 2.40 [30] below:

$$MASE = \frac{\sum_{t=1}^{t_p} \mid e_t \mid}{\frac{t_p}{t_p - 1} \sum_{t=2}^{t_p} \mid y_t - y_{t-1} \mid} \tag{2.40}$$

y_t is the t^{th} process output, e_t is the error between t^{th} prediction and process output, and t_p is the final time point of the model predictions. The MASE computation scales the error and allows for comparing outputs that have different magnitudes. Once the observer has run its course in estimating the states, the prediction phase begins. The model is fed the rest of the input information and uses it to generate the model predictions. These predictions are then compared with the true output values.

The initial state estimation and validation can be computed for multiple batches to see how the model performs for various initial conditions. The MASE is found for each batch, and then the average MASE value is found. The best models are deemed to be the ones with the smallest average MASE, indicating more accurate predictions throughout the batch, and on average, across multiple batches.

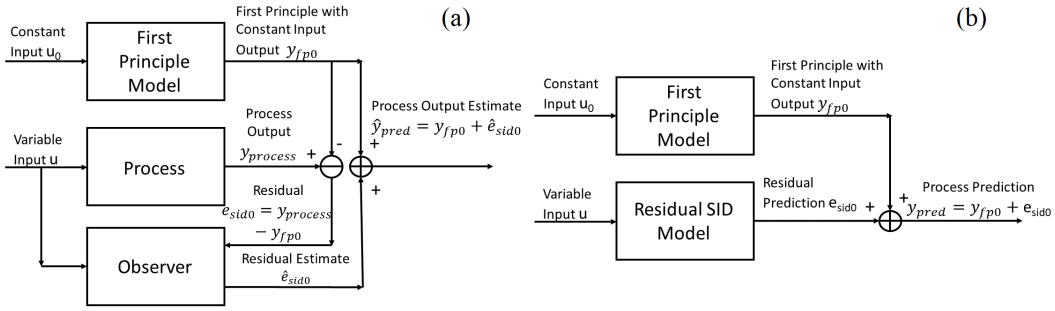

Figure 2.6: Two phases of the validation, (a) the observer and (b) prediction, for RFP0H

Figure 2.6 shows the two phases of model validation, with (a) showing the model working with the observer, and (b) showing the modelling working in the prediction phase when the model is fed the input values and predicts the outputs with no knowledge of the actual outputs.

Figures 2.4 and 2.6 (b) show the model structures of RFP2SIDH Model and the RFP0H Model, respectively. When comparing the two, the structural differences between them become clear. The RFP2SIDH Model has the same variable input data that is fed into the two sub-models. The RFP0H Model has two different input data matrices that are fed into each sub-model, a matrix of constant inputs for the first principle model and a matrix of variable inputs for the residual SID model; the two inputs have the same initial values until the variable input performs its first step change. The variable input matrices is the same across both model structures. The RFP2SIDH has an SID model to calculate the first principle while the RFP0H model uses a mechanistic model. In both model structures, the outputs of the two sub-models are then combined together to produce the process output prediction.

2.3.4 Model Predictive Controller Design

At a sampling instance k_c, the model predictive controller implementation takes the form:

$$
\begin{aligned}
& u[k], k = k_c, k_c+1, ..., k_f = arg(min\ J = (e_2^s - \tilde{e}_{sid0,2}[k_f])^2) \\
& u_{lb} \leq u[k] \leq u_{ub} && k = (k_c, k_c+1, ..., k_f) \\
& u[k-1] - u_{diff} \leq u[k] \leq u[k-1] + u_{diff} && when\ mod(k, v) = 0 \\
& u[k] = u[k-1] && when\ mod(k, v) \neq 0 \\
& \tilde{x}_{sid0}[k_c] = \hat{x}[k_c] \\
& \tilde{x}_{sid0}[k+1] = A\tilde{x}_{sid0}[k] + Bu[k] \\
& \tilde{e}_{sid0}[k+1] = C\tilde{x}_{sid0}[k+1] + Du[k+1] \\
& e_2^s = y_2^s - y_{fp0,2}[k_f]
\end{aligned}
\tag{2.41}
$$

where the various components of the MPC are explained below:

The objective function: The aim of the controller is to minimize the square difference between the output error set-point (e_2^s) and the predicted error from the SID at the end of the batch for the second moment ($\tilde{e}_{sid0,2}[k_f]$) where k_f denotes the final time index and the output error is the difference between the desired set-point of the second moment at the end of the batch (y_2^s) and the predicted second moment output from the first principle with constant input at the end of the batch ($y_{fp0,2}[k_f]$). The MPC is implemented in a receding horizon fashion, and the decision variables are all the input values between the current sampling instance of the batch (k_c) and the final time for the batch (k_f). The MPC implementations have been tested against their ability to achieve different set-points for different batches.

Constraints: All the inputs ($u[k]$) are bounded between a lower and an upper value (u_{lb} and u_{ub}) respectively for $k = (k_c, k_c + 1, ..., k_f)$. These values are the same input bounds used during batch generation and model creation. Additionally, for each step change the absolute difference between the new input ($u[k+1]$) and the previously implemented input ($u[k]$) cannot exceed the value of u_{diff}. The constraints also ensure that between every v sampling instances (when measurements are being collected but the input values is not changed), the input stays at the previous value. The constraints ensure that all the step changes have magnitudes that the model was trained and tested on, that such input changes are physically realizable, and that step changes occur at the same frequency as model training and testing.

Note that in this application, the initial conditions for the states of the first principle model are also known. The first principle 0 (FP0) model for that particular batch can therefore be integrated (with constant input) to determine the second moment ($y_{fp0,2}[k_f]$) ahead of solving the optimization. This in turn can be subtracted from the setpoint to yield e_2^s.

Thus, for the MPC, the set point value (e_2^s) represents a desired error value for the SID residual model to reach at the end of the batch, meaning that only the linear subspace model needs to be dealt with in the MPC. Due to the linear nature of the SID model, the resultant optimization problem remains a quadratic program that can readily be solved.

The schematic of the closed loop controller is shown in Figure 2.7. The state estimator is run throughout the extent of the batch. Once v iterations of the observer has occurred and the observer has calculated accurate estimates of the states using input and output data, the state estimate ($\hat{x}_{sid0}$), the set-point (e_2^s), and the constraints are fed into the MPC. The MPC calculates the input profile for the system, and the calculated input at the k_c iteration is applied to the systems for the next v sampling instances. During the v instances, the observer keeps updating the state estimates.

For the MPC to make accurate prediction it is important that step changes occur approximately as

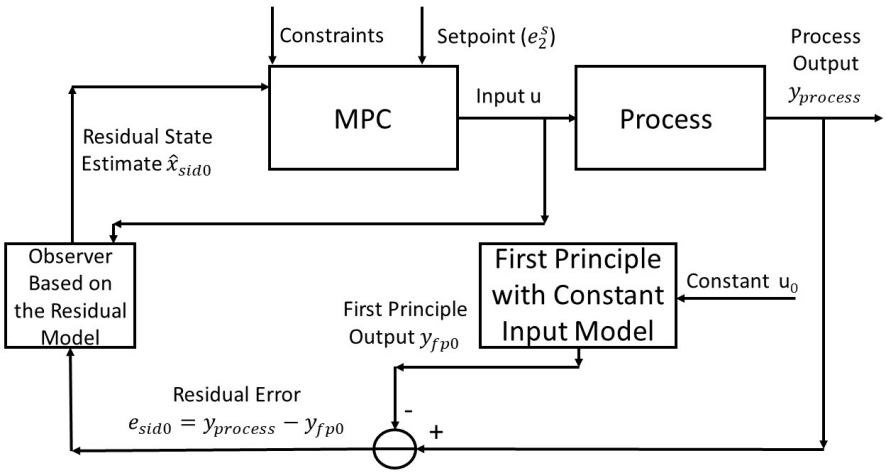

Figure 2.7: Closed loop schematic of the RFP0H model framework.

frequently in the controller as occurred in model training and validation. The sampling time of the model is every $\Delta T_m = 0.5$ minutes, and this is the rate of iterations in the SID model. In model training and validation, there is a hold time where step changes occurred every $w = 12.5$ minute. Thus, in the MPC design, every w minutes or every $v = 25$ sampling instances, the objective function is solved and chooses a new input value for the next v iterations. This step change occurs for every sampling instance k where the modulus of k/v is equal to 0. Thus the control action is held constant for v sampling instances.

It should also be noted that the FP0 model will vary for each individual batch as the initial conditions are consistent with the FP model, both the initial states and initial inputs of the process. Thus, for control applications, it is more complicated than simply adding a bias, but information is provided about the system dynamics when the desired error is determined.

Reformulating the optimization formulation

The objective function is to minimize the square difference between a predetermined error set-point and the predicted residual value of the second moment at the end of the batch. The equation of the objective function is shown in Equation 2.41. To solve this problem the objective function must be rearranged so that a solver can actively manipulate all the inputs from the current iteration of the batch until batch termination. To accomplish this, iteration is used on the state space equations to determine the predicted value at the end of batch that contains the parameter matrices, the current state ($x[k_c]$), and the current and future inputs for $k = (k_c, ..., k_f)$. A pattern emerges from the iteration and the general objective function is shown in Equation 2.42 below:

$$J = (e_2^s - X_1 - M_1 u_1 - M_2 u_2 - ... - M_r u_r)^2 \tag{2.42}$$

where M_1, M_2, etc are a summation and product of different combination of the SID parameter matrices, the general formulation is shown in Equation 2.43, u_1, u_2, etc are the input and decision variables of the objective function, and X_1 is a constant from the product of the parameter matrices and $x[k_c]$. Each input is held for v iterations of the SID starting at the current (k_c) iteration, and there are r step-changes to implement which is the floor of the difference of the final iteration and the current iteration divided by v, see Equation 2.44.

$$M_q = \sum_{k=(q-1)v+1}^{qv} CA^{kf-kc-k}B \tag{2.43}$$

$$r = floor(k_f - k_c/v) \tag{2.44}$$

Equation 2.42 can be expanded out and reconstructed into a quadratic matrix equation in the form shown in Equation 2.45, with the quadratic, linear, and scalar components of the equation shown separately below. These matrices along with the constraints can be inputted into a quadratic optimizer where the inputs are determined to minimize the value of the objective function well maintaining the constraints. Once the objective function has been solved, the first input change that the solver determines is implemented into the batch reaction. The rest of the input suggestions are discarded and not performed on the system.

$$\begin{aligned}
J =& u^T F u + G u + c \\
& u^T F u = \begin{bmatrix} u_{M_1}^T & u_{M_2}^T & ... & u_{M_z}^T \end{bmatrix} \begin{bmatrix} M_1^T M_1 & M_1^T M_2 & ... & M_1^T M_z \\ M_2^T M_1 & M_2^T M_2 & ... & M_2^T M_z \\ \vdots & \vdots & \ddots & \vdots \\ M_z^T M_1 & M_z^T M_2 & ... & M_z^T M_z \end{bmatrix} \begin{bmatrix} u_{M_1} \\ u_{M_2} \\ \vdots \\ u_{M_z} \end{bmatrix} \\
& G u = 2(X_1 - e_2^s) \begin{bmatrix} M_1 & M_2 & ... & M_z \end{bmatrix} \begin{bmatrix} u_{M_1} \\ u_{M_2} \\ \vdots \\ u_{M_z} \end{bmatrix} \\
& c = e_2^s - 2y^s X_1 + X_1^2
\end{aligned} \tag{2.45}$$

The performance of the MPC for the particular batch can then be determined by finding the absolute difference between the desired (y_2^s) set-point output and the actual output of the second moment at the end of the batch ($y_{process,2}[k_f]$), Equation 2.46.

$$\|y_2^s - y_{process,2}[k_f]\| \tag{2.46}$$

2.4 Application to the Fed Batch Crystallizer

This section does a comparison of the various modelling approaches using both validation metrics and closed-loop performance metrics. Recognizing the possible differences in the best selection of the parameters for each model structure (e.g., number of states, lags), in this section, 100 instances of models of each model structure, RFPH, RFP0H, RFP2SID, and P1SID, are trained and validated for prediction and control of the process outputs of a fed batch crystallization reaction. The instances differ in how a particular data set is split between training and validation, and different choices for number of states and rows of the Hankel matrix during SID training. The data sets, however, are kept consistent between each approach to be able to make a fair comparison. 30 batches of data are used to train each model and are validated using another 45 batches of data. The model with the smallest average MASE values for the RFP0H model, the RFP2SID model, and the P1SID model are chosen for the MPC implementation. The control performance of all the models types are then evaluated.

2.4.1 Model Creations and Comparison

45 batches of data with variable inputs and initial conditions are generated for the first principle and process using the methods discussed in Section 2.2.1. These 45 batches are meant to represent a database of historical batches and are divided into 9 groups of 5 with each group shown in Table 2.3.

Group Number	Batch Numbers
1	22, 38, 20, 19, 43
2	7, 34, 13, 26, 25
3	18, 3, 8, 16, 45
4	37, 9, 14, 30, 40
5	11, 10, 36, 29, 31
6	17, 35, 21, 27, 39
7	5, 1, 23, 2, 44
8	4, 24, 42, 6, 15
9	33, 12, 28, 41, 32

Table 2.3: The groupings of the individual batch data from the batch database.

For every model creation a number of variables are needed to determine the characteristics of the model. These include the number of rows in the Hankel matrix (i) the number of states that are used in the state space model (n) and a seed number. The value ranges of these variables are shown in Table 2.4. The seed number is used to randomly divide the batches into training and validation, 6 of the groups of

5 for the training, and 3 of the groups of 5 for validation. The seed value is used in a random number generator to determine the order of and divide the 9 groups.

Variable	Low End	High End
Number of Rows, i, in Hankel	5	15
Number of states	1	10

Table 2.4: Range of the model formulation variables; the number of rows in the Hankel matrix and the number of states determined during model creation.

The three models that had the smallest average MASE for each model structure are then validated on an additional 30 batches that had different initial conditions and input changes different to the original 45 batches. Thus it can be readily determined which model had the best predictions. The average MASE and standard deviation of the MASE for each batch is compared for these 3 models of each model structure. The model with the smallest average MASE for each model structure are considered the best model for that model structure. The Final model for each model structure are then used within the closed-loop MPC framework.

The mean and standard deviation of the MASE of the additional 30 batches are shown in Table 2.5 for the best model of each model structure. For each batch, the observer is taken out of the loop to make predictions from the twenty-fifth sampling instance, i.e., 12.5 minutes into the batch, to the k_f sampling instance /end of the batch. For the first 25 sampling instances the observer is run to estimate the SID states. This number was chosen based on 25 iterations and is found to be ample time for the predicted states to converge with the actual states and provide accurate predictions.

What can be seen in the Table 2.5 is that both the RFPH and RFP0H models have the greatest predictive ability with both having close to the smallest mean and standard deviation MASE. The RFP0H model performs slightly better than the RFPH model. A point to note is that both means are well within a standard deviation of each other, and thus it is difficult to state superior performance. Both hybrid models only use 1 state for the SID. The next best performing model is the RFP2SIDH with over two times the mean MASE than either of the hybrid model structures. The RFP2SIDH had 2 states for the residual SID and 7 states for the first principle SID. The worst model structure is the P1SID having twice the average error of RFP2SID and over four times the average error of either RFPH and RFP0H. Interestingly, the P1SID model structure performance is worse than the first principle models. This indicates extremely poor predictive ability from the P1SID model structure. The P1SID model used 2 states for the SID. Figure 2.8 and 2.9 show a typical model batch with the predictive ability of the best model for each model structure.

The results demonstrate that adding a first principle model improves the prediction ability of the model, compared to both the first principle model and compared to an SID of the first principle, as the

RFP2SIDH, RFPH, and RFP0H all outperform the first principles and P1SID. Additionally, the results show that using a first principle with constant inputs instead of variable inputs in the present application doesn't decrease the predictive ability of the model, as the RFP0H and RFPH models both had similar MASE averages and standard deviations. This can be explained by looking at the first principle and constant input first principle outputs in Figure 2.9. It can be seen that both FP and the FP0 follow the same trend. While there is deviation between the two, that mainly occurs in the later half of the reaction, and the deviation is a small difference between the relative magnitudes of the two. Thus the FP0 is still able to provide useful information for the RFP0H model to facilitate robust predictions. The next section utilizes these best models within an MPC framework to compare closed-loop performance.

Model	MASE Average	MASE SD
RFPH	24.43	12.31
RFP0H	22.63	10.49
P1SID	109.91	36.84
RFP2SIDH	54.21	31.66
FP	75.00	16.31
FP0	75.06	17.06

Table 2.5: Average and Standard Deviation of the Mean Absolute Square Error of each batch for the best version of each model structure and the first principle models for the additional 30 batches.

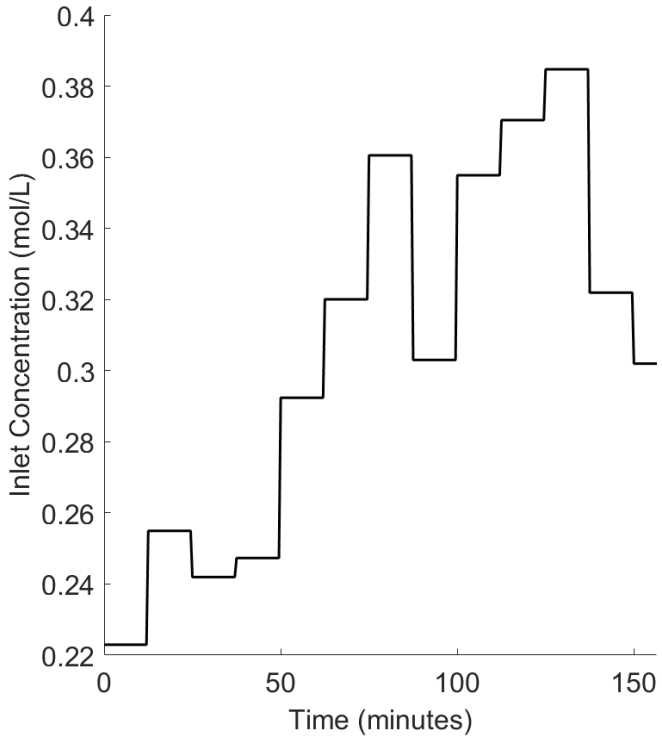

Figure 2.8: The input profile for a typical validation batch for 4 different modelling approach's best models, 2 first principles, and the process output.

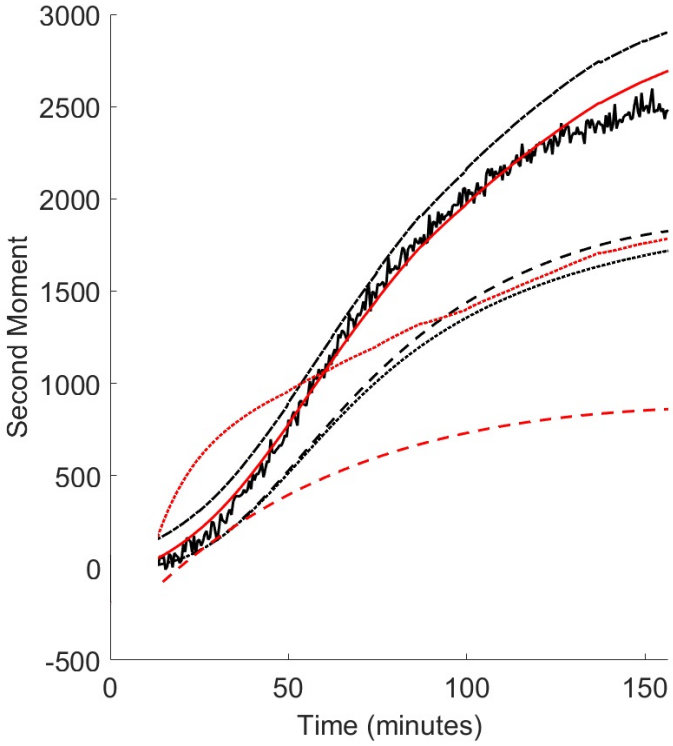

Figure 2.9: The model prediction profiles for a typical validation batch for 4 different modelling approach's, the best models, 2 first principles, and the process output. The second moment model predictions start at 12.5 minutes. The process output (black -), the first principle model (black - - -), the first principle model with constant input (black ..., the RFPH model (black -.-.), the RFP0H model (red red-black), the P1SID model(red red - - - black), and the RFP2SIDH (red red ... black).

2.4.2 MPC Performance Model Comparison

The best models for RFP0H, RFP2SIDH, and P1SID are used in an MPC framework. Note that the first principles model itself (the RFPH model) is not used to keep the comparison restricted to linear model predictive control implementations. Note that if the first principles model is relatively simple and amenable for online implementation, it could very well be utilized in a nonlinear MPC framework. Each model structure is tested on 30 batches with different initial conditions and set-points. For each model structure tested, the initial conditions and set-points of the 30 batches are consistent.

The set-point for each of the 30 batches are determined randomly between 1.1 to 1.5 times the magnitude of the end of the batch output for the second moment first principle with constant inputs ($y_{fp0,2}$). The process output is typically a value between the multiples of $1.3-1.4$ times that of the first principle. If the initial input value for the inlet concentration exceeds 0.3 mol/L, the range is limited to $1.1-1.35$, and if the input value is below 0.14 mol/L, the range is limited to 1.3 to 1.5. The reason that the range of possible set-points are limited when the initial inlet concentration is high or low is to prevent a set-point that would require inputs outside the constraints to achieve. For example, if the initial inlet concentration is 0.4 mol/L and the set-point is a multiple of 1.5 times the final second

moment first principle with constant inputs, then to achieve that set-point, the inlet concentrations throughout the batch would need to be greater than the upper bound. The same is true for a small initial inlet concentration and small factor, but this would violate the lower bound instead. In essence, it is ensured that the set-points are achievable for the batch in question, and the various factors are summarized in Table 2.4.2.

Initial Inlet (mol/L) Concentration	Low End Multiplier	High End Multiplier
≤ 0.14	1.3	1.5
≥ 0.3	1.1	1.3
≥ 0.14 & ≤ 0.3	1.1	1.5

Table 2.6: The set-point multipliers for determining the set-point values of the second moment at the end of the batch based on the initial inlet concentration of each batch. Set-points are determined by randomly choosing a multiplier between the low and high end multiplier and multiplying the value with the second moment of the first principle with constant input at the end of the batch for the 30 controlled batches. The low and high end multiplies varies based on the initial inlet concentration.

The performance is quantified as the the absolute difference between the set-point and the actual output of the second output at the end of the batch. The mean and standard deviation for the 30 batches are determined for each implementation. The model structure that had the smallest average objective value is deemed to be the best model.

The mean objective value and standard deviation for each model structure is shown in Table 2.7. Figure 2.10 and 2.11 show an exemplar batch with the process outputs for the different model structures and their corresponding inputs. From the table it can be seen that the RFP0H has the best performance and has both a smaller mean and standard deviation than either of the other models. The RFP2SIDH model has the middle performance of the three, and it still has readily good results for a number of batches with its average Mean Performance Value (MPV) being only approximately 25% greater than the RFP0H model. Thus, the P1SID model has the worse performance of all the model structures.

The results show that using a first principle with constant inputs is a reliable way to improve controller performance. This is likely because of the information that the ODE first principle provides compared to the other models; where the RFP2SIDH model has to predict the non-linear first principle using a linear SID, and the P1SID model has to determine the output with no information pertaining to the first principles. The superior performance of the RFP2SIDH over P1SID makes sense with regard to earlier work where a similar performance is shown between the two model structures [36]. Assuming minimal lose of trend and information from a constant input first principle profile when compared to a varied input first principle, a RFP0H MPC can provide superior performance in a controller using information from the non-linear dependence of the initial conditions than the other models that do not have access to it.

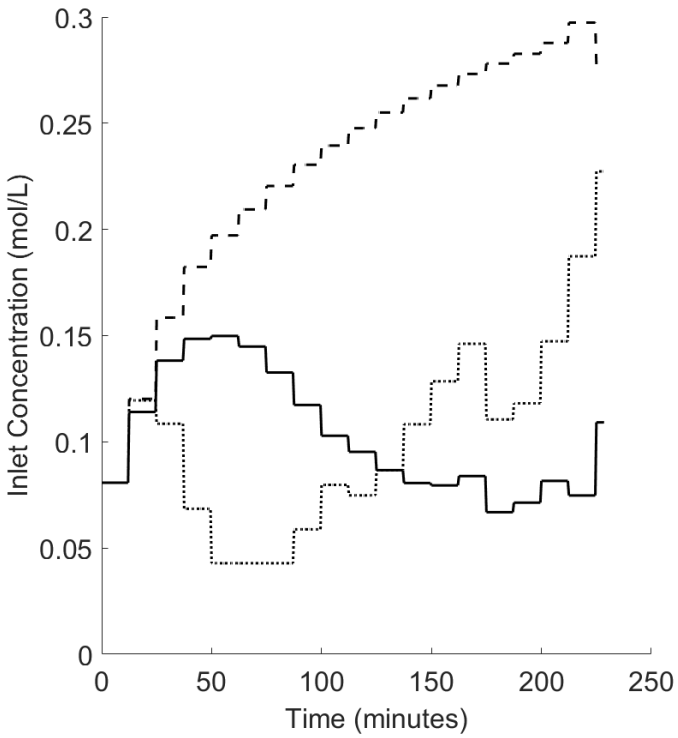

Figure 2.10: The input closed loop validation results for a typical batch for the three modelling approaches. The RFP0H model (black -), the RFP2SIDH model (black ...), and the P1SID model (black - - -).

Model	MPV	SDPV
RFP0H	192.65	220.94
RFP2SIDH	249.88	247.32
P1SID	474.95	409.72

Table 2.7: Mean (MPV) and Standard Deviation (SDPV) of the Performance Value for the Final models

2.5 Conclusion

In this paper, a novel parallel hybrid modelling approach was proposed which enabled the use of information from detailed first principles models that may be intractable if included in an optimization based control implementation. The hybrid model used the non-linear first principle dynamic model but with a constant input trajectory, and a second data driven SID model to explain the residual error between the process output and the first principle. This modelling method was illustrated using a fed batch crystallization reaction and allows the nonlinear dependence on the initial conditions to be captured while being tractable from an optimization perspective. This hybrid model's predictive ability was just as reliable as the a hybrid modelling technique with first principle with variable inputs, and had superior predictive performance compared to a parallel data driven hybrid model, a purely data driven model, and purely the first principle model. This readily implementable proposed approach enabled the im-

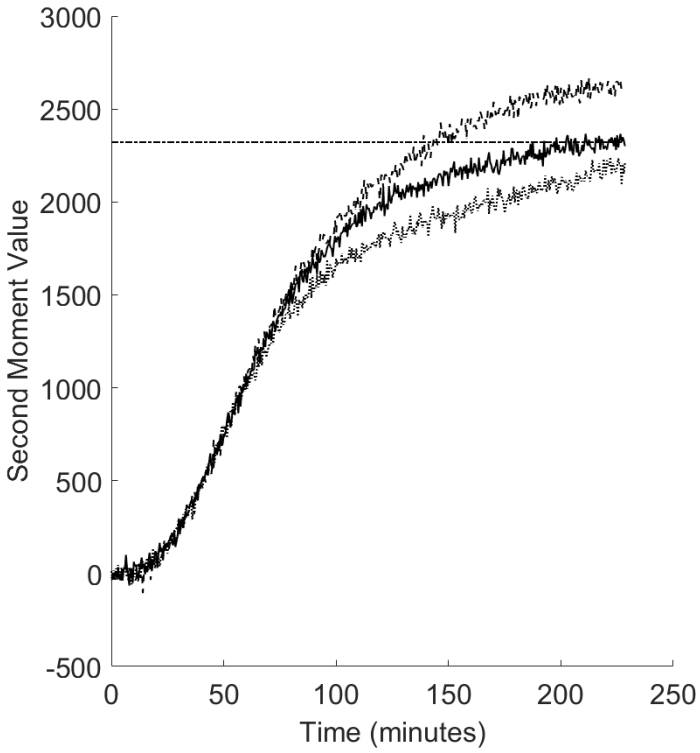

Figure 2.11: The output closed loop validation results for a typical batch for the three modelling approaches. The RFP0H model (black -), the RFP2SIDH model (black ...), and the P1SID model (black - - -). The set-point value for the end of the batch is shown as the horizontal line (black -.-.).

plementation of a model predictive controller with improved control performance, having at least 25% better control outcomes compared to the next best model. Simulation results demonstrate the usefulness of this technique.

Acknowledgements

Financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC) and the McMaster Advanced Control Consortium (MACC) is acknowledged and appreciated. Neither had any direct involvement in any step of the research and the paper writing for this paper.

Chapter 3

A Multiplicative Structure of a Parallel Hybrid Model for Optimal Temperature Model Predictive Control in a Laboratory Scale Batch Reactor

3.1 Introduction

Control of processes is difficult for a number of reasons including constraints, non-linearity, dead time, and uncertainty within processes [31, 69]. Model Predictive Control (MPC) over the last half-century has become one of the most ubiquitous tools in Control Systems within the Chemical Processing Industry to address these problems. MPC is capable of handling these challenges by providing process systems with variable input values that are likely to provide desirable outputs for the process. The reason that MPC is capable of this is that embedded within the MPC structure exist models that explain the system dynamics, the relationships of the inputs and outputs of a system [31, 69]. The model's ability to explain the process dynamics is vital to ensure accurate control of the system to obtain the desired output values [31, 69]. Thus, it is imperative for any model embedded within an MPC to closely reflect the reality of the system's dynamics.

The dynamical reality of most process systems is typically non-linear, consisting of a System of

Ordinary Differential Equations (ODE) with input, state values, and outputs. In theory, models that are based on natural laws/physics , mechanistic/first principle models, have the highest accuracy in explaining the relationship between the inputs and output, and these types of mechanistic models have been used as part of an MPC for excellent control realization [78, 10, 29]. Although, there are short comings to mechanistic models,specifically they are difficult to obtain. It can be an arduous task to completely identify the entirety of the dynamics of a systes. This is because to completely identify the dynamics, a large amount of experimentation and simplifying assumptions of the processing system are needed [2, 74]. Both of these things take a large amount of time and money from process engineers, technicians, and scientists to complete. Additionally, the difficulties of the model do not end once the model is identified, as active maintenance of the model is necessary throughout the model's use to ensure deviations do not occur between the model and the process, as process dynamics and model parameters are known to evolve over the lifetime of systems [2, 74]. These changes to process dynamics mean even more time and resources are spent maintaining the model. These challengers provide incentives for models that are easier to obtain and maintain while still providing accurate relationships of the model dynamics.

On the other spectrum of mechanistic models in terms of complexity, there exist linear data-driven models. Data-driven (DD) models use a predetermined model structure to explain the dynamic relationship of systems [55]. The simplest of which explains the system dynamics through a linear relationship. Two of the most common forms of linear DD models are Partial Least Squares (PLS) and Subspace Identification (SID). Both models work by using Single Value Decomposition (SVD) to find latent states that exist between the inputs and outputs [28, 33] . The major difference is that PLS is inherently static, and SID uses states to find the parameters of a Linear State Space System. A state space system has dynamics to predict future outputs based on current states and input values. Both of these model structures have found use within MPC frameworks [38, 39, 13, 32, 68, 51]. These models are much easier to create and maintain compared to mechanistic models due to their linear predefined structures. Although powerful, these models are limited due to their linear nature as they will never be able to fully explain the dynamics of non-linear processes.

A more complex non-linear DD model exists within Artificial Neural Networks (ANNs). Like the linear DD models, ANNs use a predetermined model structure to explain the dynamic relationships of the system. The model structure consists of layers of nodes that are summed and multiplied together with non-linear activation functions. The variations and permutations of layers and nodes within an ANN allow for a wide range of dynamic processes to be explained [42]. This has been shown in a number of MPC frameworks [1, 57, 15]. The problem with ANN is that a large number of layers and nodes require a significant number of parameters, typically in the thousands which is much higher compared

to mechanistic or linear DD models. This number of parameters mean that the model tends to over-fit the data and explain noise when there is a small data size or contains data of poor quality [16].

Hybrid models are an alternative modelling technique to either mechanistic or DD models. They work by synergizing multiple models together to create a more informed model than is possible from just one modelling domain. Typically a hybrid model will consist of models consisting from different modelling domains e.g. data driven, mechanistic, heuristic, allowing for information about the dynamics from these different domains to complement each other and have improved predictive results [84]. For a hybrid model to function there needs to be a way to combine the sub-model structures together. This can either occur in series or parallel form. In the series form, the output of one of the models becomes an input to the second model. One form of a series technique is a DD model predicting parameter values for a given mechanistic model. In the parallel form, the outputs of the models are summed together to predict the output of the process [81]. One form of a parallel model is a mechanistic model that explains the overall dynamics of the system, and a Data Driven technique explaining the error between the first principle and the process output.

Parallel hybrid models consisting of a mechanistic model and a residual DD model can be a powerful predictive technique. The mechanistic model can still be difficult to determine and optimize, but the ease of reformulating the DD model can alleviate these problems. Although, solving the optimization problem can still be difficult in real time optimization of the hybrid due to uncertainty and time to solve the non-linear problem [61, 2]. Simpler yet still accurate hybrid techniques are useful as they can be implemented into linear or quadratic MPC formulations, removing the time and uncertainly of non-linear optimization. This was accomplished by creating a hybrid model with two linear subspace based data models running in parallel [36]. This technique was further refined by keeping the non-linear dynamics of the mechanics with the residual SID, but using a consistent input profile into the mechanistic model. This modification allowed the First Principle (FP) model to be solved once at the start of the process, and required only the Linear SID model for MPC optimization. While this new technique provided a novel way to maintain the information of the dynamics of the mechanistic model while having a simple control structure, it may still be limited based on the hybrid model structure that combines the outputs of the two sub-models to predict the process output.

Parallel hybrid models typically consist of the two inputs being summed together to predict the final output. This can be seen in a wide range of literature where the two outputs are summed together [47, 73, 82]. There is only one parallel hybrid model within the literature where instead of summing the two outputs together, they are instead multiplied together [71]. There exists no place in the literature where parallel hybrid models that are summed together is compared against a parallel hybrid models where the output is multiplied together, or any work that delves into the hyper or meta heuristic of

determining a parallel hybrid model structure. This seems foolish, as in many processes, the error of a process can fluctuate through out the process, changing in many order of magnitudes. The range of value for a multiplication of error can be a smaller range than a residual. A factor that is multiplied by the first principle prediction may not need to contend with as much non-linearity and may be more consistent with how error propagates from one model to another.

Motivated by the previous factors, a new hybrid modelling approach is proposed that uses a multiplicative structure within the parallel hybrid model framework. Like the previous Linear hybrid model, this model uses a potentially intractable first principles model which is tailored for an MPC implementation. The mechanistic part of the hybrid model uses a first principle profile with a consistent input profile. The SID based data driven model is trained on the multiplicative factor from the division of the the true process outputs over the first principle predictions. This allows for the model to make more accurate predictions of the relationship between the first principle model and the process output. While also allowing for the first principle model to be solved once, and not be a part of the MPC formulation while still capturing the nonlinear dependence on the initial conditions. The proposed approach is demonstrated on a lab scale polymerization reactor where the control objective is to track a desirable temperature profile throughout the process by manipulating the jacket temperature via its inlet flowrate. The rest of the paper is organized as follows: Section 3.2 provides an overview of the Preliminaries of the paper, Section 3.2.1 provides a brief overview of the polymerization batch process, Section 3.2.2 provides an overview of Subspace Identification, Section 3.2.3 goes over hybrid modelling and the recently proposed hybrid modelling approach, Section 3.2.4 reviews a recently proposed MPC formulation that uses a linear hybrid model with a consistent input first principle. Section 3.3 describes the proposed modelling approach with Section 3.3.1 proposing the Multiplicative hybrid model that is the focus of this paper, Section 3.3.2 provides detail on model identification, Section 3.3.3 provides information on model validation, and Section 3.3.4 presents the MPC formulation. Section 3.4 provides the methodology and results of the best models and MPC control with Section 3.4.1 describing the model implementation and Section 3.4.2 describing the MPC implementation both of which for the simulated system. Section 3.4.3 provides information on the MPC implementation for the physical lab scale batch reactor. Lastly, Section 3.5 gives a final summary and conclusion to this work.

3.2 Preliminaries

This section provides a basic overview of preliminary information that serve as foundational knowledge of this paper. The first section discusses the dynamics of the polymerization batch reactor, the next section discusses the structure of the SID, followed by discussion of hybrid models, and the final section discusses the state of the art hybrid modelling technique, Residual First Principle 0 Hybrid (RFP0H),

and MPC control.

3.2.1 System of Interest: Batch Polymerization Reaction

A batch polymerization process is considered as the simulated system and the tested system, which is described in this section.

System of Interest: Dynamics

The System of Interest for this paper involves a laboratory scale polymerization batch reactor. The reactor is a closed system which at the insemination of the batch contains a fixed amount of monomer and initiator in the solution. Said components react together within the reactor to create a polymer solution. The reaction process is on going until the limiting monomer concentration is negligible within the reactor at which point the reaction is considered complete. The temperature and concentration of reactants within the reactor affects the final Molecular Weight and Molecular Number of the polymer. There exists a desirable temperature profile for the reactor to perform which has shown experimentally to provide high quality polymer product. Achieving desired outcomes is closely associated with matching the desired trajectory of the reactor temperature. To control the temperature of the system, a cooling jacket with cold water is run through the reactor removing heat from the system. The temperature of the jacket is considered for modelling and control purposes the manipulated input. Figure 3.1 shows the setup and configuration of the batch system.

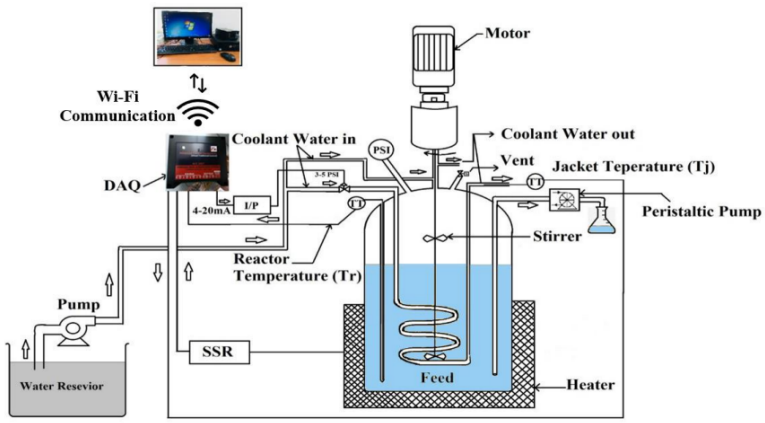

Figure 3.1: Schematic of laboratory scale polymerization batch reactor.

An ordinary differential system has been constructed of the system to help explain the dynamics and to be used as the system's test bed. There are four states to this system; monomer concentration, initiator concentration, the jacket temperature, and the reactor temperature. At the start of the reaction,

the vessel contains 0.7034 mol/L of monomer and $4.5E-3$ mol/L of initiator. Equation 3.1 shows the rate of dissociation for the initiator, R_i is the rate of change of initiator concentration, $\frac{-d[I]}{dt}$. A_d is the initiator reaction coefficient, $[I]$ is the initiator concentration, E_d is the activation energy for initiator dissociation, R is the ideal gas constant, and T_r is the reactor temperature. Equation 3.2 shows the rate of polymerization, R_p is the rate of change of the monomer concentration, $\frac{-d[M]}{dt}$. A_p is the polymerization reaction coefficient, ϵ is rate law of the initiator, θ is the rate law of the monomer, $[M]$ is the concentration of the monomer, and E_p is the polymerization activation energy. Equation 3.3 is the energy balance of the reactor which measures the rate of change of the reactor temperature, $\frac{dT_r}{dt}$. m_r is the mass of the reactor, c_{pr} is the heating capacity of the reactor, R_p is the rate of polymerization, V is the volume of the reactor, ΔH_p is the enthalpy of the reactor, U is the overall heat transfer coefficient, A is the heat transfer area, T_j is the temperature of the jacket, Q_h is the rate of heated added by the heating element, Q_s is the work provided to the system from the stirrer, and Q_{loss} is the heat lost to the surroundings. Equation 3.4 is the energy balance of the jacket which shows the evolution of the jacket temperature, $\frac{dT_j}{dt}$. m_j is the mass of the jacket, c_{pj} is the heating capacity of the jacket, F_c is the flowrate into the jacket, c_{pa} is the heating capacity of the cooling water, and T_c is the inlet temperature into the jacket.

$$R_i = \frac{-d[I]}{dt} = A_d[I]e^{\frac{-E_d}{RT_r}} \tag{3.1}$$

$$R_p = \frac{-d[M]}{dt} = A_p[I]^{\epsilon}[M]^{\theta}e^{\frac{-E_p}{RT_r}} \tag{3.2}$$

$$m_r c_{pr}\frac{dT_r}{dt} = R_p V(-\Delta H_p) - UA(T_r - T_j) + Q_h + Q_s - Q_{loss} \tag{3.3}$$

$$m_j c_{pj}\frac{dT_j}{dt} = UA(T_r - T_j) - F_c c_{pa}(T_j - T_c) \tag{3.4}$$

Equations 3.1 through 3.4 are used in two ways. One as a first principle approximation of the real physical process, and as an alternative to the process. The first principle approximation is used as part of the hybrid model for prediction and control Additionally, the equations are used as an alternative to the real physical system to test the control and prediction capabilities of the model before being implemented on the real system. There exists parameter mismatch between the first principle model and the physical process alternative. The parameters for the process alternative are randomly determined deviations that are between 0 to 81 % difference from the first principle. See table 3.1 for the differences between the parameters. It is assumed that the measurements are corrupted by noise, and noises are varied

Parameter	Process Value	FP Value	Percent Change	Unit
A_d	4.4E16	4.4E16	0%	1/s
E_d	1.44E5	1.44E5	0%	J
R	8.314	1.533E128.314	0%	J/(mol C)
A_p	2.833E9	2.833E9	+0%	$L^{0.75}/s^{0.75}$
ϵ	0.9	0.1709	-81.0%	-
θ	1.0001	1.0795	+7.93%	-
E_p	7.1E4	6.33E4	-10.8%	J
m_r	700	700	+0%	g
c_{pr}	120	120	+0%	$J/(gC)$
V	0.5074	0.6332	0%	L
ΔH_p	-8.1922E4	-8.1922E4	0%	J/mol
U	928	926	0%	w/(m^2C)
A	0.0662	0.0294	-55.6%	$/m^2$
Q_h	655.37	559.72	-14.6%	w
Q_s	14.9	14.9	0%	w
Q_{loss}	25	25	+0%	w
m_j	450	450	+0%	g
c_{pj}	7	7	+0%	J/(g C)
F_c	0 to 0.7	0 to 0.7	+0%	L/min
c_{pa}	4.184	4.184	+0%	J/gC
T_c	25	25	+5%	mol^2/kg^2C

Table 3.1: Parameters values used to solve the ODE systems of the equations in the batch polymerization reaction for both the simulated process outputs and the first principle (FP) model, the percent difference between the two scenarios, and the units of the parameters.

in a random uniform distribution by as much as a half standard deviation of the process data. Euler method is used to integrate the ODE with a 0.02 second interval. The sampling time of the physical and simulated systems are 0.5 seconds.

Data Base Generation and Use

15 batches of experimental data have been provided to construct the hybrid model. These batches contain time series data on the jacket temperature, the reactor temperature, and the control signal to control the flow rate of the jacket. Eight of which are closed loop and seven of them are open loop. The closed loop tries to track the desired temperature profile that provides desirable polymer concentration, molecular distribution, and quality at the end of the batch. Of the 15 batches, 6 batches were determined to be consistent and close to approximating the desired temperature profile. These six batches are used in the training and validation of the hybrid model.

There exists a relationship between the control signal to the valve which is measured and the flow rate into the coolant jacket which is inferred from the signal value. This relationship is linear in nature and tends to provide decent approximations over small sustained jumps in the profile. Although, this relationship breaks down with frequent changes in the signal and large deviations between small time

points, so much so that it can be detrimental to the predictive ability of the system. It is decided to use the jacket temperature as the input into the system for modelling and control purposes. The measured jacket temperatures are fed directly into the ODE system to calculate the reactor temperature, monomer concentration, and the initiator concentration. The monomer and initiator concentration are not directly measured and assumed to be states of the system. The reactor temperature is measured and assumed to be the output of the system.

To generate the data meant to be the test bed of our system, a matrix U with the dimension of mxN, parameter values, a system of ordinary differential equations, and the initial conditions are used. m is the number of inputs into the system which has a value of 1, the jacket temperature. The jacket temperature profiles are the same profiles that are measured on the physical batch system. N is the number of sample iterations/time steps that occur in the batch which vary from batch to batch. The initial conditions for the reactor temperature is the value of the first temperature measurement of the physical system, and for the monomer and initiator concentration are $4.5E-3$ mol/L and 0.7034 mol/L, respectively.

3.2.2 Subspace Identification

Subspace Identification is a data driven modelling technique which uses Single Value Decomposition, QR decomposition, or other similar techniques to create a Linear Time Invarient model in the form of a state space equations [66]. The form of the equations are shown in Equations 3.5 and 3.6. $x_{sid}[k]$ is the state vector at the k^{th} sampling instance with the dimensions $nx1$. n is the number of states of the SID system, and is determined during modeling identification. $u[k]$ is the input vector at the k^{th} with the dimensions $mx1$. m is the number of inputs of the SID system. $y_{pred}[k]$ is the output vector at the k^{th} sampling instance with the dimensions $lx1$ l is the number of outputs of the SID system. A, B, C, and D are the matrix parameters which are identified during model creation.

$$x_{sid}[k+1] = Ax_{sid}[k] + Bu[k] \tag{3.5}$$

$$y_{pred}[k] = Cx_{sid}[k] + Du[k] \tag{3.6}$$

The creation of the SID model involves two steps. The first being to create a state sequence trajectory using SVD which finds a row space intersection of the Hankel matrix. This state sequence represents latent information contained in the input-output data which undergoes SVD using the highly organized Hankel matrix containing both input and output data. Once the state sequence has been determined, this along with the input and output data can be used together to perform a linear regression to determine the parameter matrices, A, B, C, and D [66].

3.2.3 Hybrid Modelling

Hybrid Models consist of multiple model structures combined together for superior predictions. Often this will occur in combination of different model structures. Such as mechanistic models used to describe the system, and data drive models which are used to provide a more informative model. There are two common hybrid model strategies; parallel and series. Series consists of the output of one model to be the input of the other. Parallel, which is the focus of this paper, consists of the two sub-models being run concurrently and independently of each other, each with their own inputs and outputs, of which the input can be the same or different. The output of the two models are then combined together for the final prediction of the systems outputs [84].

A parallel hybrid model consisting of both a first principle and a SID model has been studied over several literature papers. In this form, the first principle is meant to explain the overall non-linear dynamics of the system. As the first principles are not capable of fully explaining the dynamics of the system due to parameter error and assumptions used to construct the model, there exists error between the first principle predictions and the physical system outputs. The SID model is then used explain the error that exists between the two [35].

The structure of a parallel hybrid model consisting of a first principle and an SID are shown in the Equations 3.7 to 3.11. Equations 3.7 and 3.8 represent the first principle equations, where x_{fp} represents the vector of states, u is the system inputs, y_{fp} is the outputs of the system, $f(x_{fp}, u)$ represents the system of ordinary equations that explain the dynamics of the state, and $h(x_{fp}, u)$ is the equations that explain the relationship between the state and the system outputs. Equations 3.9 and 3.10 are the SID component of the hybrid model. $x_{sid}[k]$ is the state vector at the k^{th} iteration, $u[k]$ is the input vector at the k^{th} iteration, and $e_{pred}[k]$ is the prediction of the error at the k^{th} iteration. Equation 3.11 shows the parallel structure of the model with the error prediction of the SID, $e_{pred}[k]$, and the output prediction from the first principle, $y_{fp}[k]$. These are summed together to predict the output of the process, $y_{pred}[k]$.

$$\dot{x}_{fp} = f(x_{fp}, u) \tag{3.7}$$

$$y_{fp} = h(x_{fp}, u) \tag{3.8}$$

$$x_{sid}[k+1] = Ax_{sid}[k] + Bu[k] \tag{3.9}$$

$$e_{pred}[k] = Cx_{sid}[k] + Du[k] \tag{3.10}$$

$$y_{pred}[k] = e_{pred}[k] + y_{fp}[k] \tag{3.11}$$

3.2.4 Linear Hybrid 0 Model for MPC

The mechanistic equations of the hybrid model while highly informative about the dynamics of the system have difficulties when it comes to implementing them into processes for prediction and control. The reason for this is two fold. Firstly, maintaining first principle models can be expensive and time consuming. As systems go through use, deviations in process dynamics occur due to entropy in the system, meaning the predictive ability decreases over time. Thus requiring updates of model parameters over time which requires optimizing and reviewing experimental data. Additionally, because of the non-linearity of dynamics systems, it can be difficult to solve the optimization in real time. Such difficulties can provide sub-optimal input suggestions to the process and may not provide suggested inputs at the desired frequency.

To address it, a novel modification was made to the hybrid model design in McKay et al that changed the first principle structure of the system, known as a Residual First Principle 0 Hybrid (RFP0H) model [64]. In this modified hybrid structure design, a constant input profile is provided to the first principle (FP) model, where the SID is fed the realized input of the process. The constant input trajectory that is fed into the FP is consistent with the initial input/0^{th} iteration of the sampled input, hence the names First Principle 0 (FP0) and RFP0H. This modification to the first principle means that the first principle will remain consistent through out the process and needs to be solved only once at the start of the batch process. For MPC, there exists a desired output trajectory or values that an operator wishes their system to undergo. With the the FP0, the desired output can be subtracted by the output of FP0 model, providing a desired error trajectory. Thus control with the MPC can be completed only using the Linear SID sub-model. This allows for linear or quadratic optimization of the MPC, which is trivial to solve compared to the non-linear dynamics that would be included in the mechanistic model, but at the same time this modification still allows for the first principle to explain majority of the variance of the system, making the predictive job of the SID easier and more reliable.

Equations 3.12 to 3.16, show the model structure of the RFP0H Model formulation. Equation 3.12 explains the the states of the FP0 model, where x_{fp0} is the state vector of the system, u_0 is the constant value input vector, and $f(x_{fp0}, u_0)$ represents the Ordinary Differential Equation. Equation 3.13 represents the outputs of the first principle of the FP0 model, where y_{fp0} is the output vector of the FP0, and $g(x_{fp0}, u_0)$ is the system of equations that relate the states and the output. Equations 3.14 and 3.15 represent the residual SID prediction, where $x_{sid0}[k]$ is the states of the SID, $e_{sid0}[k]$ is the predicted residual output vector, $u[k]$ is the input vector which fluctuates, and the parameter matrices

A, B, C, and D. Note that u_0 is the fix input and $u[k]$ is variable, they are not equivalent to each other except at the start of the process. Equation 3.16 represents the hybrid component of the model where the first principle output and the SID residual prediction are added together to predict the process output, $y_{pred}[k]$.

$$\dot{x}_{fp0} = f(x_{fp0}, u_0) \tag{3.12}$$

$$y_{fp0} = g(x_{fp0}, u_0) \tag{3.13}$$

$$x_{sid0}[k+1] = Ax_{sid0}[k] + Bu[k] \tag{3.14}$$

$$e_{sid0}[k] = Cx_{sid0}[k] + Du[k] \tag{3.15}$$

$$y_{pred}[k] = y_{fp0}[k] + e_{sid0}[k] \tag{3.16}$$

While the approach in McKay et al provides a way to use a first principle structure at the start of the process and allows for control actions to be determine exclusively using the linear SID for prediction and control for a desired error, the structure of the model may not be fully utilizing all information that can be provided by the two sub-models. Put another way, there maybe better ways of combining knowledge of the two a hybrid model to obtain a more informative predictive model while still maintaining the simplified control structure of the RFP0H model. In the context of the RFP0H, a variation of the process that more accurately relates the relationship of error to the first principle would be valuable in both prediction and control and is presented in the next section.

3.3 Proposed Modelling and Control Strategy

In the present work, a linear hybrid modelling approach containing a first principle model and a SID model, where the outputs of the two sub-models are multiplied together is proposed and implemented into an MPC framework. This is compared to the old linear hybrid modelling framework where a first principle model and an SID model are summed together to predict the output. The modelling approach is first presented which is then followed by the identification and validation of the model. The MPC control strategy is presented in the final subsection.

3.3.1 Hybrid Modelling Strategy

A novel change to the RFP0H is proposed where instead of the outputs of the two submodels being summed together to predictive the process output, they are instead multiplied together. This means that instead of predicting the error of the process, the SID output is predicting a factor to multiple the first principle output to predict the process output. This factor is refereed to as the Scaling Factor (SF). Additionally, the first principle structure of the model remains unchanged, except the input into the first principle is a representative of a typical input for the process instead of constant.

Equation 3.17 to 3.21 show the structure of the Scaling Factor First Principle 0 Hybrid (SFFP0H Model). Equations 3.17 shows the first principle state prediction with x_{fp0} representing the states, u_0 represents the consistent input, and $f(x_{fp0}, u_0)$ represents the ODE system. Equation 3.18 shows the first principle prediction of the outputs, with y_{fp0} representing the first principle output predictions, and $g(x_{fp0}, u_0)$ representing the system of equations between the states and the outputs. Equation 3.19 shows the SID state equation, with $x_{sid0}[k+1]$ the SID state vector, and $u[k]$ the realized process input vector. Equation 3.20 shows the prediction of the Scaling Factor, SF_{sid0}. Finally, equation 3.21, shows the structure of the multiplicative nature pf the hybrid model, where the SF and the first principle process output are multiplied together to predict the process output, $y_{pred}[k]$.

$$\dot{x}_{fp0} = f(x_{fp0}, u_0) \tag{3.17}$$

$$y_{fp0} = g(x_{fp0}, u_0) \tag{3.18}$$

$$x_{sid0}[k+1] = Ax_{sid0}[k] + Bu[k] \tag{3.19}$$

$$SF_{sid0}[k] = Cx_{sid0}[k] + Du[k] \tag{3.20}$$

$$y_{pred}[k] = SF_{sid0}[k]^T * y_{fp0}[k] \tag{3.21}$$

3.3.2 Model Identification

Presented in this section is the general procedure for building the SID component of the SFFP0H model. To identify the SF component of the model, Subspace Identification is used to construct the State Space model. There are three main steps of Subspace Identification. The first is to construct a highly organized matrix of the input and output data known as a Hankel Matrix [66]. To accommodate for

multiple batches of data, a Pseudo Hankel matrix is used which incorporates discontinuous data [61]. The second step is to create a theoretical state matrix containing state vectors that correspond to the row intersection of the input output data. The final step of model identification is to perform a linear regression using the input, output, and found state values to determine the values of the parameter matrices, A, B, C, and D [66].

Equation 3.22 shows the generic structure of the Pseudo Hankel matrix. The entries of this matrix is constructed from smaller Hankel matrices which take the form shown in equations 3.23 and 3.24. Where Y^b_{H1} is the hankel matrix of the output data for the b^th batch starting at the first data instance, and Y^b_{H2} is the hankel matrix of the output data for the b^th batch starting at the ith plus one data instance. The instances of the output hankel matrices, $y^b[1]$, $y^b[2]$, ect are vectors of the output data for the bth batch with the dimensions $(lx1)$. l is the number of output variables. N_t is the number of batches which contain data. The Hankel matrix for the inputs are also included in the Pseudo Hankel Matrix and have the same form as is seen in equations 3.23 and 3.24. The only difference in structure is that the inputs have the dimensions of $(mx1)$. Where m is the number of inputs into the system. i and j are two parameters that need to be chosen for the creation of the Hankel matrices, and represent the number of rows and columns that occur in each Hankel Matrix, respectively. i will remain consistent between batches, but j is variable from batch to batch depending on the length of the data. The dimensions of the pesudo Hankel matrix is $i(2m+2l)xM$ where M is the sum of all the js from all the batch data.

$$H = \begin{bmatrix} Y^1_{H1} & Y^2_{H1} & \dots & Y^{N_t}_{H1} \\ U^1_{H1} & U^2_{H1} & \dots & U^{N_t}_{H1} \\ Y^1_{H2} & Y^2_{H2} & \dots & Y^{N_t}_{H2} \\ U^1_{H2} & U^2_{H2} & \dots & U^{N_t}_{H2} \end{bmatrix} \tag{3.22}$$

$$Y^b_{H1} = \begin{bmatrix} y^b[1] & y^b[2] & \dots & y^b[j] \\ y^b[2] & y^b[3] & \dots & y^b[j+1] \\ \vdots & \vdots & \ddots & \vdots \\ y^b[i] & y^b[i+1] & \dots & y^b[i+j-1] \end{bmatrix} \tag{3.23}$$

$$Y^b_{H2} = \begin{bmatrix} y^b[1+i] & y^b[2+i] & \dots & y^b[j+i] \\ y^b[2+i] & y^b[3+i] & \dots & y^b[j+i+1] \\ \vdots & \vdots & \ddots & \vdots \\ y^b[2i] & y^b[2i+1] & \dots & y^b[j+2i-1] \end{bmatrix} \tag{3.24}$$

Once the Hankel matrix has been constructed from the batch data, it is necessary to find the states of the system using Single Value Decomposition (SVD) which occurs twice. Equation 3.25 shows the

decomposition into three matrices U, S, and V^t. U and S are further divided into sub-matrices of which U_{11}, U_{12}, and S_{11} are important. U_{11} has dimensions of $(mi + li)x(2mi + n)$, U_{12} has the dimensions of $(mi + li)x(2li - n)$, and S_{11} has the dimensions of $(2mi + n)x(2mi + n)$. These three sub-matrices are then multiplied and undergo SVD to create the new matrices $U_2S_2V^t$ which can be seen in equation 3.26. These matrices are also broken down into sub-matrices of which U_q is important. U_q has the dimensions of $nx(2li - n)$. U_q, U_{12}, and $H1$ are multiplied together to find the theoretical states which is shown in equation 3.27.

$$H = USV^t = \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix} \begin{bmatrix} S_{11} & 0 \\ 0 & 0 \end{bmatrix} V^t \tag{3.25}$$

$$U_{12}^t U_{11} S_{11} = U_2 S_2 V_t = \begin{bmatrix} U_q & U_q^T \end{bmatrix} \begin{bmatrix} S_q & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_q^t \\ V_q^{t,T} \end{bmatrix} \tag{3.26}$$

$$\widehat{x_{theoretical}} = U_q^t U_{12} H_1 \tag{3.27}$$

Once the states of the SID are found, all that remains is to find the parameter matrices of the State Space Equations. To do that, the input and output data are necessary. The data can be found within the Hankel Matrix. The output data is found within the Hankel Matrix in the dimensions of $(i(l+m)+1 : i(l+m)+l)x(1 : M-1)$, and the input data is found in the dimensions of Hankel Matrix in the dimensions of $(i(2l+m)+1 : i(2l+m)+m)x(1 : M-1)$. This data is represented by Y_j and U_j and are seen from equations 3.29 and 3.28, respectively. The regression occurs in the form of Equation 3.28. Φ represents the parameters matrices to be calculated, A, B, C, and D. By performing a linear regression with the corresponding input, output, and state values, the parameter matrices are found.

$$U_j = H(i(2l+m)+1 : i(2l+m)+m)x(1 : M-1) \tag{3.28}$$

$$Y_j = H(i(2l+m)+1 : i(2l+m)+m)x(1 : M-1) \tag{3.29}$$

$$\Phi * \begin{bmatrix} \widehat{x_2}[:, 1 : M-1] \\ U_j \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} * \begin{bmatrix} \widehat{x_2}[:, 1 : M-1] \\ U_j \end{bmatrix} = \begin{bmatrix} \widehat{x_2}[:, 2 : M] \\ Y_j \end{bmatrix} \tag{3.30}$$

3.3.3 Model Validation

Model validation evaluates the predictive capability of the model using data not used within the training portion of the model creation. There are two steps to model validation, the initial state estimation and

the prediction phase. For a SID model to make accurate predictions, the model needs good value of the states. As the states are latent values of the process and are not inherently knowable, they need to be inferred by comparing the measured output against the predicted output to modify the state. This occurs iteratively through out the process until there is a convergence between the predicted and measured output. At that point, it is assumed that the states are accurate and the model is ready to start predicting. The model is fed the input data to predict the output without information from the measured data. This predicted data is then compared to the output using a Metric to measure the amount of error.

A Luenberger is used as the observer to correct the states. The form of the state prediction with the observer is shown in equation 3.31. L is the pole gain matrix which multiplies the error of the predicted and measured output, $\hat{y}[k]$ and $y_{measured}[k]$, at the k^{th} sample instance. The gain matrix is determined using the SID parameter matrices to correct the state for the next time instance. The rest of the equation $\hat{x}[k+1] = A\hat{x}[k] + Bu[k]$ is the typical state prediction where the state at the $k^{th}+1$ iteration is predicted based on the states and inputs of the previous iteration.

$$\hat{x}[k+1] = A\hat{x}[k] + Bu[k] + L(\hat{y}[k] - y_{measured}[k]) \tag{3.31}$$

To determine the predictive ability of the matrix, an error metric needs to be used. For this process, the metric of use is Mean Absolute Scaled Error (MASE). The form of MASE calculation is shown in 3.32 [21]. In the numerator of the equation of the error between the measured and predicted output is summed from the first prediction instance to the last prediction instance. The denominator sums the difference of the output from the second sampling instance to the final instance. The numerator measures the total error from prediction, and the denominator scales the error based on the changes of the output. This measurement is down for multiple batches and the average MASE is considered.

$$MASE = \frac{\sum_{t=1}^{tp} |e_t|}{\frac{t_p}{t_p - 1} \sum_{t=2}^{t_p} |y_t - y_{t-1}|} \tag{3.32}$$

Figure 3.2 shows the two phase of model validation. The observer phase is the first phase in which the measured output is compared against the predictive output to correct the state of the SID states. This is done for v iterations where it has been reliably determined to convergence between the predicted and measured output. The second phase is the predicted phase, where no information of the measured output is used. Input is fed into the model and used to make the process predictions for the remainder of the batch.

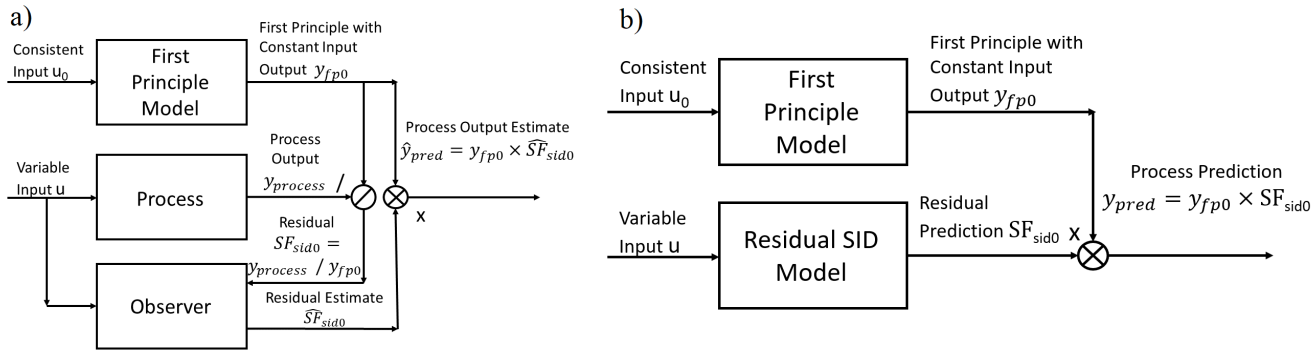

Figure 3.2: Model structure of the SFP0H model in the validation. a) shows the structure with the observer, b) shows the prediction phase.

3.3.4 Model Predictive Control Design

At the sampling instance k_c, the Model Predictive Control Loss function consists of the following form:

$$
\begin{aligned}
& u[k], k = k_c, k_c+1, \ldots, k_{horz} = arg(min \quad J = \sum_{k_c}^{k_c+horz} (SF_{sp}[k] - SF_{sid}[k])^2) \\
& u_{lb} \leq u[k] \leq u_{ub} \qquad k = (k_c, k_c+1, ..., k_c+horz) \\
& u[k-1] - u_{diff} \leq u[k] \leq u[k-1] + u_{diff} \\
& \tilde{x}_{sid0}[k_c] = \hat{x}[k_c] \\
& \tilde{x}_{sid0}[k+1] = A\tilde{x}_{sid0}[k] + Bu[k] \\
& \tilde{SF}_{sid0}[k+1] = C\tilde{x}_{sid0}[k+1] + Du[k+1] \\
& SF_{sp}[k] = y_{sp}[k]/y_{fp0}[k]
\end{aligned}
\tag{3.33}
$$

The objective function: The aim of the controller is to minimize the squared error between the predicted SF_{sid} and the desired SF_{sp} between the current iteration of the batch process to $horz$ iterations in the future, k_c to $k_c + horz$. The desired scaling factor comes from the prediction of the FP0 and the desired output of the process. The desired output is divided by the FP0 for each iteration to determine the desired scaling factor. The desired SF is calculated using the SID model. The MPC is implemented in a moving horizon fashion where a fixed horizon of predictions are predicted. This becomes a receding horizon problem at the end of the batch. The reason for this is that the desired output trajectory has a limited number of iterations. When the horizon is longer than the remaining trajectory, the horizon is shrunk to perform accordingly.

The constraints: A number of constraints are added to the function to ensure good control and prevent large disturbances in the system. All input values determined by the model are bounded between a higher and lower input value, u_{lb} and u_{ub}. These bounds are to ensure that the inputs are physically realizable

as constraints from the system make implementing inputs outside these bounds difficult. Additionally, the model was not trained with inputs outside these bounds which would limit the model's predictive ability. Additionally, for each step change that the models computes, the absolute difference between the inputs needs to smaller than a tolerance of u_{diff}. This is to ensure that the model predicts inputs that have similar differences and trajectories that the model was trained on and that it is difficult for the system to realize large step changes over small time ranges. Additionally, the most recent measured value of the state needs to be used within the state prediction, and all future values of the state need to have the relationship of the State Space Equation. The same holds for predicting the Scaling Factor, the prediction of future SF should follow the relationship of the State Space Model. Finally, to calculate the SF set point at the kth iteration, $SF_{sp}[k]$, the output set point, $y_{sp}[k]$ needs to be divided by the predicted value of the setpoint from the FP0 model, $y_{fp0}[k]$.

The SF set point, $SF_{sp}[k]$ represents a target for the SID model to try and obtain. Much of the variance within the system is explained the FP0 model, meaning that the SID can explain more of the remaining variance of the system. This allows for the model to make more accurate predictions of the necessary inputs to get the desired outputs.

The MPC is solved iteratively throughout the course of the batch reaction. Initially the observer is run to ensure accurate state values and to make accurate output predictions. The input, set point, states, and parameters are fed into the MPC controller where an input values are chosen for every iteration of the control horizon. The first iteration of determined desirable inputs is implemented onto the system, the rest are discarded. The system then continues to run until the next output measurement where the states are updated and the MPC can run the optimization problem again. Figure 3.3 shows the closed loop structure of the MPC and what needs to be fed into the MPC for accurate predictions.

3.4 Application to Laboratory Scaled Polymerization Batch Reactor

This section looks at the performance of the models in both a predictive and closed loop setting. The SFFP0H and the RFP0H are both tested and compared against each other. 50 instance of each model structure are tested using various values of the rows of the Hankel matrix and number of states of the systems. 6 batches of experimental data are used in model creation; four for training and two for validation. Models with the smallest MASE of each model type are tested on the MPC for comparison. Multiple batches undergo closed loop control with both model structures to compare performance against each other. The initial conditions between batches are variable.

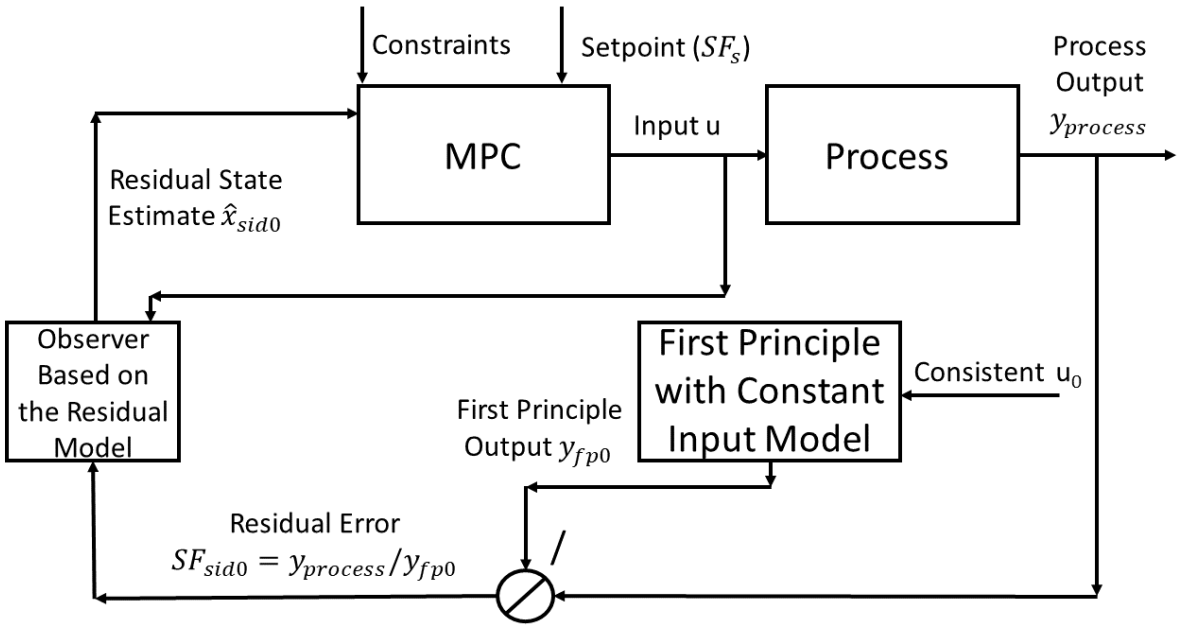

Figure 3.3: Closed Loop Schematic of the SFFP0H model within the MPC framework.

3.4.1 Model Creation and Comparison

From the original 15 batches of experimental data from the polymerization batch reactor; 6 batches have been determined to fit the profile and modelling technique well enough to use as part of model creation. 4 batches are used for training and two batches are used for validation. Additionally, this data is used to create simulated data as well, where the input, jacket temperature, is kept consistent between the experimental and simulated system. The discrepancy is between experimental output and the simulated output.

To create the model, the model parameters needs to be determined. The parameters that can be manipulated are the number of rows in the Hankel matrix formulation, i, the number of realized states in the SID model creation, n, and a seed number that divides the batches into training and validation. i and n affect the relationship that SID finds between the input and output data, and the best values of each parameter will vary based on the input output data. The seed will pick four batches for training and the two batches for validation. The values of these parameters are determined randomly for 50 instances of each model structure. There are bounds of each of the model parameters for each instance of a model and are shown in Table 3.4.1.

The MASE is found for each validation for the prediction portion of validation. The MASE of the FP0 model is also computed. The MASE of the first principle is subtracted by the MASE of the predicted output. This metric, called explained MASE, shows the improved performance of the SID model. A larger value shows that more error has been explained by the SID model. The averaged

Parameter	Lower Bound	Upper Bound
i	5	15
n	1	10
seed	1	1000

Table 3.2: The range of parameters for model construction for each instance of a model structure.

explained MASE of the validation batches is used to determine the overall performance of each instance of a model structure. The largest average explained MASE for each model structure is determine to have the best predictive ability of that model structure. The largest removed MASE of each model structure can then be compared against the values of the other model structures. The largest average MASE can then be determined to have superior predictive ability.

Mean of the explained MASE of the best performing model structure are shown in Table 3.4.1. What can be seen from the table is that the predictive ability of the SF is better than compared against the Residual. That SF has two states in the SID, and the Residual has two states in the SID. Figure 3.4 shows a typical model predictive ability for the best model of each model structure.

Model structure	Removed MASE
RFP0H	4.147
SFFP0H	8.21

Table 3.3: The range of parameters for model construction for each instance of a model structure.

The results shows that the model structure of a hybrid model can affect the predictive ability of a model. A multiplicative factor is capable of removing approximating twice as much error in the first principle prediction compared to an additive model structure. Reasons for this could included increased linearity in predicting a scaling factor versus a residual, and that the scaling factor more closely resembles the nature of error/uncertainty in the model [3, 40]. The best version of each model structure are now used within the MPC framework.

3.4.2 MPC Performance: Simulation

The best models of the SFFP0H and RFP0H model structures are used within an MPC framework. Each model is tested on 30 batches. The desired output between the batches are consistent, but the initial conditions vary between the batches. The initial temperature of the reactor and jacket temperature are chosen randomly. The reactor temperature is randomly determined between the bounds of 30 and 50 degrees Celsius. The jacket temperature is typically between 1 and 5 degrees cooler than the reactor temperature. Once the reactor initial temperature is determined, the jacket initial temperature is determined randomly to be 1 to 5 degrees cooler.

The performance of the control for each batch is quantified by the sum of squared error between the

Model structure	Mean SSE	Standard Deviation SSE
RFP0H	3.24E4	1.25E3
SFFP0H	3.16E4	83

Table 3.4: Mean and Standard Deviation of the Sum of Squared Error for 30 batches for each model structure.

desired reactor temperature and the actual reactor temperature from reaction start to end. The mean and standard deviation of the sum of error is found for each model structure. The model structure with the smallest average sum of error is considered the best as that model performs the closest to the desired temperature profile.

The means and standard deviations of squared error for the 30 batches are shown for each model structure in Table 3.4.2. Figures 3.5 and 3.6 show an exemplar batch showing the output of the SFFP0H model. From the table it can be seen that SFFP0H model has slightly smaller average error indicating that this model had the better performance. The standard deviation of the SFFP0H model is considerably smaller for the SF model compared to the Residual model. The results show that a multiplicative structure of error can help both in the prediction and control of process. The SID is more capable of explaining the variation which helps make more informed decisions during control for a multiplicative structure versus an additive one. Overall, a hybrid model with a first principle with consistent input is still an informative way to structure a hybrid model. Additionally, a multiplicative model more closely resembles the structure of a error within this process or captures the relationships better between the first principle and the process output. Further exploration of the nature and structure of a hybrid model could continue to provide even more superior predictive and control methods.

3.4.3 MPC Performance: Physical System

The Matlab code for the MPC that uses the SF model within it has been provided to our collaborators which own, maintain, and operate the scale batch reactor. The obtaining of physical results through batch experimentation is currently on going. An exemplar run is shown in figures 3.7 and 3.8. It can be seen that the MPC does not perfectly track the model output and more work needs to be done before desirable control is faithfully enacted.

3.5 Conclusion

In this paper, The SFFP0H model was used as a novel modelling technique for prediction and control of lab scale polymerization reaction processes. This model structure uses a modification to the RFP0H model, which uses a first principle with consistent input to make the control aspect linear. This modification made it so that the SID model predicted a scaling factor instead residual error of a process

and allowed for superior predictive and control abilities compared to the older technique. This was demonstrated using a simulated and experimental simulation of a pilot scale polymerization batch reactor. Where the SFFP0H model has a better predictive ability of 98% and minimized control error by 2.5% compared to the RFP0H model. This demonstrates the usefulness of the model structure, and brings more light on making informative feature decisions for hybrid models. Future work should go into analyzing and understanding parallel hybrid design choices for making more informative hybrid models.

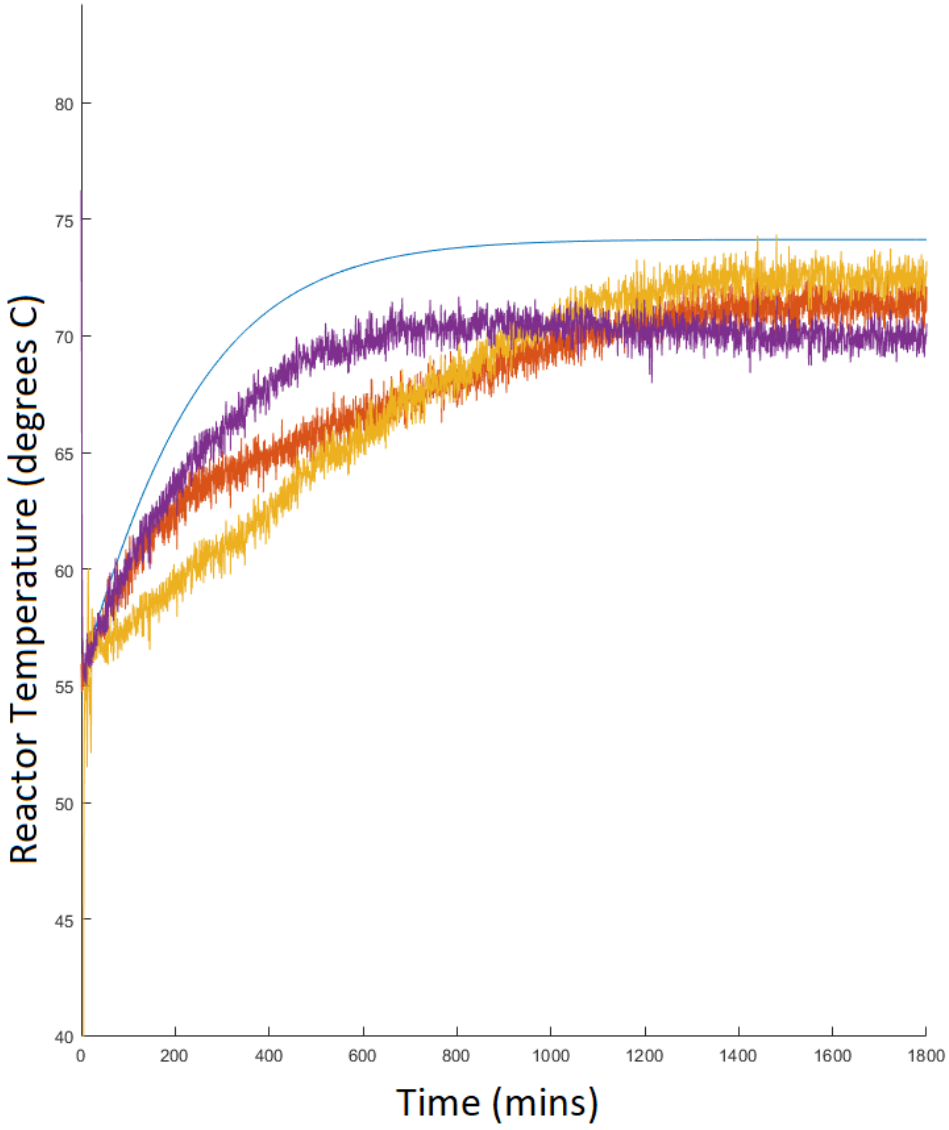

Figure 3.4: Prediction of an exemplar batch. Yellow is the prediction using the SFFP0H model, Purple is the prediction using the RFP0H model, the blue shows the prediction of the FP0 model, and the orange shows the simulated process output.

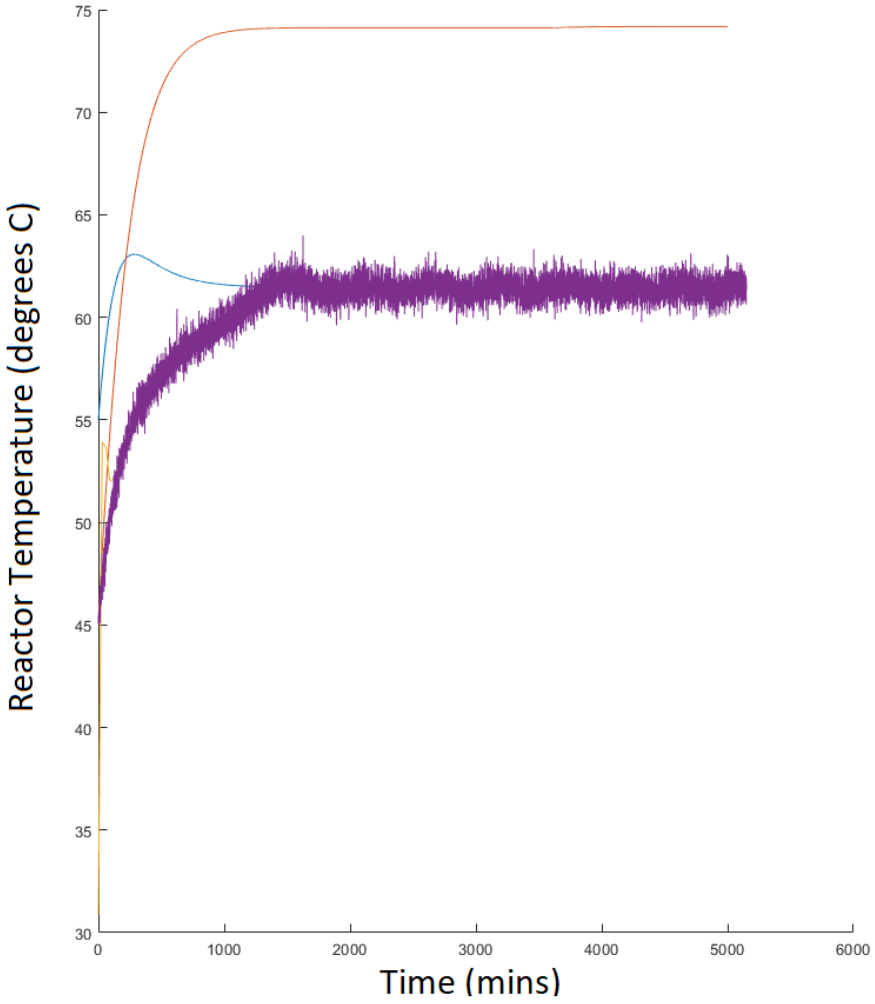

Figure 3.5: Output prediction (yellow), desired (blue), FP0 (red) and actual output of the reactor (purple) temperature using the SFFP0H model.

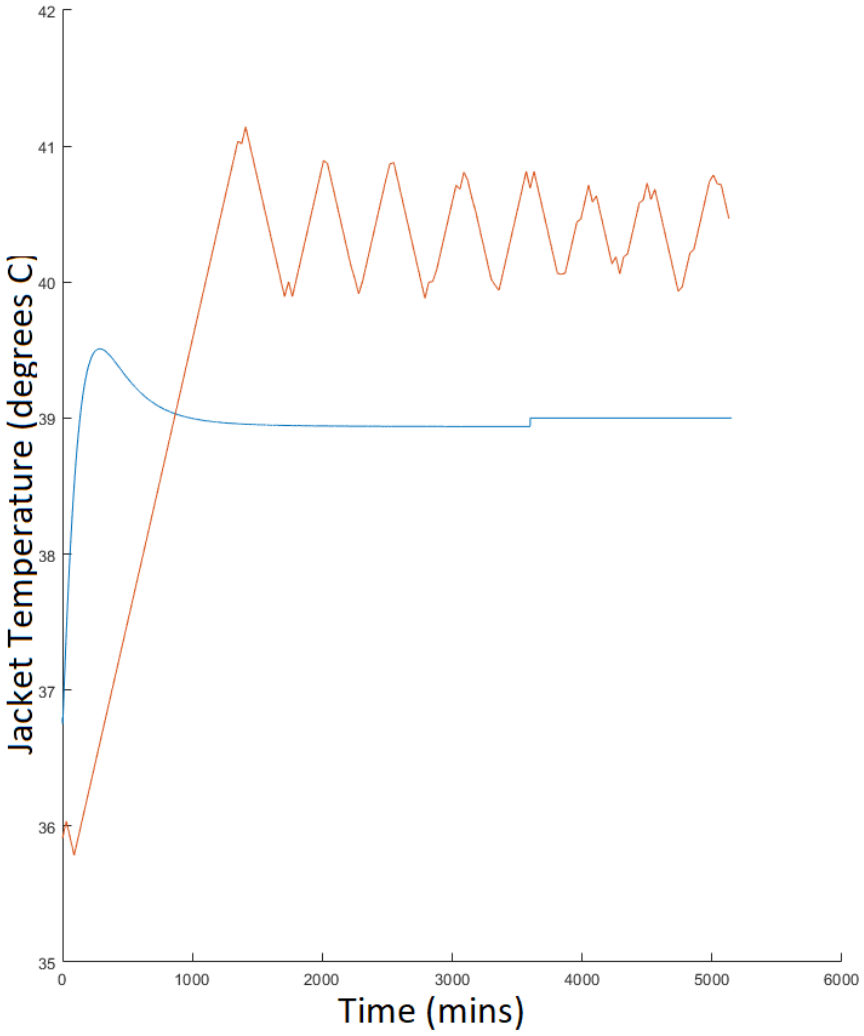

Figure 3.6: Input choices (red) (Jacket temperature) of the MPC and the consistent input profile for the FP0 model (blue). Input choices from MPC using the SFFP0H model.

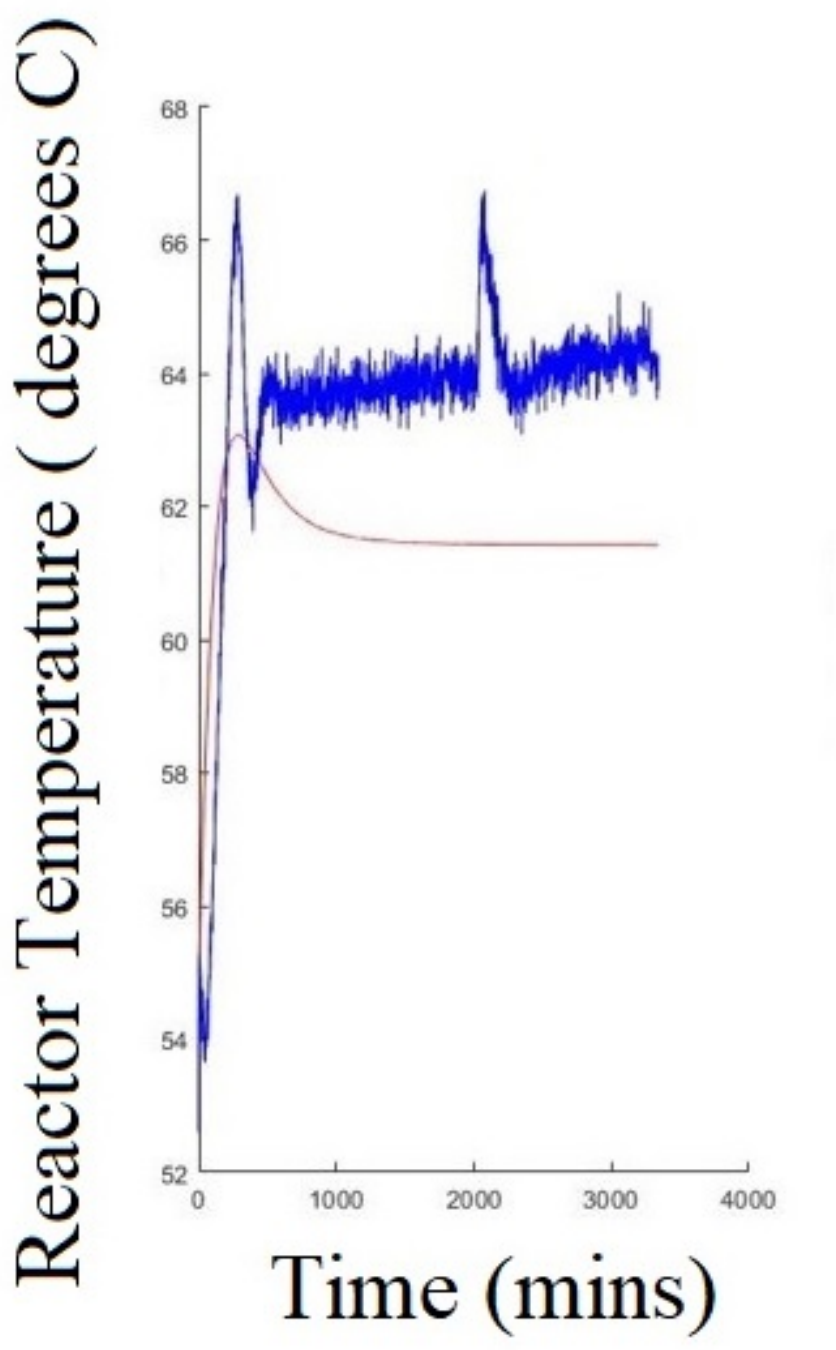

Figure 3.7: Actual, blue, and desired output, red, of the reactor temperature using the RFP0H model.

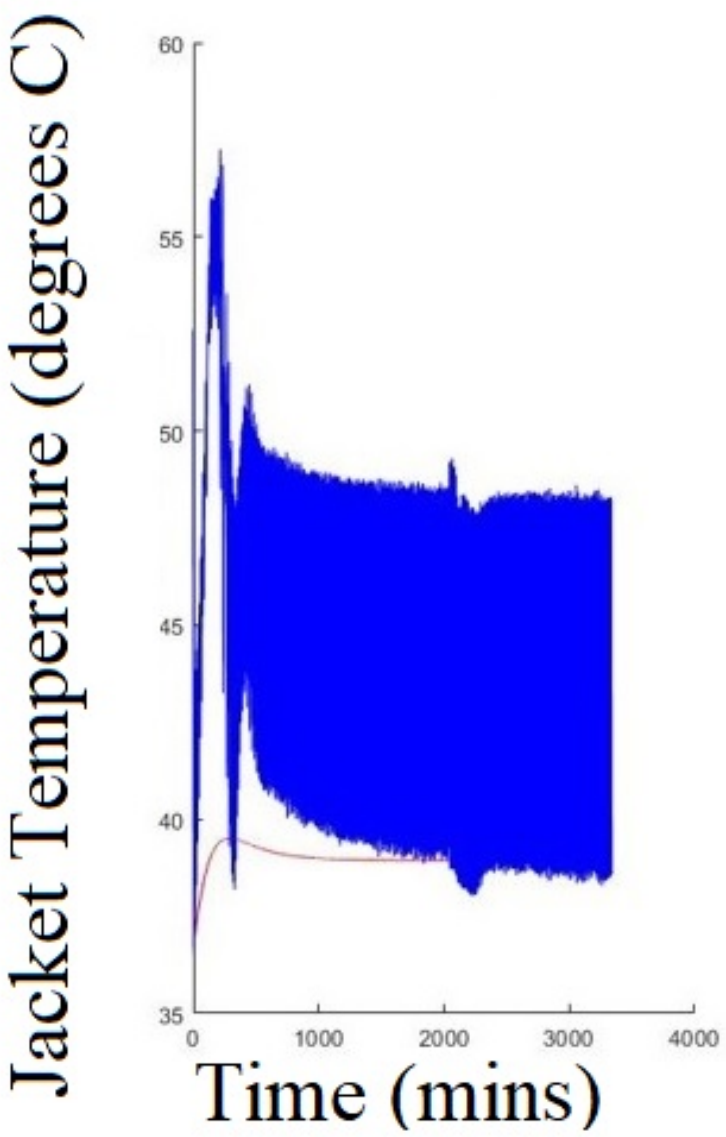

Figure 3.8: Input choices of the MPC, blue, and the consistent input profile for the FP0 model, red. For the real process system Input choices come from MPC using the RFP0H model.

Chapter 4

Conclusions and Recommendations

In this chapter a short summary of findings and recommendations for future research are provided.

4.1 Conclusions

Due to the rapid growth of Model Predictive Control (MPC) within industry due to their versatility and usability has brought out a flurry of literature to create models that accurately represent process system dynamics. There are current limitations for the various model structures within an MPC framework. They also can be intractable making them impossible to optimize. Mechanistic models have high complexity and difficulty to create and optimize, but can provide highly accurate understanding of system dynamics. Linear Data Driven Models have low complexity with ease of creation and optimization, but can be limited in their predictive ability due to their linear structure. Hybrid models attempt to use multiple model structures and have the pluses and minuses of both model structures. Modifications to hybrid model that are both easy for MPC optimization and explain the dynamics accurately are highly desirable in industry. Modifications to the parallel hybrid model structure that provides accurate linear control of the model with an explanation of the non-linear dynamics based on the initial conditions have been shown in an exemplar batch and an exemplar fed-batch settings, providing tools to help deal with the limitations of MPC.

4.1.1 Residual 0 First Principle Hybrid Model

A modification to a parallel hybrid model was made where a first principle with constant inputs is used to explain the overall dynamics of the process system. The residual SID model explains the error that exists between the first principle and the process model where the inputs in the SID are consistent with the realized inputs of the process system. This modifications allows the hybrid model to be non-linear

in prediction but linear in control. As the first principle only needs to be solved once at the start of the process, any desired output values can be subtracted by the predicted value from the FP0 to provide a desired error value for the process. A linear MPC can the be computed to find input values to obtain the desired error of the process. It was shown using a fed-batch crystallization process that the modification to a constant input first principle did not impact the predictive ability of the model and had superior control ability compared to other linear modelling control techniques.

4.1.2 Scaling Factor 0 First Principle Hybrid Model

A modification to the residual 0 first principle hybrid model was made where instead of summing the outputs together, the SID model predicts a scaling factor to multiple by the prediction from the first principle model to predict the process output. This modification maintains the linear aspects of the RFP0H model in control. The non-linear dynamics are still explained using the first principle model, but control only the linear SID model is needed for linear process control. The multiplicative structure allows for the relationship between the first principle and the process output to be modelled more accurately. As the multiplicative relationship better explains the variance between the process output and the first principle model, allowing for better prediction and control of the process system. This was exemplified in a laboratory scaled polymerization batch reactor where the predictions and control were improved over the results from the RFP0H model.

4.2 Future Work

A short listing of future work possibilities that can be continued based of the research from the last two years are listed.

4.2.1 Continued Study of the Lab Scale Polymerization Reactor

The system under study in chapter 3 has provided insightful information into the usefulness of Parallel Hybrid models, and was used to demonstrate the usefulness of the multiplicative hybrid model. The realization of an impactful MPC on the system has not been fully realized on the real process. More model creation and modification, using newly collected process data will be necessary for a fully refined process model that will provide excellent control in the process output.

4.2.2 Meta Analysis of Parallel Hybrid Model Structures

There exists a wide range of both parallel and series hybrid model structures that exist in literature. These model structures extract features from the process data to attempt to explain the variance of a

process system, and are attempted to be used in real time MPC applications. These model structures vary widely, and explain various amount of variances in process and have various abilities to control the process to achieve desired outcomes. The structures of these hybrid models should be studied more, and a greater understanding of why these models function well should also be looked into. Particularly positives and negatives of different structures should be critiqued and viewed through the different dynamics that are being attempted to be explained. Such introspection of the field will help make more informed choices in the proposed model structure formulations for dynamic systems.

4.2.3 Initial State Observer Analysis

There exist the potential of more features within the hybrid model structure to provide more informative models for prediction and control. One short coming of state space models is there need for observers to make accurate predictions. As the states of an SID model are not inherently measurable, they need to be inferred based off the error between model prediction and measured output. The states are modified proportionally to the error using a Luenberger observer, this takes multiple iterations of measured data for the states to converge. This takes away time that the MPC could be running, to provide input suggestions. There is a noticeable relationship between the SID states and the states of the process. A feature that takes advantage of such a correlation, and uses system states to predict the SID states could potentially convergence faster and provide more time for control. This is especially important for batch processes, where a significant portion of the final process characteristics are determined by the initial properties of the batch.

4.2.4 Computers, Society, and Chemical Engineering

Computers over the last seventy years have increasingly come to dominate numerous aspects of human life, society, and industry. In the field of chemical engineering, computers are vitally important for automation of processes, recording of data, and the use of computational power to perform large quantities of calculations in short periods. It also seems that humanity is on the precipice of a fourth industrial revolution, where there is a world of inter-connectivity between domains of study. The problems that chemical engineers now need to solve will increasingly need multiple domains of knowledge to solve. Requiring knowledge from different expertise. Additionally, the problems of MPC and modelling are increasingly routed in data analysis and machine learning, their own domains of knowledge. These domains have much promise in providing information and informative models for better understanding the dynamics of humanity. For a Process System Engineer, it appears vitally important to have a fully rounded education to have a domain of knowledge in computer science and other areas.

Bibliography

[1] Abdul Afram, Farrokh Janabi-Sharifi, Alan S Fung, and Kaamran Raahemifar. Artificial neural network (ann) based model predictive control (mpc) and optimization of hvac systems: A state of the art review and case study of a residential hvac system. *Energy and Buildings*, 141:96–113, 2017.

[2] Paul Serban Agachi, Zoltán K. Nagy, Mircea Vasile Cristea, and Árpád Imre-Lucac. *Model Based Control: Case Studies in Process Engineering.* Wiley-VCH, 2007.

[3] George Alogoskoufis and Ron Smith. On error correction models: specification, interpretation, estimation. *Journal of economic surveys*, 5(1):97–128, 1991.

[4] Stuart Bennett. Development of the pid controller. *IEEE Control Systems Magazine*, 13(6):58–62, 1993.

[5] Benjamin S Blanchard. *System engineering management.* John Wiley & Sons, 2004.

[6] G Buffa, L Donati, L Fratini, and L Tomesani. Solid state bonding in extrusion and fsw: process mechanics and analogies. *Journal of Materials Processing Technology*, 177(1-3):344–347, 2006.

[7] Roland Burns. *Advanced control engineering.* Elsevier, 2001.

[8] Shengze Cai, Zhiping Mao, Zhicheng Wang, Minglang Yin, and George Em Karniadakis. Physics-informed neural networks (pinns) for fluid mechanics: A review. *Acta Mechanica Sinica*, pages 1–12, 2022.

[9] Hon Huin Chin, Petar Sabev Varbanov, Jiři Jaromír Klemeš, Michael Francis D Benjamin, and Raymond R Tan. Asset maintenance optimisation approaches in the chemical and process industries–a review. *Chemical Engineering Research and Design*, 164:162–194, 2020.

[10] Timothy Chiu and Panagiotis D. Christofides. Nonlinear control of particulate processes. *American Institute of Chemical Engineers*, 45(6):1279–1297, 1999.

[11] Helena Chojnacka. Nuptiality patterns in an agrarian society. *Population Studies*, 30(2):203–226, 1976.

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

[13] 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.

[14] Neil Cowie. Observation. In *Qualitative research in applied linguistics*, pages 165–181. Springer, 2009.

[15] Lehel Huba Csekő, Michal Kvasnica, and Béla Lantos. Explicit mpc-based rbf neural network controller design with discrete-time actual kalman filter for semiactive suspension. *IEEE Transactions on Control Systems Technology*, 23(5):1736–1753, 2015.

[16] Lehel Huba Csekő, Michal Kvasnica, and Béla Lantos. Explicit mpc-based rbf neural network controller design with discrete-time actual kalman filter for semiactive suspension. *IEEE Transactions on Control Systems Technology*, 23(5):1736–1753, 2015.

[17] M Darby and M Nikolaou. Identification test design for multivariable model-based control: An industrial perspective. *Control Engineering Practice 2*, 22:165, 2014.

[18] George Edward Davis. *A Handbook of Chemical Engineering: Illustrated with Wroking Examples and Numerous Drawing from Actual Instalations*, volume 2. Davis Bross, 1904.

[19] Kingsley Davis. Human society. 1949.

[20] Henry Winram Dickinson. *A short history of the steam engine*. Cambridge University Press, 2011.

[21] Tom Dietterich. Overfitting and undercomputing in machine learning. *ACM computing surveys (CSUR)*, 27(3):326–327, 1995.

[22] Yining Dong and S. Joe Qin. Dynamic-inner partial least squares for dynamic data modeling. *The International Federation of Automatic Control*, 1:117–122, 2-15.

[23] Keith J Dreyer, DS Hirschhorn, James H Thrall, and M PACS. *A guide to the digital revolution*. Springer, 2006.

[24] George Ellis. *Control System Design Guide: Introduction to Observers in Control Systems*, pages 185–212. BH, 2012.

[25] Martin R Evans and Tom Hanney. Nonequilibrium statistical mechanics of the zero-range process and related models. *Journal of Physics A: Mathematical and General*, 38(19):R195, 2005.

[26] Richard M Felder, Ronald W Rousseau, and Lisa G Bullard. *Elementary principles of chemical processes*. John Wiley & Sons, 2020.

[27] J Flores-Cerrillo and J.F. MacGregor. Control of particle size distributions in emulsion semibatch polymerization using mid-course correction policies. *Industrial Engineering Chemical Research*, 21:1805 1814, 2002.

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

[29] Bjarne A. Foss, Tor A. Johansen, and Aage V. Sorensen. Nonlinear predictive control using local models - applied to a batch fermentation process. *Control Engineering Practice*, 3(3):389–396, 1995.

[30] Philip Hans Franses. A note on the mean absolute scaled error. *International Journal of Forecasting*, 32(1):20–22, 2016.

[31] Carlos Garcia, David PReti, and Manfred Morari. Model predictive control: Theory and practice a survey. *Automatica*, 25(3):335–348, 1989.

[32] 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.

[33] Paul Geladi and Bruce R.Kowalsk. Partial least-squares regression: a tutorial. *Analytica Chimica Acta*, 185:1–17, 1986.

[34] P. Georgieva and S. Feyo de Azevedo. Neural network-based control strategies applied to a fed-batch crystallization process. *International Journal of Computational Intelligence*, 3(3):1304–2386, 2006.

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

[36] Debanjan Ghosh, Jesús Moreira, and Prashant Mhaskar. Model predictive control embedding a parallel hybrid modeling strategy. *Industrial and Engineering Chemistry Research*, 60(6):2547–2562, 2021.

[37] Fausto Giunchiglia and Toby Walsh. A theory of abstraction. *Artificial intelligence*, 57(2-3):323–389, 1992.

[38] Masoud Golshan, John F. MacGregor, Mark-John Bruwer, and Prashant Mhaskar. Latent variable mpc for trajectory tracking in batch processes: Role of the model structure. *Proceedings of the American Control Conference*, 1:4779–4784, 2009.

[39] 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.

[40] Clive WJ Granger and Andrew A Weiss. Time series analysis of error-correction models. In *Studies in econometrics, time series, and multivariate statistics*, pages 255–278. Elsevier, 1983.

[41] Roger Castle Griffin. *Technical Methods of Analysis as Employed in the Laboratories of Arthur D. Little, Inc., Cambridge, Mass.* McGraw-Hill book Company, Incorporated, 1921.

[42] Robert Hecht-Nielsen. Iii.3 - theory of the backpropagation neural network. *Computation, Learning, and Architectures*, 1:65–93, 1992.

[43] Tom Hewitt and David Wield. *Industrialization and development.* Oxford University Press, 1992.

[44] David Mautner Himmelblau and James B Riggs. *Basic principles and calculations in chemical engineering.* FT press, 2012.

[45] KS Holkar and LM Waghmare. An overview of model predictive control. *International Journal of control and automation*, 3(4):47–63, 2010.

[46] Bao-Gang Hu, George KI Mann, and Raymond G Gosine. A systematic study of fuzzy pid controllers-function-based evaluation approach. *IEEE transactions on fuzzy systems*, 9(5):699–712, 2001.

[47] Francis J. Doyle III, Christopher A. Harrison, and Timothy J.Crowley. Hybrid model-based approach to batch-to-batch control of particle size distribution in emulsion polymerization. *International Journal of Applied Mathematics and Computer Science*, 27(8-9):1153–1163, 2003.

[48] Robert B Jackson, Corinne Le Quéré, RM Andrew, Josep G Canadell, Jan Ivar Korsbakken, Zhu Liu, Glen P Peters, and Bo Zheng. Global energy growth is outpacing decarbonization. *Environmental Research Letters*, 13(12):120401, 2018.

[49] S.K. Jha, S. Karthika, and T.K. Radhakrishnan. Modelling and control of crystallization process. *Resource-Efficient Technologies*, 3(1):94–100, 2017.

[50] Landon Johnson. What is a system? *Student Works*, 2021.

[51] R Kadali, B Huang, and A Rossiter. A data driven subspace approach to predictive controller design. *Control Engineering Practice*, 11:261–278, 2003.

[52] Richard Shelton Kirby. *Engineering in history.* Courier Corporation, 1990.

[53] Donald E. Knuth. *Industrial Crystallization: The population balance equation*, chapter 7. Cambridge University Press, 2015.

[54] Thomas Kuhn. The history of science. In *Philosophy, Science, and History*, pages 106–121. Routledge, 2014.

[55] J. Nathan Kutz. *Data-Driven Modeling and Scientific Computation: Methods for Complex Systems and Big Data*. Oxford, 1 edition, 2013.

[56] JS-I Kwon, M Nayhouse, G Orkoulas, D Ni, and PD Christofides. Run-to-run-based model predictive control of protein crystal shape in batch crystallization. *Industrial Engineering Chemical Research*, 54:4293–4302, 2015.

[57] Nicolas Lanzetti, Ying Zhao Lian, Andrea Cortinovis, Luis Dominguez, Mehmet Mercangöz, and Colin Jones. Recurrent neural network based mpc for process industries. In *2019 18th European Control Conference (ECC)*, pages 1005–1010. IEEE, 2019.

[58] Heiner Lasi, Peter Fettke, Hans-Georg Kemper, Thomas Feld, and Michael Hoffmann. Industry 4.0. *Business & information systems engineering*, 6(4):239–242, 2014.

[59] Heiner Lasi, Peter Fettke, Hans-Georg Kemper, Thomas Feld, and Michael Hoffmann. Industry 4.0. *Business & information systems engineering*, 6(4):239–242, 2014.

[60] Steve Lawrence, C. Lee Giles, and Ah Chung Tsoi. Lessons in neural network training: Overfitting may be harder than expected. *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, 1:540–545, 1997.

[61] João R. Leal, Andrey Romanenko, and Lino O. Santos. Daedalus modeling framework: Building first-principle dynamic models. *Industrial and Engineering Chemistry Research*, 56(12):3332–3346, 2017.

[62] Shosh Leshem and Vernon Trafford. Overlooking the conceptual framework. *Innovations in education and Teaching International*, 44(1):93–105, 2007.

[63] Gang Li, Baosheng Liu, S. Joe Qin, and Donghua Zhou. Quality relevant data-driven modeling and monitoring of multivariate dynamic processes: The dynamic t-pls approach. *IEEE Transactions on Neural Networks*, 22(12):2262 – 2271, 2011.

[64] Alexander McKay, Debanjan Ghosh, Lu Zhu, Li Xi, and Prashant Mhaskar. A novel linear hybrid model predictive control design: Application to a fed batch crystallization process. *Digital Chemical Engineering*, page 100033, 2022.

[65] Hourshad Mohammadi. *Mechanistic modeling, design, and optimization of alkaline/surfactant/polymer flooding*. The university of Texas at Austin, 2008.

[66] Marc Moonen, Bart De Moor, Lieven Vandenberghe, and Joos Vandewalle. On- and off-line identification of linear state space models. *International Journal of Control*, 49:219–232, 1989.

[67] E Morgenroth and PA Wilderer. Sequencing batch reactor technology: concepts, design and experiences (abridged). *Water and Environment Journal*, 12(5):314–320, 1998.

[68] A Narasingam and JS-I 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.

[69] Michael Nikolaou. Model predictive controllers: A critical synthesis of theory and industrial needs. *Advances in Chemical Engineering*, 2001.

[70] William G Olchi. The transmission of control through organizational hierarchy. *Academy of management Journal*, 21(2):173–192, 1978.

[71] R. Oliveira. Combining first principles modelling and artificial neural networks: a general framework. *Computers Chemical Engineering*, 28(5):755–766, 2004. ESCAPE 13.

[72] Ronald Peeters, Luciano Simone, Koen Nelissen, Maddalena Fabbri-Destro, Wim Vanduffel, Giacomo Rizzolatti, and Guy A Orban. The representation of tool use in humans and monkeys: common and uniquely human features. *Journal of Neuroscience*, 29(37):11523–11539, 2009.

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

[74] S. Joe Qin and Thomas A. Badgwell. An overview of nonlinear model predictive control applications. *Nonlinear Model Predictive Control*, 1:369–392, 2000.

[75] Anne Raich and Ali Cinar. Statistical process monitoring and disturbance diagnosis in multivariable continuous processes. *AIChE Journal*, 42(4):995–1009, 1996.

[76] Jacques Richalet, André Rault, JL Testud, and J Papon. Model predictive heuristic control: Applications to industrial processes. *Automatica*, 14(5):413–428, 1978.

[77] Colette Rolland. A comprehensive view of process engineering. In *International Conference on Advanced Information Systems Engineering*, pages 1–24. Springer, 1998.

[78] Lino O. Santos, Paulo A.F.N.A. Afonso, Jose A.A.M. Castro, , Nuno M.C. Oliveira, and Lorenz T. Bieglerb. On-line implementation of nonlinear mpc: an experimental case study. *Control Engineering Practice*, 9:847–857, 2001.

[79] MP Sealy and YB Guo. Surface integrity and process mechanics of laser shock peening of novel biodegradable magnesium–calcium (mg–ca) alloy. *Journal of the mechanical behavior of biomedical materials*, 3(7):488–496, 2010.

[80] John R Searle, y S Willis, et al. *The construction of social reality*. Simon and Schuster, 1995.

[81] 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.

[82] 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, 2007.

[83] Rens van de Schoot, Sarah Depaoli, Ruth King, Bianca Kramer, Kaspar Märtens, Mahlet G Tadesse, Marina Vannucci, Andrew Gelman, Duco Veen, Joukje Willemsen, et al. Bayesian statistics and modelling. *Nature Reviews Methods Primers*, 1(1):1–26, 2021.

[84] 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(10):86–101, 2014.

[85] Grady Williams, Nolan Wagener, Brian Goldfain, Paul Drews, James M. Rehg, Byron Boots, and Evangelos A. Theodorou. Information theoretic mpc for model-based reinforcement learning. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 1:1714–1721, 2017.

[86] Jinzhong Zhang and Robin Smith. Design and optimisation of batch and semi-batch reactors. *Chemical Engineering Science*, 59(2):459–478, 2004.

[87] Maciej Ławryńczuk. A family of model predictive control algorithms with artificial neural networks. *International Journal of Applied Mathematics and Computer Science*, 17(2):217–232, 2007.