Multi-Phase Batch SSID With Applications to Rotomoulding

Multi-Phase Subspace Identification Formulations for Batch Processes With Applications to Rotational Moulding

by

Evan Ubene, B.Eng.Mgmt

A Thesis
Submitted to the School of Graduate Studies
in Partial Fulfillment of the Requirements for
the Degree Master's of Science

Master's of Science (2023)	McMaster University
(Chemical Engineering)	Hamilton, Ontario, Canada

TITLE:	Multi-Phase Subspace Identification Formulations for Batch Processes With Applications to Rotational Moulding
AUTHOR:	Evan Ubene, B.Eng.Mgmt (McMaster University, Hamilton, ON)
SUPERVISOR:	Dr. Prashant Mhaskar
NUMBER OF PAGES:	xiv, 94

LAY ABSTRACT

The control of chemical processes is an important factor in achieving high quality products. To control a process well, the mathematical model of the system must be accurate. In the past, mathematical models for process control were designed based on engineering approximations. Now, with major advances in computing and sensor technology, it is possible to design a simulation of the entire process. These simulations can be designed using first-principles or black box approaches. First-principles approaches utilize rigorous models that are based on the complex chemical and physical formulas that govern a system. Black box approaches do not look at the first-principles dynamics. They only utilize the measured process inputs and outputs to form a model of the system. They are widely used because of their ease of implementation in comparison to first-principles approaches. In this thesis, a new black box process control model is proposed and is found to yield better theoretical results than existing techniques. This model is tested on data from a plastics manufacturing process called rotational moulding, which involves loading polymer powders into a mould that is simultaneously rotated and heated to yield seamless plastic parts. Lastly, a control framework that is compatible with the new black box model is proposed to be used for future experimental tests.

ABSTRACT

This thesis focuses on the implementation of subspace identification (SSID) for nonlinear, chemical batch processes by introducing a model identification method for multi-phase processes. In this thesis, a multi-phase process refers to chemical or biological batch-like processes with properties that cause a change in the dynamics during the evolution of the process. This can occur, for example, when a process undergoes a change of state upon reaching a melting point. Existing SSID techniques are not designed to utilize any known, multiphase nature of a process in the model identification stage. The proposed approach, Multiphase Subspace Identification (MPSSID), is conducted by first splitting historical data into phases during the identification step and then building a subspace model for each phase. The phases are then connected via a partial least squares (PLS) model that transforms the states from one phase to the next. This approach makes use of existing SSID techniques that allow for model construction using batches of nonunifrom length. Here, MPSSID is applied to a uniaxial rotational moulding process. In rotational moulding, the dynamics switch as the process undergoes heating, melting, and sintering stages that are visibly distinct and recognizable upon a certain temperature (not time) being reached. Results demonstrate the ability of multiphase models to better predict the temperature trajectories and final product quality of validation batches. As an extension to this rotational moulding analysis, additional MPSSID methods of implementation are proposed and the results are compared. A MPSSID mixed integer linear program is then introduced for implementation within model predictive control. The applications to rotational moulding are presented within the context of plastics manufacturing and the impact of plastic on the global climate crisis, with suggestions for future work.

ACKNOWLEDGEMENTS

There are a few people I would like to acknowledge. First I would like to thank Dr. Mhaskar for having more patience than the Guns N' Roses song (the Chris Cornell cover is better). Without his considerate approach to graduate studies and the students who work with him, I would not have finished.

I would like to thank my brother for being someone who I can constantly talk to, and my parents who in many ways do not understand my motivations in life but provide support when they can anyway. I would like to thank my Uncle Mike and late Auntie Pat for supporting me as well.

I would like to thank the NHL 2005 soundtrack for sounding just as fire as it did 18 years ago.

I would like to thank masks for reducing the number of times I have gotten sick since the start of the pandemic. Far fewer people would catch COVID and other illnesses if more folks wore masks and institutions reintroduced masking protections.

Lastly, I would like to thank everyone in the world who constantly takes the time to work towards political initiatives that will produce positive social change, especially the individuals I have had the pleasure of working with in Hamilton on what has been too many projects to count. We are currently going through intersecting existential crises related to the climate and public health, and in both cases those with the power to act have abdicated their responsibility. Organizing in these conditions requires a constant belief in the good nature of people and that things will get better if enough people choose to fight for it. It's not always easy, but it's worth it. Nothing good for working people was ever won without a struggle.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The control landscape in chemical process control is increasingly dominated by advances in computing and sensor technologies that enable real-time tracking and computational control of processes. Whereas traditional control was dominated by the use of PID and first principles models, the often vast amount of historical data that operators can now collect has increased the use of black-box models that only require knowledge about the inputs and outputs of a process rather than its inner-workings. These models are especially useful for batch processes. Batch processes often require offline measurements to determine the final product quality, and so these values are not measured during the process operation. The focus of previous batch control methods was on trajectory tracking approaches where the control program is set to follow an operating path based on past experimental success, but these approaches only control for variation in the output variables and so do not handle disturbances well. Black box models, such as subspace identification, can utilize the inputs and outputs of a process to calculate a state estimate that is updated based on the process dynamics, rather than variation in the output from pre-defined trajectories. These models allow for direct quality control because the final product quality can be attributed to the final process states through additional model relationships. The model structure of subspace identification also makes it easy to implement within model predictive control (MPC). The limitations of subspace identification include its over-reliance on measured variables, which may not always accurately describe the process, and its linear time invariant structure, which makes it difficult to approximate nonlinear systems. These limitations provide the motivation for this thesis. New approaches to subspace identification are presented to account for the nonlinearity of multi-phase systems and tested on a lab-scale rotational moulding process. Rotational moulding is a plastics manufacturing process where a polymer powder is loaded into a mould before it is simultaneously rotated and heated to form a seamless plastic part. This

method is commonly used in the production of high-quality plastic parts such as construction materials and storage containers. Unfortunately, plastic waste is a significant contributor to the ecological crisis facing the world in 2023, and so additional engineering investments into this sector must be carefully considered.

The remainder of this section first provides added detail to contextualize plastics manufacturing in the present day before presenting a brief outline of each of the chapters and the contributions within.

1.2 Plastics Manufacturing in the Present Day

The OECD 2019 Global Plastics Outlook reported that global plastics production doubled from 2000 to 2019 to reach 460 million tonnes (Mt) [1]. During the same time period, plastic waste generation *more* than doubled, reaching 353 Mt. The report also found that in 2019, 22% of this waste was mismanaged, with an estimated 6.1 Mt leaking into aquatic environments. This is the equivalent weight of 61 cargo ships. Other evaluations put this number closer to 15 Mt (150 ships), with estimates that it will reach levels between 23 and 37 Mt by 2040 under "business as usual" scenarios [2]. Plastics also accounted for 3.4% of global greenhouse gas emissions as of 2019.

The abundance of plastic in the environment is now a cause for serious concern to biological life. Studies based on environmentally-relevant concentrations of microplastic exposure have quantified negative impacts to human cells in laboratory settings [3]. Microplastics have been found in human tissues such as the lungs [4] and blood [5]. There are an extensive number of studies discussing the negative impacts to aquatic life that result from entanglement and ingestion of plastics [6]. The finding from the OECD report that plastic waste is increasing at a higher rate than overall

plastic production indicates that there is an immediate need to enforce waste management policies and develop technological applications to mitigate the harmful impact of plastic waste that already exists in the natural environment and dramatically slow its accumulation.

In [7], potential technological strategies to obtain a microplastics-free environment are reviewed. One notable finding in this study is that research has shown a link between plastic waste in landfills and microplastics in the surrounding environment, showing that mechanical waste disposal of plastics is not good enough. One Canadian study recently showed how plastics recycling can *increase* the amount of microplastics entering the environment due to contamination of the water supply [8]. In [7], the authors suggest a combination of mechanical and chemical approaches to plastics waste management. They rank membrane approaches as the best mechanical method, and rank photocatalysis as the best method for chemical degradation. Strategies must also be developed for use in existing environmental reservoirs of plastic, since the slow degredation that occurs in these locations will add to the microplastic probelm if it is not addressed. One process control application to help address the plastic waste problem is to explore control improvements of solvolysis reactions, which chemically break down plastics into their original form for recycling as a way to avoid the degradation problems of mechanical recycling. Additional solvolysis reaction control research is needed due to the uneven temperature distribution in reactors that is caused by the low thermoconductivity of plastic, and uncontrollable reactions that can occur from plastics that undergo side-degradation reactions [7].

Other suggestions for plastic waste in [7] include using waste products to produce energy, converting plastics to carbon materials, or replacing plastics with biodegradable polymers. Producing energy from plastic would entrench greenhouse gas production from inefficient fuel sources. Also, environmental impact assessments on the efficiency of using waste plastic to create carbon materials compared to other feed stocks

have not been done yet[9]. Lastly, studies have concluded that biodegradable alternatives can still contribute to the microplastic problem, especially under poor regulatory frameworks [10]. In [10], the authors conclude that biodegradable products must meet strict definitions and should probably only be used for products where degradation is part of the lifecycle (such as single-use products).

In Canada, by federal government's own admission, their current plastics management policy does not consider consumption, reduction, or reuse approaches and is only projected to reduce plastic waste by 3% by 2030 [11]. That's because the policy focuses on recycling approaches and bans of a select few products instead –the federal government recently banned six single-use plastic items: checkout bags, cutlery, foodservice ware, ring carriers, stir sticks, and straws [12]. A report commissioned by the federal government indicated that only 8% of plastic waste that was generated in Canada in 2016 was recycled [13]. Only 33% of plastic that was sent to sorting facilities was recycled. Additionally, nearly half of all plastic waste is packaging. This highlights how a focus on recycling will only solidify the production processes that have caused the problem. Greenpeace Canada estimates that Canada's plastic recycling capabilities in 2016 only amounted to a maximum of 17% of the total waste produced [14], but even if it was much larger, recycling processes (especially the more popular mechanical kind) still require varying proportions of virgin material to meet product quality specifications. A focus on recycling will maintain the use of fossil fuels as a raw material at a time where emissions must be dramatically reduced. Rather than giving the companies at fault billions in government subsidies to build more facilities so that they can continue to profit from environmental pollution, efforts should be spent reducing the waste being produced. Policies that standardize production sizes and make the producer responsible for waste have already been proven in Ontario, such as the bottle and can return system at the Beer Store. An approach that focuses on reduction of waste and reuse of material instead of recycling also aligns with waste management hierarchies.

Lastly, most plastic is made from fossil fuels, which gets shipped through oil pipelines across the country to petrochemical refineries. These pipelines cut through Indigenous territory and create significant risks to the surrounding ecosystem from potential leakages. The Line 5 Pipeline that runs from Superior, Wisconsin to Sarnia, Ontario is a prime example, as it is now more than 70 years old and goes through important watersheds [15]. Researchers must avoid getting caught in areas of innovation that give companies an excuse to generate more pollution and make sure any work they do around plastics manufacturing innovates in areas that meaningfully reduce waste. They should be on the front lines pressuring governments to implement strict regulation of the industry and policies that benefit the public good rather than policies that allow companies to generate ever-increasing profits. A failure to do so will see plastic pollution continue to proliferate.

Towards the end of section 1.2, the rotomoulding plastics manufacturing process to which the theoretical ideas contained within this thesis are applied is briefly mentioned. How does rotomoulding fit into the current plastics landscape? Rotomoulding is commonly used to produce long-lasting plastic parts rather than the single-use types that are used for packaging, and so a key consideration for researchers should be to make sure products are paired with comprehensive waste management protocols that contain technically sound specifications for how products will be collected at the end of their lifetime and disposed of in ways that ensure toxic chemicals do not enter the natural environment. Researchers should constantly ask themselves if the product they are helping produce can be replaced by a simple societal change, such as how plastic bags have been replaced by reusable grocery bags, or how plastic water bottles can be replaced by water stations and reusable bottles. Finally, for products that are necessary and meet the waste management criteria, recycling is another area to be explored from both the mechanical and chemical perspective.

1.3 Outline of the thesis

The purpose of Chapter 2 is to introduce a novel subspace identification approach to improve the control of multi-phase processes. Computational enhancements have driven a shift from first-principles and PID techniques to black-box type models for chemical process control that make use of the vast amount of process data that can now be collected. Subspace identification is a popular black box model that is valued for its compatibility with model predictive control. This work discusses the varying approaches to batch process control, explains the structure of the proposed model, and lastly shows the benefits of the proposed model in comparison to traditional subspace identification approaches.

In Chapter 3, different formulations of the multi-phase subspace identification approach that is introduced in Chapter 2 are explored and compared. The multi-phase model is also compared to a subspace "reidentification" method that has been implemented experimentally with promising results. Finally, a model predictive control (MPC) optimization program is presented for use with Multi-Phase Subspace Identification models.

The last chapter is used to summarize the modeling approaches that are presented in this thesis and the results that were obtained. Future work is also proposed.

Bibliography

[1] OECD, "Global plastics outlook economic drivers, environmental impacts and policy options," *OECD*, 2019.

[2] W. W. Lau, Y. Shiran, R. M. Bailey, E. Cook, M. R. Stuchtey, J. Koskella, C. A. Velis, L. Godfrey, J. Boucher, M. B. Murphy, *et al.*, "Evaluating scenarios toward zero plastic pollution," *Science*, vol. 369, no. 6510, pp. 1455–1461, 2020.

[3] H. A. Leslie, M. J. Van Velzen, S. H. Brandsma, A. D. Vethaak, J. J. Garcia-Vallejo, and M. H. Lamoree, "Discovery and quantification of plastic particle pollution in human blood," *Environment international*, vol. 163, p. 107199, 2022.

[4] L. C. Jenner, J. M. Rotchell, R. T. Bennett, M. Cowen, V. Tentzeris, and L. R. Sadofsky, "Detection of microplastics in human lung tissue using μftir spectroscopy," *Science of The Total Environment*, vol. 831, p. 154907, 2022.

[5] E. Danopoulos, M. Twiddy, R. West, and J. M. Rotchell, "A rapid review and meta-regression analyses of the toxicological impacts of microplastic exposure in human cells," *Journal of Hazardous Materials*, vol. 427, p. 127861, 2022.

[6] S. C. Gall and R. C. Thompson, "The impact of debris on marine life," *Marine pollution bulletin*, vol. 92, no. 1-2, pp. 170–179, 2015.

[7] J. Chen, J. Wu, P. C. Sherrell, J. Chen, H. Wang, W.-x. Zhang, and J. Yang, "How to build a microplastics-free environment: strategies for microplastics degradation and plastics recycling," *Advanced Science*, vol. 9, no. 6, p. 2103764, 2022.

[8] E. Brown, A. MacDonald, S. Allen, and D. Allen, "The potential for a plastic recycling facility to release microplastic pollution and possible filtration remediation effectiveness," *Journal of Hazardous Materials Advances*, vol. 10, p. 100309, 2023.

[9] O. Vieira, R. S. Ribeiro, J. L. D. de Tuesta, H. T. Gomes, and A. M. Silva, "A systematic literature review on the conversion of plastic wastes into valuable 2d graphene-based materials," *Chemical Engineering Journal*, vol. 428, p. 131399, 2022.

[10] T. P. Haider, C. Völker, J. Kramm, K. Landfester, and F. R. Wurm, "Plastics of the future? the impact of biodegradable polymers on the environment and on society," *Angewandte Chemie International Edition*, vol. 58, no. 1, pp. 50–62, 2019.

[11] "Proposed single-use plastics prohibition regulations – summary of public comments," *Environment and Climate Change Canada*, 2023. https://www.canada.ca/en/environment-climate-change/services/managing-reducing-waste/reduce-plastic-waste/single-use-plastics-comments-response.html.

[12] "Single-use plastics prohibition regulations -overview," *Environment and Climate Change Canada*, 2023. https://www.canada.ca/en/environment-climate-change/services/managing-reducing-waste/reduce-plastic-waste/single-use-plastic-overview.html.

[13] "Economic study of the canadian plastic industry, markets and waste," *Environment and Climate Change Canada*, 2019.

[14] L. Yates, S. King, J. Firempong, A. Speers-Roesch, and I. Roy, "Plastic recycling: That's not a thing," *Greenpeace*, 2020.

[15] M. Woodhouse, K. Brooks, and L. Thomas, "Closing enbridge's line 5 pipeline: What are the options and alternatives available?," *Environmental Defense*, 2021.

Chapter 2

Data-Driven Modelling for Multi-Phase Processes: Application to a Rotomoulding Process

This chapter presents a novel subspace model identification approach to improve the control of multi-phase processes. A "multi-phase process" refers to chemical or biological batch-like processes with properties that cause a change in the dynamics during the evolution of the process. Existing subspace identification (SSID) methods are limited in their nonlinear process modelling capabilities, and so do not do a good job approximating multi-phase processes. Multi-Phase Subspace Identification (MPSSID) improves on existing techniques by identifying SSID models for each phase and connecting the phases with partial least squares models. An additional strength of SSID techniques lies in their ability to easily predict the final product quality of a process, which is facilitated by making use of the states calculated by the subspace model. The key idea in this work is to present the MPSSID approach and demonstrate its superior predictive ability over traditional, one-phase SSID models.

Ubene, Evan, and Prashant Mhaskar. "Data-Driven Modeling for Multiphase Processes: Application to a Rotomolding Process." *Industrial & Engineering Chemistry Research*, 62.18 (2023): 7058-7071.

2.1 Abstract

This paper addresses the problem of capturing the multi-phase nature of a rotational moulding process using subspace identification (SSID) to enable improved control. Existing SSID techniques are not designed to utilize any known, multi-phase nature of a process in the model identification stage. This work adapts existing SSID methods to account for multiple phases by splitting the data into phases during the identification step and building a distinct SSID model for each phase, while carefully connecting the individual models through the means of subspace states. This is achieved via a partial least squares (PLS) model that relates the final states of the preceding phase to the initial states of the proceeding phase. This **M**ulti-**P**hase **S**ubspace **ID**entification (MPSSID) approach exploits the ability of SSID techniques for dynamic modelling of batch processes, which allows for model construction using batches of non-uniform length. In this work the proposed approach is applied to the rotational moulding process. For rotational moulding, the final product quality is dependent on the temperature trajectory of the polymer inside the mould and the process goes through visibly distinct phases that can be recognized when a specific temperature (not time) is reached. Data from past experiments is used to build the model and validate it, comparing the predictive ability of multi-phase models to conventional one-phase models. Results demonstrate the ability of the multi-phase models to better predict both the temperature trajectories and final product quality of validation batches.

2.2 Introduction

Batch processes have become common in various chemical industries such as pharmaceuticals, biotechnology, specialty chemicals, agriculture, and microelectronics [1]. They are especially suited for low-volume processes where the final product quality is

important, creating a need for the development of modelling and control approaches that are able to deliver on-spec product [1]. The trouble with batch processes is that they are transient. Nonlinear dynamics observed during batch process runs lead to challenges with batch process control [2, 3, 4]. Traditionally, PID controllers have been used in batch processes to track a pre-defined trajectory of a process variable. To deal with the nonlinearity, gain scheduling is used where the PID tuning parameters vary as the batch progresses [5].

Good mechanistic models, where available, are best suited for batch control [6, 7, 8, 9, 10, 11, 12, 13, 5, 14, 15]. In general, developing and maintaining good mechanistic models is not an easy task. One example of a process that demonstrates this difficulty is rotational moulding (rotomoulding). For the case of the rotomoulding process, the mechanistic model must account for heat transfer, bubble formation, bubble evaporation, polymer degradation, dimensional stability, and polymer rheology [16]. It needs to account for the transitions that happen when the polymer begins to melt, sinter, and cool to form the mould. Most importantly, the mechanistic model must be able to predict product qualities such as impact strength and sinkhole area to meet product specifications. Such models currently do not exist for the rotational moulding process in particular, and for several process in general, motivating the need for data driven models.

More recently, data-driven methods that are able to utilize the often vast record of historical batches have been shown to perform well for the trajectory tracking approach. This includes linear ARX models [4] and neural networks [17, 18]. One of the most well-researched methods is latent variable modelling techniques such as Principal Component Analysis (PCA) and Partial Least Squares/Projection to Latent Spaces (PLS) [19, 20, 21, 22, 23]. A good review of these can be found in Ref. 24. A significant advantage of latent variable methods is that they can be integrated into model predictive control (MPC). Latent variable model predictive control (LV-MPC)

is introduced in Ref. 25 and explored further in Refs. 26 ,27, and 28.

The issue with all trajectory tracking approaches is that they only consider variations in the output variables and not the underlying states. Thus, this method of control may not be able to handle disturbances or deviations in the process feed very well, primarily with respect to the objective of quality control. To address this, PLS-based latent variable approaches have been used to relate the batch trajectory to the product quality using scores[20, 29, 30, 31]. An issue with PLS methods is that they are time indexed; they require the designer to align batches so that they are all the same length. Aligning batches is usually done using an indexing variable, such as conversion. In cases where indexing variable measurements are not available, dynamic time warping can be used [32]. The other challenge is that the quality models of the PLS-based approaches are constructed by grouping the inputs and outputs together. Because these approaches fail to intrinsically distinguish between process inputs and outputs, the ubiquity and validity of the applicable MPC control schemes are limited.

Conversely, subspace-based identification approaches identify a Linear Time-Invariant (LTI) state-space model [33, 34, 35, 36, 37]. The model states obtained form state-space techniques provide an easy method to relate the process trajectory to the product quality and implement direct quality control. This approach was adapted for batch processes [38] and was further adapted to handle discrete inputs while handling the quality control problem that arises for batches of variable lengths [39]. While it was shown in [38] that Batch SSID can outperform PLS and PI methods for batch process feedback control systems, there is potential to improve Batch SSID by modifying the identification process to address process nonlinearity. Piecewise linear systems can be used as approximations of nonlinear systems, and there is a vast amount of work that has been done to develop SSID algorithms for piecewise linear systems that is summarized in Ref. 40. These systems "switch" their modelling parameters for each phase of the batch. Many of these approaches depend on algebraic switching detection

mechanisms that make use of the subspace states, which is more relevant to signal processing, and none of the approaches are readily applicable for batch processes. Furthermore, the existing approaches often consider situations where the switching is time dependent, while the focus in the present work is on switching that is state or process output dependent. In this paper, the terms "multi-phase processes," or processes that exhibit a "multi-phase nature" are used to refer to chemical and biological batch-like processes with inherent properties that cause a significant shift in dynamics during the evolution of the process. Note that in the context of these processes, dynamic shifts often occur under specific, measurable conditions, and so it is much more consistent with the process dynamics to be switching when, for example, a temperature reaches the melting point. Most importantly, in contrast to existing results, the approach presented in this manuscript allows for eventual incorporation of the model in an MPC framework (the demonstration remains outside the scope of the present work).

The current work draws on knowledge from existing SSID methods, specifically those described in Refs. 35, 38 and 39 to develop a novel **M**ulti-**P**hase **S**ubspace **ID**entification (MPSSID) method for batch processes. The key idea is to build subspace models for individual phases that are identified by process knowledge and then appropriately "connect" them to handle the multi-phase nature of the process. The remainder of this work is organized as follows: in Section 2.3, the rotomoulding process is described in detail and used to illustrate existing modelling techniques. Also, the mathematical background and theory required for an understanding of the ensuing sections is presented. In Section 2.4, the proposed MPSSID model is described. In Section 2.5, the proposed MPSSID model is applied to the rotomoulding process and results are discussed. Finally, conclusions are summarized in Section 2.6.

2.3 Preliminaries

In this section, we first describe the rotomoulding process and then give an overview of existing batch SSID approaches.

2.3.1 Rotomoulding Process Description

The rotomoulding process is a plastics manufacturing method that works by heating polymer-filled moulds (usually in an oven) while rotating them [16]. This method of production evenly distributes the polymer throughout the walls of the mould and creates seamless plastic products. Key advantages of rotomoulding over other plastics manufacturing processes like injection moulding include superior product quality, greater mould shape flexibility, and a reduced amount of waste (some plastic is wasted for injection moulding on the sprue and runners that result from where the injection takes place, whereas in rotomoulding all of the material is used to make the product) [16, 41]. These advantages make rotomoulding desirable for low-volume processes where a high-quality specification must be met. The main disadvantages of rotomoulding are that the process duration is longer than it is for injection moulding, the range of suitable polymers is limited, and failed parts can be wasteful since rotomoulding is more commonly used for custom parts that may be larger in size [16, 41]. To diminish the impact of a longer run time and reduce the amount of waste created from failed parts, an accurate control scheme is necessary to ensure product quality is met consistently.

The rotomoulding process goes through a sequence of distinct phases while the polymer is heating and cooling as described in Ref. 42 and depicted in figure 2.1. Initially, the polymer is loaded into the mould in powder form. During the adhesion and melting phases, the powder is heated until it begins to melt. Next, gas bubbles within the

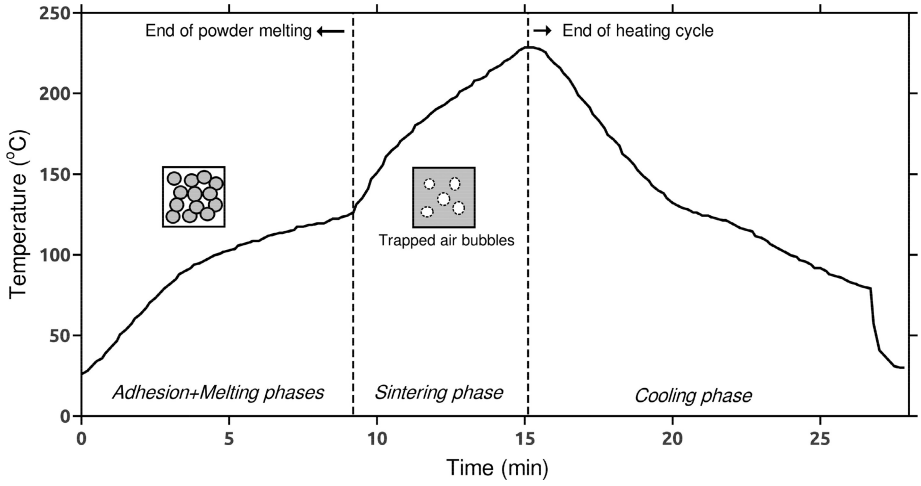

Figure 2.1: The internal mould temperature profile of a rotomoulding process batch run shows the distinct phases of the process. Figure from From Ref. [42].

polymer begin to evaporate during what is known as the sintering phase. During the sintering phase the rate of temperature change increases. This is the most important phase in the process. If the polymer does not sinter for long enough it will have too many deformities to meet quality specifications, whereas if it sinters for too long it will degrade. Once the sintering stage is done, the mould is removed from the oven so that the polymer can cool and solidify. Sometimes, water or compressed air is used to cool the polymer. Once the finished part has cooled and solidified, the product quality is tested. Section 2.5 details the methods that were used for measuring the product quality in this work.

It is well known that the final quality of rotomoulded parts is related to the temperature trajectory of the process [41, 43]; therefore, a model that can accurately predict the trajectory is useful for process control. A common approach is to build a first-principles mechanistic model of the process. For rotomoulding this is difficult because of the numerous dynamic relationships at play including conductive and convective heat transfer, discrete particle dynamics, bubble formation, bubble evaporation, and polymer rheology [16, 43, 44]. Obtaining the parameters required for all of these relationships requires many identifying experiments that must be re-done if the process

is ever changed or updated. Another option is to estimate some of the parameters, but this reduces the accuracy of the model.

Further complicating the use of mechanistic models for rotomoulding is that quality variables cannot be measured until the process finishes. As a result, mechanistic models cannot relate the process dynamics to the final product quality. Batch SSID, which is described in detail in the next subsection, is a data-driven model that provides us with a method of doing this. Because the final subspsace states can be expressed as a linear combination of all previous inputs and states, the math behind SSID presents a way to relate the state trajectory of the process to the product quality. A linear model can be built with the final states as the inputs and the final product quality variables as the outputs. The use of SSID also makes it easy to integrate into an MPC scheme. This is especially useful because it allows for the introduction of process constraints into control schemes. It is not possible to do this with traditional methods such as PID control.

SSID was tested on a laboratory-scale, uni-axial rotomoulding process in Refs. 45 and 46, but the underlying models utilized in these works are not designed to capture the multi-phase nature of the process. This is especially problematic early in the process when the prediction horizon of the model is large and spans multiple phases. In the subsequent sections we will introduce a method of SSID for multi-phase process, MPSSID. The accuracy of MPSSID in comparison to traditional, one-phase SSID will then be tested using data sets obtained from historical batch runs, some of which were analyzed in Refs. 45 and 46. The mathematical background required for an understanding of MPSSID is presented in the next subsection.

2.3.2 Batch Subspace Identification

This subsection explains the basics of an existing batch SSID approach. SSID methods use historical input-output data and linear algebra techniques (Singular Value Decomposition and/or QR factorization) to identify the underlying states and to identify system matrices that are then used to build a linear, black-box model of a process. Previous state-space methods relied on iterative, least-squares methods to identify system parameters (see, e.g., Ref. 47). The use of linear algebra techniques makes SSID more stable. Batch SSID exploits the structure of the SSID algorithm so that unequal batch lengths can be incorporated into the development of a black-box model.

For the purposes of this work, we consider a simple, deterministic SSID model which takes the following form:

$$\mathbf{x}[k+1] = \mathbf{A}\mathbf{x}[k] + \mathbf{B}\mathbf{u}[k] \tag{2.1a}$$

$$\mathbf{y}[k] = \mathbf{C}\mathbf{x}[k] + \mathbf{D}\mathbf{u}[k] \tag{2.1b}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the state, $\mathbf{u} \in \mathbb{R}^m$ is the input, $\mathbf{y} \in \mathbb{R}^l$ is the output; and $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times m}$, $\mathbf{C} \in \mathbb{R}^{l\times n}$, and $\mathbf{D} \in \mathbb{R}^{l\times m}$ are the system matrices. $\mathbf{n}$ is the number of states, $\mathbf{m}$ is the number of inputs, and $\mathbf{l}$ is the number of outputs. k is a time indexing variable that indicates a regular sample-time interval.

To begin the identification, Hankel matrices are constructed using input and output data of each batch, b, within a set of B training batches:

$$\mathbf{U}^b_{1|i} = \begin{bmatrix} \mathbf{u}^b[1] & \mathbf{u}^b[2] & \cdots & \mathbf{u}^b\left[j^b\right] \\ \vdots & \vdots & & \vdots \\ \mathbf{u}^b[i] & \mathbf{u}^b[i+1] & \cdots & \mathbf{u}^b\left[i+j^b-1\right] \end{bmatrix} \tag{2.2}$$

$$\mathbf{Y}^b_{1|i} = \begin{bmatrix} \mathbf{y}^b[1] & \mathbf{y}^b[2] & \cdots & \mathbf{y}^b\left[j^b\right] \\ \vdots & \vdots & & \vdots \\ \mathbf{y}^b[i] & \mathbf{y}^b[i+1] & \cdots & \mathbf{y}^b\left[i+j^b-1\right] \end{bmatrix} \tag{2.3}$$

The $1|i$ subscript is used to indicate that the first column of the Hankel matrix contains the inputs/outputs from 1 to i (where i is a chosen parameter indicating the number of 'Hankel rows'). j^b represents the number of columns. Since we are generalizing to multi-input/multi-output processes containing $\mathbf{m}$ inputs and $\mathbf{l}$ outputs, each input, $\mathbf{u}[k]$, can be assumed to be $\mathbf{m} \times \mathbf{1}$ and each output, $\mathbf{y}[k]$, can be assumed to be $\mathbf{l} \times \mathbf{1}$. For example:

$$\mathbf{U}^b_{1|i} = \begin{bmatrix} u^b_1[1] & u^b_1[2] & \cdots & u^b_1\left[j^b\right] \\ \vdots & \vdots & & \vdots \\ u^b_m[1] & u^b_m[2] & \cdots & u^b_m\left[j^b\right] \\ \vdots & \vdots & & \vdots \\ u^b_1[i] & u^b_1[i+1] & \cdots & u^b_1\left[i+j^b-1\right] \\ \vdots & \vdots & & \vdots \\ u^b_m[i] & u^b_m[i+1] & \cdots & u^b_m\left[i+j^b-1\right] \end{bmatrix} \tag{2.4}$$

A vector of n states can be defined for each batch as:

$$\mathbf{X}^b_1 = \begin{bmatrix} \mathbf{x}^b[1] & \mathbf{x}^b[2] & \cdots & \mathbf{x}^b\left[j^b\right] \end{bmatrix} \tag{2.5}$$

where $\mathbf{X}^b_1$ is $n \times j^b$

Now that the Hankel and state vector matrices have been defined, they can be sub-

stituted into equations 2.1a and 2.1b

$$\mathbf{X}^b_{i+1} = \mathbf{A}^i\mathbf{X}^b_1 + \mathbf{\Delta}_i\mathbf{U}^b_{1|i}, \tag{2.6a}$$

$$\mathbf{Y}^b_{1|i} = \mathbf{\Gamma}_i\mathbf{X}^b_1 + \mathbf{H}_i\mathbf{U}^b_{1|i} \tag{2.6b}$$

where $\mathbf{A}^i$ is the system matrix, $\mathbf{A}$, raised to the ith power; $\mathbf{\Gamma}_i$ and $\mathbf{H}_i$ are matrices that are derived in ref. 35 by iterative substitutions of equations 2.1a and 2.1b. $\mathbf{\Delta}_i$ is also derived by iterative substitution and is defined as:

$$\mathbf{\Delta}_i = \begin{bmatrix} \mathbf{A}^{i-1}\mathbf{B} & \mathbf{A}^{i-2}\mathbf{B} & \cdots & \mathbf{AB} & \mathbf{B} \\ \mathbf{A}^{i-2}\mathbf{B} & \mathbf{A}^{i-3}\mathbf{B} & \cdots & \mathbf{B} & 0 \\ \vdots & \vdots & & \vdots & \\ \mathbf{AB} & \mathbf{B} & \cdots & 0 & 0 \\ \mathbf{B} & 0 & \cdots & 0 & 0 \end{bmatrix} \tag{2.7}$$

Similar to equation 2.6b, an output matrix with initial row values spanning from $i+1$ to $2i$ can be defined:

$$\mathbf{Y}^b_{i+1|2i} = \mathbf{\Gamma}_i\mathbf{X}^b_{i+1} + \mathbf{H}_i\mathbf{U}^b_{i+1|2i} \tag{2.8}$$

where

$$\mathbf{Y}^b_{i+1|2i} = \begin{bmatrix} \mathbf{y}^b[i+1] & \mathbf{y}^b[i+2] & \cdots & \mathbf{y}^b\left[i+j^b\right] \\ \vdots & \vdots & & \vdots \\ \mathbf{y}^b[2i] & \mathbf{y}^b[2i+1] & \cdots & \mathbf{y}^b\left[2i+j^b-1\right] \end{bmatrix} \tag{2.9}$$

and

$$\mathbf{X}_{i+1}^{b} = \begin{bmatrix} \mathbf{x}^{b}[i+1] & \mathbf{x}^{b}[i+2] & \cdots & \mathbf{x}^{b}\left[i+j^{b}\right] \end{bmatrix} \tag{2.10}$$

Equation 2.6b can also be solved for $\mathbf{X}_{1}^{b}$ using the pseudo-inverse of $\mathbf{\Gamma}_{i}$ (denoted by *), which gives:

$$\mathbf{X}_{1}^{b} = -\mathbf{\Gamma}_{i}^{*}\mathbf{H}_{i}\mathbf{U}_{1|i}^{b} + \mathbf{\Gamma}_{i}^{*}\mathbf{Y}_{1|i}^{b} \tag{2.11}$$

When this definition of $\mathbf{X}_{1}^{b}$ is substituted into equation 2.6a the result is:

$$\mathbf{X}_{i+1}^{b} = \begin{bmatrix} \mathbf{\Delta}_{i} - \mathbf{A}^{i}\mathbf{\Gamma}_{i}^{*}\mathbf{H}_{i} & \mathbf{A}^{i}\mathbf{\Gamma}_{i}^{*} \end{bmatrix} \begin{bmatrix} \mathbf{U}_{1|i}^{b} \\ \mathbf{Y}_{1|i}^{b} \end{bmatrix} \tag{2.12}$$

Equations 2.8 and 2.12 show the key concept behind SSID methods. A "future" state sequence ($\mathbf{X}_{i+1}^{b}$) can be defined using the inputs and outputs of a previous interval $(1|j)$. Since $\mathbf{X}_{i+1}^{b}$ appears in both equations, the input and output matrices $\mathbf{Y}_{1|i}^{b}$, $\mathbf{U}_{1|i}^{b}$ and $\mathbf{Y}_{i+1|2i}^{b}$, $\mathbf{U}_{i+1|2i}^{b}$ can be used to solve for the system matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$ which appear in the unknown quantities $\mathbf{\Delta}_{i}$, $\mathbf{\Gamma}_{i}$, and $\mathbf{H}_{i}$. In this work we do this by using the singular value decomposition methods outlined in Ref. 35.

Remark 1. *There are a couple of points that must be clarified about the "future" state sequence, $\mathbf{X}_{i+1}^{b}$, which, calculated by using SSID, runs from $i+1/i+j$. First, the term "future" is historically used due to the artificial splitting of training data into past $(1|i)$ and future $(i+1|2i)$ blocks for computations- all the data is training data, and thus not really "future." Second, the state sequence that is computed in this process does not include the values $\mathbf{x}^{b}[1]$, ...*

, $\mathbf{x}^b[i]$ *(i entries) nor* $\mathbf{x}^b\left[i+j^b+1\right]$*, ... ,* $\mathbf{x}^b\left[N^b\right]$ *(i-1 entries).*

Remark 2. *Statistically, as* j^b *increases, the probability of the input/output matrices being full rank (which is required for SSID) increases. Therefore,* j^b *is conventionally chosen to be the largest possible value. Since both* $\mathbf{Y}^b_{1|i}$ *and* $\mathbf{Y}^b_{i+1|2i}$ *are required to perform the identification procedure, the largest possible value of* j^b *is* N^b-2i+1 *(where* N^b *is the number of observations within the batch). If* N^b-2i+1 *is substituted into the last row/last column of* $\mathbf{Y}^b_{i+1|2i}$ *the result is an index of* N^b*.*

To identify a model that accounts for all available batch data, the Hankel matrix for each batch is combined horizontally into one "batch Hankel" matrix:

$$\mathbf{Y}_{1|i} = \begin{bmatrix} \mathbf{Y}^1_{1|i} & \mathbf{Y}^2_{1|i} & \cdots & \mathbf{Y}^B_{1|i} \end{bmatrix} \tag{2.13}$$

Once the matrices are set up in this way, we can use the same SSID techniques introduced above and outlined in Ref. 35 to first calculate a state sequence,

$$\mathbf{X}_{i+1} = \begin{bmatrix} \mathbf{X}^1_{i+1} & \mathbf{X}^2_{i+1} & \cdots & \mathbf{X}^B_{i+1} \end{bmatrix} \tag{2.14}$$

and then use it to calculate the system matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$. It is important to note that this identification procedure can handle batches of varying lengths. The batch data is concatenated horizontally, so only the number of rows in each submatrix need to match. This is easily done by keeping the number of inputs and outputs consistent across batches.

The purpose of this work is to show how one can carefully utilise the existing SSID algorithms that have been introduced in this section for multi-phase batch processes. The numerically stable SSID methods implemented in this work are described in detail in Ref. 35. Reference 48 provides a good review and analysis of a variety of

SSID methods. In Ref. 38, the SSID methods described in Ref. 35 were generalized to batch processes by modifying the Hankel matrices (as shown above) and outperformed existing LV-MPC models. In Ref. 39, these methods were further generalized to batch processes with discrete events. This work draws inspiration these three sources to address the problem of capturing process nonlinearity by formulating a multi-phase batch subspace identification procedure.

2.4 Proposed Model: Multi-Phase SSID

In this section we describe the proposed **M**ulti-**P**hase **S**ubspace **ID**entification (MPSSID) technique.

2.4.1 Hankel Matrices

To begin, we organize the batch training data into distinct phases. This can be done by using a *unique process identifier* (i.e. a set of process conditions that indicate a phase change when they are met). For example, once a measured output reaches a specific value, the previous phase ends and the next phase begins.

> **Remark 3.** *The present manuscript (and approach) is described using the situation where there exists a unique (single-variable) process identifier. This assumption readily holds for chemical and biological processes where the dependence of phase changes on measured process variables such as pressures, temperatures etc. are well known. The dependence of the phase change will affect the piece-wise conditions in equation 2.15 below. For the rotomoulding case, the structure of equation 2.15 is based on a phase change that is induced when one, constantly increasing measured output (temperature), reaches spec-*

ified values.

Thus, let there be a phase identifier function, which for a set $P = [1, 2, \dots P]$ is such that $p(y[k]) : R^l \rightarrow P$. In this case:

$$p(y[k]) = \begin{cases} 1\ , & y < y_1 \ \ \forall\ y \\ 2\ , & y \leq y_2 \ \ \forall\ y\ and\ \exists\ y \geq y_1 \\ 3\ , & y \leq y_3 \ \ \forall\ y\ and\ \exists\ y \geq y_2 \\ \dots & \\ p\ , & y \leq y_p \ \ \forall\ y\ and\ \exists\ y \geq y_{p-1} \end{cases} \tag{2.15}$$

In other words, for this formulation, the phase, p, permanently changes each time the output first reaches a predefined value $(y_1, y_2,\ \dots\ , y_p)$. We define the points at which these phase changes occur as $k^1_{change}, k^2_{change},\ \dots\ , k^p_{change}$.

The identifying Hankel matrices take the following form for each phase of each batch:

$$\mathbf{Y}^{b,p}_{1|i} = \begin{bmatrix} \mathbf{y}^{b,p}[1] & \mathbf{y}^{b,p}[2] & \cdots & \mathbf{y}^{b,p}\left[j^{b,p}\right] \\ \vdots & \vdots & & \vdots \\ \mathbf{y}^{b,p}[i^p] & \mathbf{y}^{b,p}[i^p + 1] & \cdots & \mathbf{y}^{b,p}\left[i^p + j^{b,p} - 1\right] \end{bmatrix} \tag{2.16}$$

The number of Hankel rows, i^p, is a user-defined parameter that can differ for each phase. For identifiability, i^p should be greater than the number of states, n^p, chosen for the corresponding phase. $j^{b,p}$ is unique to each phase of each batch because it is dependent on the number of measurements, $N^{b,p}$, within a phase, p, of a batch, b. It is also dependent on the choice of i^p. Note that we reset the time index, k, to 1 at the beginning of a new phase. Also note that the number of phases that a user can model for a process is limited by the size of each phase ($N^{b,p} > 2i^p - 1$, at minimum). If a phase does not have enough observations, the Hankel matrices will become too square or become mathematically impossible to create.

Remark 4. *Recall from section 2.3.2 that* $j^b = N^b - 2i + 1$. *In this case* $j^{b,p} = N^{b,p} - 2i^p + 1$.

Next, we horizontally concatenate the submatrices:

$$\mathbf{Y}^p_{1|i} = \begin{bmatrix} \mathbf{Y}^{1,p}_{1|i} & \mathbf{Y}^{2,p}_{1|i} & \cdots & \mathbf{Y}^{B,p}_{1|i} \end{bmatrix} \tag{2.17}$$

For MPSSID, we have P batch Hankel matrices (i.e. $\mathbf{Y}^1_{1|i}$, $\mathbf{U}^1_{1|i}$, ... $\mathbf{Y}^P_{1|i}$, $\mathbf{U}^P_{1|i}$). When we do the SSID for each phase we end up with P sets of system matrices: $\mathbf{A}^1$, $\mathbf{B}^1$, $\mathbf{C}^1$, $\mathbf{D}^1$... $\mathbf{A}^P$, $\mathbf{B}^P$, $\mathbf{C}^P$, $\mathbf{D}^P$. This added complexity means we need an extra step for transitioning between phases.

Remark 5. *It is important to clarify how the data is grouped by phase. One method is to assume the phase change occurs instantaneously. Under this assumption, input/output data-points corresponding to the time index at which the phase change occurs are included for both the* p *and* $p+1$ *phases (i.e.* $\mathbf{y}^{b,p}\left[N^{b,p}\right] = \mathbf{y}^{b,p+1}[1]$*). The results presented in this work are based on this assumption. Another method is to assume all phases are separate. For this method there is no duplication of data points in adjacent phases. This second method was tested and gave similar results, and the details are therefore omitted for the sake of brevity.*

2.4.2 Phase Transitions: Propagation

Calculating system matrices for each phase makes it possible to define an LTI state space system for each phase:

$$\mathbf{x}^p[k+1] = \mathbf{A}^p\mathbf{x}^p[k] + \mathbf{B}^p\mathbf{u}^p[k]\,, \tag{2.18a}$$

$$\mathbf{y}^p[k] = \mathbf{C}^p\mathbf{x}^p[k] + \mathbf{D}^p\mathbf{u}^p[k] \tag{2.18b}$$

The next step is the transition from one phase to the next. There are two important things to consider for the phase transitions: first, we must define process conditions that initiate a switch from $\mathbf{A}^p$, $\mathbf{B}^p$, $\mathbf{C}^p$, $\mathbf{D}^p$ to $\mathbf{A}^{p+1}$, $\mathbf{B}^{p+1}$, $\mathbf{C}^{p+1}$, $\mathbf{D}^{p+1}$, and second, we must recognize that due to our phase-by-phase identification procedure, the calculated states sequences ($\mathbf{X}^p_{i+1}$) are phase-specific. Therefore, we need a method to relate $\mathbf{x}^{b,p}\left[N^{b,p}\right] \in \mathbb{R}^{n^p\times 1}$ to $\mathbf{x}^{b,p+1}[1] \in \mathbb{R}^{n^{p+1}\times 1}$. Building from Section 2.3.2, Remark 1, we know the state sequence derived from Batch SSID for the p^{th} phase is missing the first i^p values and last $i^p - 1$ values of each batch. It is necessary to obtain these missing values to construct a relationship between $\mathbf{x}^{b,p}\left[N^{b,p}\right]$ to $\mathbf{x}^{b,p+1}[1]$.

Calculating $\mathbf{x}^{b,p}[N^p]$ is easily done by using the identified system matrices to propagate forward through the $i-1$ missing values:

$$\mathbf{x}^{b,p}\left[i^p + j^{b,p} + \kappa\right] = \mathbf{A}^p\mathbf{x}^{b,p}\left[i^p + j^{b,p}\right] + \mathbf{B}^p\mathbf{u}^{b,p}\left[i^p + j^{b,p}\right] \quad \forall\kappa \in 1, 2, ..., i^p - 1 \tag{2.19}$$

Calculating $\mathbf{x}^{b,p+1}[1]$ is more intricate. It is possible to rearrange equations 2.1a and 2.1b and use the inverse of $\mathbf{A}^p$ to back propagate:

$$\mathbf{x}^{b,p+1}[k] = \left(\mathbf{A}^{p+1}\right)^{-1}\left(\mathbf{x}^{b,p+1}[k+1] - \mathbf{B}^{p+1}\mathbf{u}^{b,p+1}[k]\right) \quad \forall k \in i^p, i^p - 1, ..., 1 \tag{2.20}$$

In practice, the inverse $\mathbf{A}$ matrix is often not stable, and so the back-propagated states can blow up when using this method. A better way is to use the observer defined in Ref. 49, which is modified in Ref. 39 to be useful for back-propagation as shown below:

$$\min_{\mathbf{x}^{b,p+1}[k]} \left(\mathbf{y}^{b,p+1}[k] - \hat{\mathbf{y}}^{b,p+1}[k]\right) \mathbf{R} (\mathbf{y}^{b,p+1}[k] - \hat{\mathbf{y}}^{b,p+1}[k]) + \omega_k \mathbf{Q} \omega_k \quad \forall k \in i^p, i^p - 1, ..., 1 \tag{2.21}$$

subject to

$$\omega_k = \mathbf{x}^{b,p+1}[k+1] - \left(\mathbf{A}^{p+1}\mathbf{x}^{b,p+1}[k] + \mathbf{B}^{p+1}\mathbf{u}^{b,p+1}[k]\right) \tag{2.22}$$

$$\hat{\mathbf{y}}^{b,p+1}[k] = \mathbf{C}^{p+1}\mathbf{x}^{b,p+1}[k] + \mathbf{D}^{p+1}\mathbf{u}^{b,p+1}[k] \tag{2.23}$$

where $\mathbf{x}^{b,p+1}[k]$ is the unknown state at the previous time index, which the optimization problem solves for; $\mathbf{x}^{b,p+1}[k+1]$ is the known state that we back-propagate from; $\hat{\mathbf{y}}^{b,p+1}[k]$ is the model predicted output from the previous time index; $\mathbf{y}^{b,p+1}[k]$ is the measured output from the previous time index; and $\mathbf{u}^{b,p+1}[k]$ is the previous input. $\mathbf{y}^{b,p+1}[k]$, and $\mathbf{u}^{b,p+1}[k]$ are known since we are using training data at this stage and so know the measured inputs and outputs from each time index. $\mathbf{R}$ and $\mathbf{Q}$ are tuning matrices of the optimization problem. For this work they are set to the identity matrix.

Remark 6. *Note that for the back-propagation case, the entire trajectory,* $\mathbf{x}^{b,p+1}[1]$ *to* $\mathbf{x}^{b,p+1}[i^{p+1}]$*, can be solved for in one step if the objective function is expanded for* i^{p+1} *decision variables. In this work we use the method above, and so the objective function is solved* i^{p+1} *times in an iterative manner. The results did not significantly change between methods.*

2.4.3 The Phase Transition PLS model

For each batch, we have now calculated the states at the end of each phase, $\mathbf{x}^{b,p}\left[N^{b,p}\right]$, and beginning of each following phase, $\mathbf{x}^{b,p+1}[1]$. Note that for our method, $\mathbf{x}^{b,p}\left[N^{b,p}\right]$ and $\mathbf{x}^{b,p+1}[1]$ correspond to the same time index. Therefore we redefine $\mathbf{x}^{b,p+1}[1]$ as $\mathbf{x}^{b,p+1}[1^-]$ to indicate that one iteration through equation 2.18a is required to reach the first model prediction in the state space of the next phase. Next, we construct a model to connect the two phases. This is done by using Partial Least Squares (PLS). For more on PLS, see Ref. 50. We define $\mathbf{X}^p_{PLS}$ as the matrix of predictor variables and $\mathbf{Y}^p_{PLS}$ as the matrix of response (calculated) variables:

$$\mathbf{X}^p_{PLS} = \begin{bmatrix} \left(\mathbf{x}^{1,p}[N^{1,p}]\right)^T \\ \vdots \\ \left(\mathbf{x}^{B,p}\left[N^{B,p}\right]\right)^T \end{bmatrix} \tag{2.24}$$

$$\mathbf{Y}^p_{PLS} = \begin{bmatrix} \left(\mathbf{x}^{1,p+1}[1^-]\right)^T \\ \vdots \\ \left(\mathbf{x}^{B,p+1}[1^-]\right)^T \end{bmatrix} \tag{2.25}$$

Note that the number of columns within the $\mathbf{X}^p_{PLS} \in \mathbb{R}^{B\times n^p}$ and $\mathbf{Y}^p_{PLS} \in \mathbb{R}^{B\times n^{p+1}}$ matrices are equivalent to the number of states, n^p and n^{p+1}, respectively. Performing the PLS step yields the parameters $\mathbf{W}^p_{PLS} \in \mathbb{R}^{n^p\times\Lambda^p}$, $\mathbf{W}^p_{PLS}{}^* \in \mathbb{R}^{n^p\times\Lambda^p}$, $\mathbf{P}^p_{PLS} \in \mathbb{R}^{n^p+\times\Lambda^p}$, and $\mathbf{C}^p_{PLS} \in \mathbb{R}^{n^{p+1}\times\Lambda^p}$ where Λ^p is the number of user-specified components of the PLS model constructed for the transition of phase p to $p+1$. These matrices are necessary to calculate $\mathbf{x}^{p+1}[1^-]$ from $\mathbf{x}^p[N^p]$ during new test runs.

2.4.4 Quality Model

We can now use the system matrices for each phase in combination with the PLS models for the phase transitions to predict a state and output trajectory for the entire process. The last feature of the MPSSID model is a prediction method for the final product quality. Recall from subsection 2.3.1 that the final product quality of the rotomoulding process is related to the temperature trajectory. The information of the temperature trajectory is coded in the final value of the system state (by definition of the notion of state space), and thus a quality model is built that relates only the final states to the product quality.

The quality model implemented in this work is constructed by using PLS. We define $\mathbf{X}_{PLS}^{Quality}$ as the matrix of predictor variables and $\mathbf{Y}_{PLS}^{Quality}$ as the matrix of response variables:

$$\mathbf{X}_{PLS}^{Quality} = \begin{bmatrix} \left(\mathbf{x}^{1,P}\left[N^{1,P}\right]\right)^T \\ \vdots \\ \left(\mathbf{x}^{B,P}\left[N^{B,P}\right]\right)^T \end{bmatrix} \tag{2.26}$$

$$\mathbf{Y}_{PLS}^{Quality} = \begin{bmatrix} q_1^1 & q_2^1 & \dots & q_{N_q}^1 \\ \vdots & \vdots & \ddots & \vdots \\ q_1^B & q_2^B & \dots & q_{N_q}^B \end{bmatrix} \tag{2.27}$$

where N_q represents the number of measured quality variables. Note that the $\mathbf{X}_{PLS}^{Quality} \in \mathbb{R}^{B \times n^P}$ matrix is populated with the final state values of the last phase, P, for each training batch. Performing this PLS step yields the parameters $\mathbf{W}_{PLS}^{Quality} \in \mathbb{R}^{n^P \times \Lambda^{Quality}}$, $\mathbf{W}_{PLS}^{Quality^*} \in \mathbb{R}^{n^P \times \Lambda^{Quality}}$, $\mathbf{P}_{PLS}^{Quality} \in \mathbb{R}^{n^P \times \Lambda^{Quality}}$, and $\mathbf{C}_{PLS}^{Quality} \in \mathbb{R}^{N_q \times \Lambda^{Quality}}$ where $\Lambda^{Quality}$ is the number of components for the PLS quality model.

2.4.5 Testing/Validation

Using the system matrices and PLS relationships that have been identified from the training data, we can implement the predictive model for new process runs to validate the approach. A key recognition is that MPSSID can be used for prediction purposes only *after* the model states have been estimated by using a few of the initial data points. In the present manuscript, this is done with a Luenberger observer.

We test the model's prediction capability from various points in a test batch, and the following procedure describes how to test it from an arbitrary point, $y[k_c^p]$, in phase $p(y[k_c^p]) \equiv p_c$. Note that this is done to mimic the model's ability to generate good predictions in an MPC type application, where the control action needs to be computed throughout the batch and relies on the model's predictive capability at various time points during the batch. To this end we define points, k^p_{change}, that represent the instances where the criteria in equation 2.15 are met, triggering a transition to the next phase. We also need the following sets of equations:

$$\mathbf{x}^p[k+1] = \mathbf{A}^p\mathbf{x}^p[k] + \mathbf{B}^p\mathbf{u}^p[k] + \mathbf{L}^p(\mathbf{y}^p[k] - \hat{\mathbf{y}}^p[k]) \tag{2.28a}$$

$$\hat{\mathbf{y}}^{[k]} = \mathbf{C}^p\mathbf{x}^p[k] + \mathbf{D}^p\mathbf{u}^p[k] \tag{2.28b}$$

$$\mathbf{x}^{p+1}\left[1^-\right] = \left(\left(\mathbf{x}^p\left[k^p_{change}\right]\right)^T \mathbf{W}^p_{PLS}{}^*(\mathbf{C}^p_{PLS})^T\right)^T \tag{2.29a}$$

$$\mathbf{x}^{p+1}[1] = \mathbf{A}^{p+1}\mathbf{x}^{p+1}\left[1^-\right] + \mathbf{B}^{p+1}\mathbf{u}^p\left[k^p_{change}\right] + \mathbf{L}^{p+1}\left(\mathbf{y}^p\left[k^p_{change}\right] - \hat{\mathbf{y}}^p\left[k^p_{change}\right]\right) \tag{2.29b}$$

$$\mathbf{x}^p[k+1] = \mathbf{A}^p\mathbf{x}^p[k] + \mathbf{B}^p\mathbf{u}^p[k] \tag{2.30a}$$

$$\hat{\mathbf{y}}^p[k] = \mathbf{C}^p\mathbf{x}^p[k] + \mathbf{D}^p\mathbf{u}^p[k] \tag{2.30b}$$

$$\mathbf{x}^{p+1}\left[1^-\right] = \left(\left(\mathbf{x}^p\left[k^p_{change}\right]\right)^T \mathbf{W}^p_{PLS}{}^*(\mathbf{C}^p_{PLS})^T\right)^T \tag{2.31a}$$

$$\mathbf{x}^{p+1}[1] = \mathbf{A}^{p+1}\mathbf{x}^{p+1}\left[1^-\right] + \mathbf{B}^{p+1}\mathbf{u}^p\left[k^p_{change}\right] \tag{2.31b}$$

$$\begin{bmatrix}\mathbf{q}_1 & \mathbf{q}_2 & \dots & \mathbf{q}_{N_q}\end{bmatrix} = \left(\mathbf{x}^P[N^p]\right)^T \mathbf{W}^{Quality*}_{PLS}\left(\mathbf{C}^{Quality}_{PLS}\right)^T \tag{2.32}$$

Using these sets of equations, we implemented the following algorithm for testing:

1. Initialization: $k = 1$, $p = 1$
2. Checking: While $(p < p_c)$ or $(p = p_c$ and $k \leq k^p_c)$, continue with steps 3 and 4 (enabling observer-aided state estimation up-to the current point, k^p_c). If $(p = p_c$ and $k > k^p_c)$ or $(p > p_c)$, continue with steps 6 and 7 (state estimation without observer). For prediction of terminal qualities, use step 8.
3. Propagation: Until $p(y[k]) = p + 1$ conditions are satisfied (i.e., as the process is evolving in phase p), repeat equations 2.28a and 2.28b followed by a step 2 check after each iteration.
4. Switching: Once $p(y[k]) = p + 1$ conditions are satisfied at a point, k^p_{change} (i.e., a change of phase has to be accounted for during the estimation), use equation 2.29a followed by equation 2.29b. Go to step 5 (i.e., continue state estimation under the next phase).

5. Update Counters: $k = 1$, $p = p + 1$, go back to step 2.

6. Propagation (prediction stage): Note that this step corresponds to time points for which the model is being used to predict process variables. Thus, until $p(y[k]) = p + 1$ conditions are satisfied, repeat equations 2.30a and 2.30b followed by a step 2 check after each iteration. These equations correspond to the propagation using the state space equations without the observer.

7. Switching (prediction stage): Once $p(y[k]) = p + 1$ conditions are satisfied at a point, k^p_{change}, do equation 2.31a followed by 2.31b (this accounts for change of phase during the prediction stage). Go to step 5.

8. Termination: At the last step, use the last state of the last phase, $x^P\left[N^P\right]$, to predict the product quality using equation 2.32.

To summarize, **up to a point**, k^p_c, the outputs have been measured, and so the state is estimated using a Luenberger observer. If a phase transition is triggered by a measured output, a PLS step that connects the previous phase to the next phase is required before continuing with the Luenberger observer state space equations in the next phase. **Past** k^p_c, the observer is no longer used and so the state space equation is used to generate predictions. During this stage as well, if a phase transition is triggered by a predicted output, a PLS step is required to account for the transition. Once the process is finished, the PLS model that connects the terminal phase to the terminal quality is used to predict the product quality. Note that in step 2 of the algorithm, stop criteria would be required to indicate process completion and trigger step 8. For this work, we used predetermined inputs from historical batches to compare the predictive ability of one-phase and multi-phase models to observed data, so stop criteria was unnecessary.

Remark 7. *For this work, the poles of the Luenberger observer are user-specified for each phase, hence the subscript, p, for* L_p*. In the present work,*

the observer poles were chosen to be the same for all phases for simplicity.

Remark 8. *We wrote* $\mathbf{W}_{PLS}^{p}{}^{*}$ *and* $\mathbf{W}_{PLS}^{Quality*}$ *in equations 2.29a, 2.31a, and 2.32 for simplification. These matrices are calculated from PLS using the equation:* $\mathbf{W}^{*} = \mathbf{W}\left(\mathbf{P}^{'}\mathbf{W}\right)^{-1}$. *To avoid inconsistencies that can stem from an ill-conditioned* $\left(\mathbf{P}^{'}\mathbf{W}\right)$ *inverse, we recommend using the score matrix,* $\mathbf{T}_{PLS,new} \in 1 \times \Lambda$, *instead (where* Λ *is the number of components used in the PLS algorithm). The components of* $\mathbf{T}_{PLS,new}$ *are determined by doing element-by-element calculations (see Ref. 50 for more). For the phase change case this looks like:*

$$\mathbf{x}_1^T = \left(\mathbf{x}^p\left[k_{change}^p\right]\right)^T \tag{2.33a}$$

$$\mathbf{t}_\lambda = \mathbf{x}_\lambda^T \mathbf{w}_\lambda^p \qquad \forall\ \lambda \in 1:\Lambda \tag{2.33b}$$

$$\mathbf{x}_{\lambda+1}^T = \mathbf{x}_\lambda^T - \mathbf{t}_\lambda (\mathbf{p}_\lambda^p)^T \qquad \forall\ \lambda \in 1:\Lambda-1 \tag{2.33c}$$

where $\mathbf{w}_\lambda^p, \mathbf{p}_\lambda^p \in \mathbb{R}^{n^p \times 1}$ *represent the* λ^{th} *component of* $\mathbf{W}_{PLS}^p$ *and* $\mathbf{P}_{PLS}^p$, *respectively. Once the components have been calculated by using equations 2.33b and 2.33c in an iterative fashion, we can use* $\mathbf{T}_{PLS,new}$ *to calculate* $\mathbf{x}^{p+1}[1^-]$

$$\mathbf{T}_{PLS,new} = \begin{bmatrix} \mathbf{t}_1 & \mathbf{t}_2 & \cdots & \mathbf{t}_\lambda \end{bmatrix} \tag{2.34a}$$

$$\mathbf{x}^{p+1}\left[1^-\right] = (\mathbf{T}_{PLS,new}(\mathbf{C}_{PLS}^p))^T \tag{2.34b}$$

For the quality calculation this looks like:

$$\mathbf{x}_1^T = \left(\mathbf{x}^P\left[N^P\right]\right)^T \tag{2.35a}$$

$$\mathbf{t}_\lambda = \mathbf{x}_\lambda^T \mathbf{w}_\lambda^{Quality} \qquad \forall\ \lambda \in 1:\Lambda \tag{2.35b}$$

$$\mathbf{x}_{\lambda+1} = \mathbf{x}_\lambda^T - \mathbf{t}_\lambda \left(\mathbf{p}_\lambda^{Quality}\right)^T \qquad \forall\ \lambda \in 1:\Lambda-1 \tag{2.35c}$$

where $\mathbf{w}_{\lambda}^{Quality}, \mathbf{p}_{\lambda}^{Quality} \in n^P \times 1$ *represent the* λ^{th} *component of* $\mathbf{W}_{PLS}^{Quality}$ *and* $\mathbf{P}_{PLS}^{Quality}$*, respectively.*

$$\mathbf{T}_{PLS,new} = \begin{bmatrix} \mathbf{t}_1 & \mathbf{t}_2 & \cdots & \mathbf{t}_{\lambda} \end{bmatrix} \tag{2.36a}$$

$$\begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_{N_q} \end{bmatrix} = \mathbf{T}_{PLS,new} \left(\mathbf{C}_{PLS}^{Quality} \right)^T \tag{2.36b}$$

Using equations 2.28a-2.36b, we can calculate the entire process trajectory from any point $y[k_c^p]$ in the process while propagating through multiple phases.

2.5 Application to the Motivating Example

The key question we seek to test is if the MPSSID model proposed in section 2.4 yields superior prediction accuracy in comparison to one-phase SSID models. We tested this question using data obtained from historical batch runs of the laboratory-scale, uni-axial rotomoulding process that was mentioned in section 2.3.1, which is characterized by multiple, distinct phases. Previous one-phase SSID model-based predictive control formulations were able to meet the specified quality standards [45, 46]; however, the emphasis here is to seek improvements to the predictive capability of the data-driven model to yield improved control performance. Increased confidence in the model and controller are likely to make it easier to transition from lab to industrial scale and experiment with new MPC formulations such as economic control. It should also improve the model's ability to handle disturbances. Note that in Refs. 45 and [46], no control action was active during the cooling phase. Therefore, our analysis focuses only on the heating phases.

For the historical batch runs, the experimental setup consists of two heaters and one air compressor that drive the process as the mould rotates to create a cubic plastic

object from a high-density polyethylene powder (ExxonMobilTM HD8660.29) donated by Imperial Oil (Sarnia, ON). See figure 2.2 for a visual depiction.

The following methods are used to quantify the product quality of the moulded boxes. First, the sinkhole area of the rotomoulded parts is measured to examine the degree of trapped air bubbles, which is influenced by sintering and has a negative effect on durability. This is done by surface image analysis. A thin layer of copper and graphite lubricant is applied to one face of the rotomoulded product to fill in the surface voids so that they stand out in images. A digital image captures a 40 mm x 40 mm area and is post-processed to enhance the contrast of the surface voids. Second, the samples are cut and stored at a temperature of -40 °C for 24 h in preparation for dart impact testing. Dart impact testing measures the impact strength of the moulded part and is influenced by both sintering and degradation effects during the rotomoulding process. A standard dart weight of 6.804 kg is dropped from predetermined heights following a step-wise procedure (if the previous sample failed, decrease dart height by 0.1524 m (0.5 feet); or if sample did not fail, increase height by 0.1524 m). Each face of the mould is used for analysis from each batch, thus resulting in four measurements per batch, to ultimately yield an average value that is calculated using the ASTM D5628 procedure. In Refs. 45 and 46, rheology and ultrasonic spectroscopy quality tests are also done, but subsequent analyses found these measures to be insignificant in determining product quality as they were highly correlated with the other tests. Therefore, these quality measurements were not available for each historical data set and so are omitted from our analysis.

We used a data set comprised of 55 historical batch process runs. Note that for this analysis, the input moves were predetermined from the historical record of the batches. The purpose of this analysis was to determine if the MPSSID method would improve the prediction accuracy of the process over a one-phase model for the *same* input moves as the historical runs. To utilize MPSSID on a new batch, the input

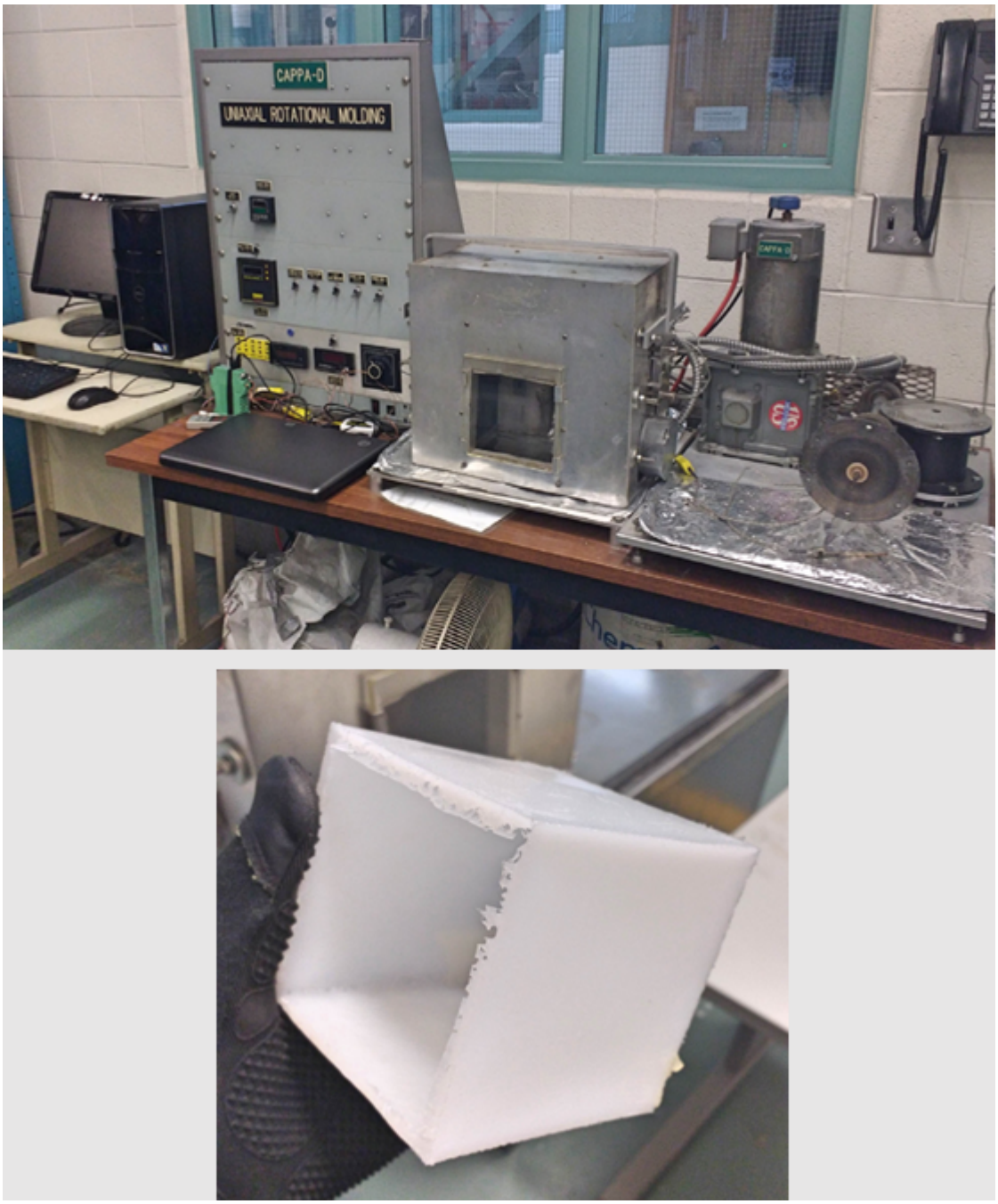

Figure 2.2: The experimental test setup (top) and rotomoulded object (bottom) as shown in Ref. 45. The polymer powder is placed into the mould which is then screwed onto the rotating axis located on the right (top image), and then the oven in the middle is placed on top.

values would be the decision variables of an MPC formulation.

Using the historical input and output data, we built a standard, one-phase SSID model; a two-phase MPSSID model; and a three-phase MPSSID model. 39 of the 55 batches were reserved for the training set while the other 16 were placed in the testing set. The test batches were selected in a semi-random fashion because past experimenters had slightly modified the MPC formulation over time, with three main changes. To be characteristic of the experiments that had been done, batches were first grouped by MPC formulation and then batches from each grouping were selected for the test set at random. The number of batches selected to the test set from each group was weighted by group size.

For this analysis, we used the two heater temperatures as model inputs because they are easy variables to manipulate and were found to describe the evolution of the process better than the power inputs of the two heaters and air compressor. The input profiles of the testing batches are given in figure 2.3.

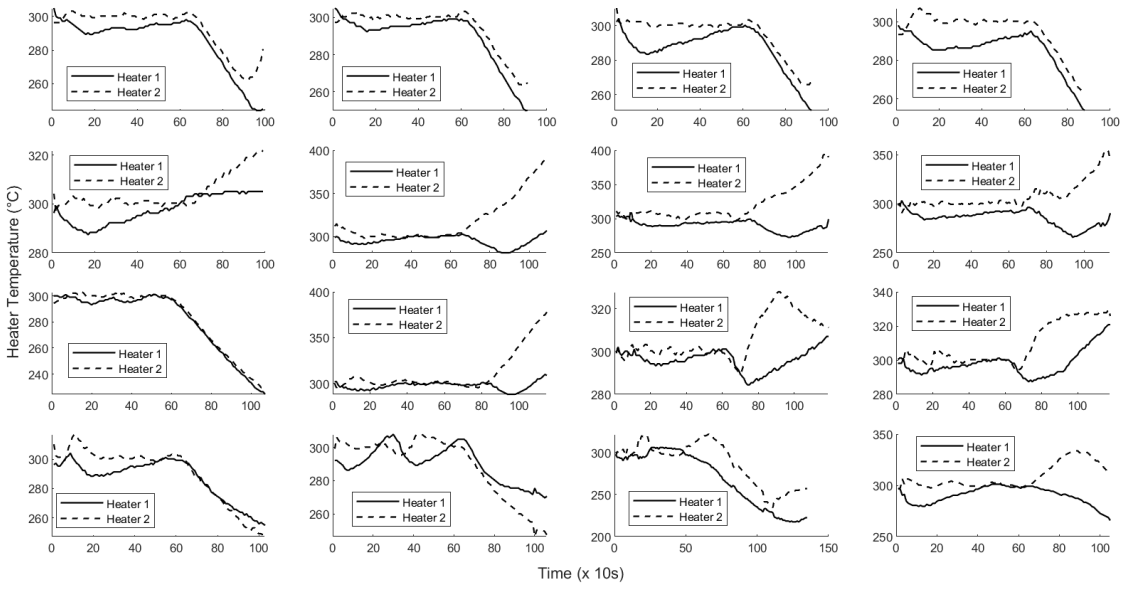

Figure 2.3: Input profiles for the 16-batch testing set.

There are large variations between some of the input profiles due in part to the different MPC formulations that were used. The output of the process is the internal mould temperature. Recall from the beginning of section 2.4.1 that there are two user defined parameters for each phase of SSID: the number of states, n^p, and the number of Hankel rows, i^p. For our MPSSID formulation, we must also specify the unique process identifier that indicates a phase change. For the rotomoulding case, measuring the output temperature gives sufficient indication. Cross-validation was used as an initial test within the 39 training batches to determine the optimal number of states, number of Hankel rows, and phase change temperatures for the one-phase, two-phase, and three-phase models. These phase options were chosen intuitively based on the process dynamics. The two-phase model separates the melting phase from the sintering phase, whereas the three-phase model also accounts for a levelling-off effect that is observed partway through the sintering phase. This is shown in figure 2.4

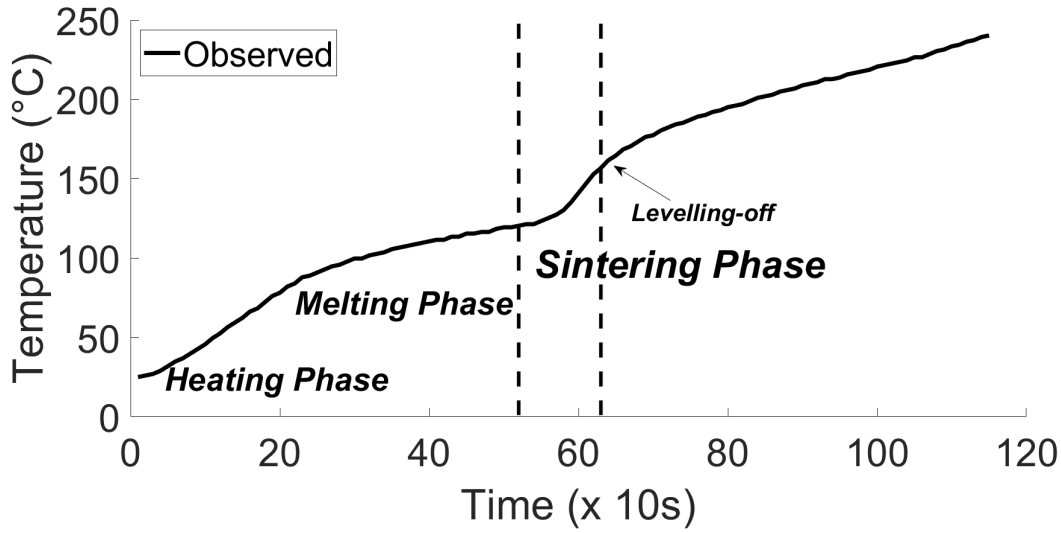

Figure 2.4: A characteristic mould temperature output profile that depicts where the phase changes occur during heating.

Recall that up to an arbitrary point, k_c^p, which can be thought of as the live process measurement, an observer is used to improve the SSID prediction. The prediction capability for each model was tested for five different cases of k_c^p based on batch

completion: 5 %, 20 %, 40 %, 60 %, and 80 %. For instance, for the 5 % case, if the batch has 100 total measurements an observer is used in the model until the fifth observation ($k_c^p = 5, p = 1$) and then is shut off, and the appropriate one-phase, or two-phase or three-phase model is used to predict the rest of the output trajectory. The quality is then predicted using the predicted states at the end of the batch operation. This method of evaluating model accuracy is consistent with its potential use in a model predictive control framework, where the accuracy of the control action at any point in time in the batch would depend on the ability of the model to predict the process evolution for the rest of the batch.

The optimal one-phase, two-phase, and three-phase models were chosen by comparing four metrics which were summed over all 16 test batches for each of the five cases: the summed squared prediction error (SSPE) of the temperature predictions, the SSPE of both product quality predictions, and the count of product quality predictions that were less than zero (which is physically meaningless). The optimal model configurations are shown in table 2.1.

Remark 9. *For a new batch, operators would not know when a specific percentage of completion has been reached. For this analysis, the observed data was available prior to testing, so different batches could be compared in this manner. In a real implementation this information is not required to implement the MPSSID model, but is used here only for comparison purposes.*

Remark 10. *Note that for this analysis we only seek to compare the predictive ability of different models, not the control system performance of them. It should follow that a more accurate predictive model will result in a process with better quality control. This will be tested in future work for confirmation.*

Next, we applied the model configurations outlined in table 2.1 to the 16-batch testing set. Again we tested the five different batch completion cases and used the same

Table 2.1: Optimal model configurations determined from cross-validation. Note the number of states and Hankel rows for each phase are separated by commas for the two- and three-phase models.

Model	n^p	i^p	Phase Change Temperatures
One-phase	2	3	N/A
Two-phase	2, 2	4, 4	128 °C
Three-phase	3, 3, 2	4, 4, 3	120 °C, 155 °C

metrics for comparison, the results are shown in table 2.2.

Note that data in table 2.2 has been summed over all 16 test batches. The two-phase model yields more accurate model predictions than the one-phase model at all stages of batch completion. The three-phase model yields even more accurate predictions. It is intuitive that the predictions should improve as the batch progresses since the Luenberger observer, which has a stabilizing effect on the model state, is left on longer when more measurements become available. This improvement is illustrated in figure 2.5 as the predictions for testing batch 10 improve from left (5% batch completion) to right (80% batch completion). The more notable observation in figure 2.5 is the improvement from the top (one-phase) to the bottom (three-phase). The multi-phase models are clearly more accurate, especially in the neighbourhood of the phase change. A zoomed in version for the 40% batch completion case is shown in figure 2.6. We have only plotted the temperature trajectory of one of the 16 batches for the sake of brevity, but table 2.2 shows the accuracy improvement is generally the case. Future work could focus on minimizing the larger errors that are observed where the batch undergoes a phase change, as for some batches the change is not as smooth for the multi-phase case.

Improvements in temperature trajectory accuracy should result in improvements to the final product quality prediction accuracy. Table 2.2 shows that the sum of squared errors does decrease for both quality metrics. This is also shown in figures 2.7 and 2.8, where the absolute errors of the predicted sinkhole areas (figure 2.7) and impact

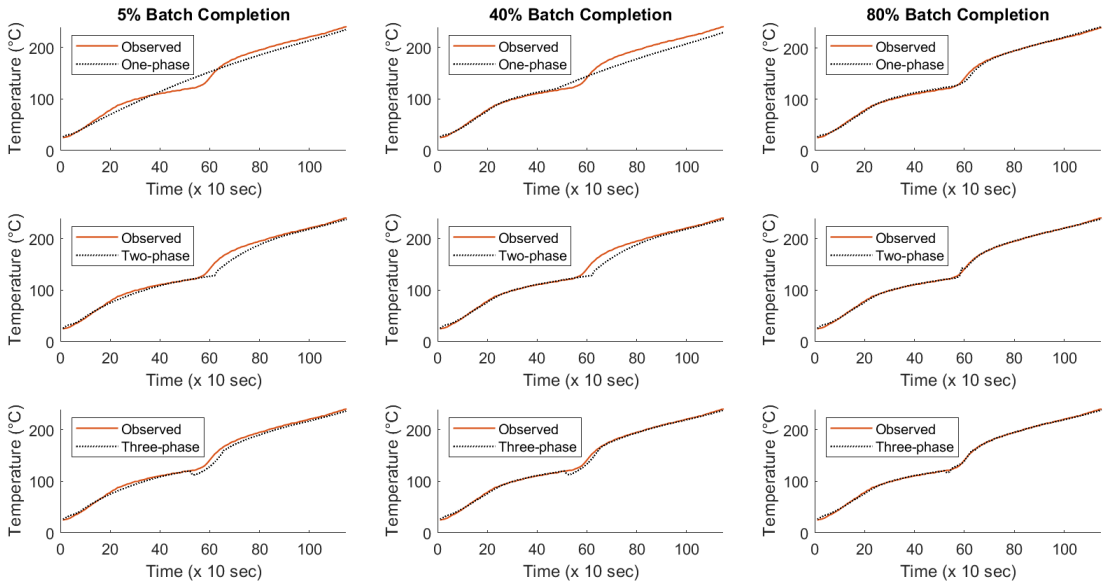

Figure 2.5: Temperature trajectory predictions (·····) from 5%, 40%, and 80% of the way through testing batch 10 for the one-phase (top), two-phase (middle), and three-phase (bottom) models in comparison to the observed (——) batch values. These graphs were constructed from the 10th batch of the 16 batch testing set.

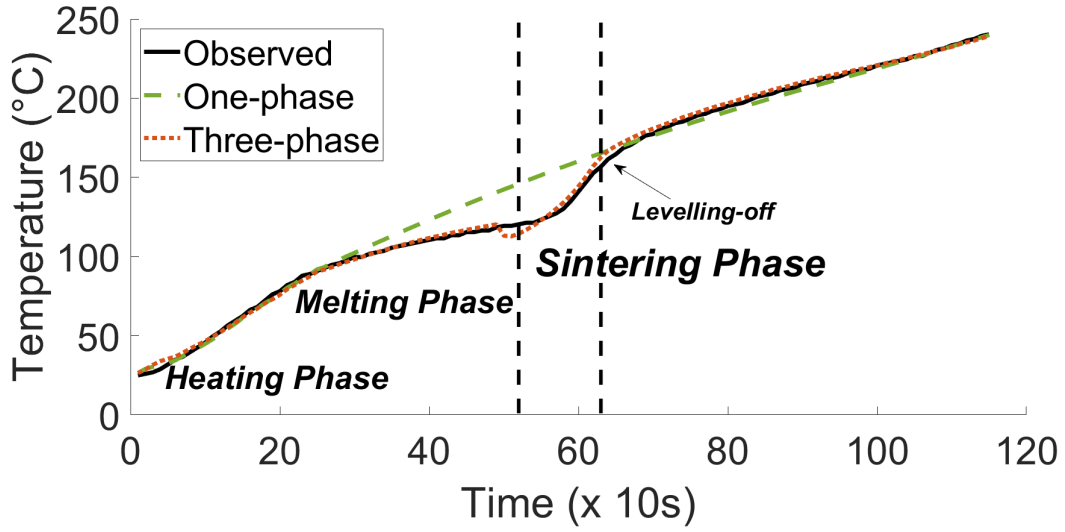

Figure 2.6: Mould temperature output profile observations and predictions of testing batch 10 for the 40% batch completion case. Predictions are shown for the one phase (- - - - -) and three phase (·····) models

Table 2.2: Model prediction comparison metrics for differing batch completion cases. The $\mathbf{q}_1$ and $\mathbf{q}_2$ quality variables represent the Sinkhole Area and Impact Strength, respectively. Note that the last column reports the number of times the model predicted quality values that are physically impossible. In practice, this is handled with control constraints. A better model makes less use of the constraints.

	Batch Completion	Temperature SSPE $(^\circ C)^2$	$\mathbf{q}_1$ SSPE (% area coverage)^2	$\mathbf{q}_2$ SSPE $(kg \cdot m)^2$	$\mathbf{q}_1$ and $\mathbf{q}_2$ predictions < 0
One-phase	5%	210,343	21.47	11.19	0
	20%	208,649	24.44	11.07	3
	40%	206,614	21.09	11.50	0
	60%	172,798	46.72	8.50	3
	80%	16,327	24.30	11.07	5
	TOTAL	814,730	138.03	53.33	11
Two-phase	5%	185,333	11.46	10.41	0
	20%	113,078	10.37	9.99	1
	40%	102,762	10.14	10.20	1
	60%	38,081	9.06	9.49	2
	80%	11,676	7.84	10.04	2
	TOTAL	450,929	48.87	50.13	6
	% improvement from one-phase	44.65	64.60	6.00	45.45
Three-phase	5%	113,496	6.16	10.26	1
	20%	50,741	6.58	9.75	1
	40%	48,136	6.91	10.13	0
	60%	20,549	6.65	9.97	1
	80%	9,655	6.72	9.36	2
	TOTAL	242,577	33.01	49.48	5
	% improvement from one-phase	70.23	76.08	7.22	54.54

strengths (figure 2.8) are plotted to compare for each of the 16 batches. For reference, figure 2.9 and figure 2.10 show the quality predictions plotted against the observed measurements. Although the multi-phase models do not always outperform the predictions of the one-phase model, it is generally the case that they do and that the three-phase model is best. This gives a theoretical basis to conduct observational testing with three-phase MPC formulations, which will help explore any model sensitivities and shortcomings in greater detail. In cases where the multi-phase models do not outperform the one-phase model, they do not fare much worse, whereas the one-phase model can be very unreliable as shown by the large incidences of error in figures

2.7 and 2.8 and the larger number of meaningless predictions in the last column of table 2.2. We call these predictions "meaningless" because it is physically impossible for quality measurements to be less than zero. MPC control constraints account for this model imprecision in practice, but better results can be attained if the model is less reliant on constraints.

The sinkhole area predictions improved much more than the impact strength predictions. The most likely reason for this is that industrial-standard tests are more accurate for determining sinkhole area coverage than impact strength. The impact strength is measured by dropping a falling dart from standardized intervals to calculate an average fail height. The more discrete nature of this test makes it less precise than the sinkhole area test, which is done through imaging. Furthermore, it is likely that the impact strength may depend more on the distribution of temperatures in the mould, which presently is not available as a measurement. Subsequent research will utilize temperature distribution information from thermal images to improve the quality prediction and control, and also more clearly demonstrate the advantage of using multi-phase models for impact strength control.

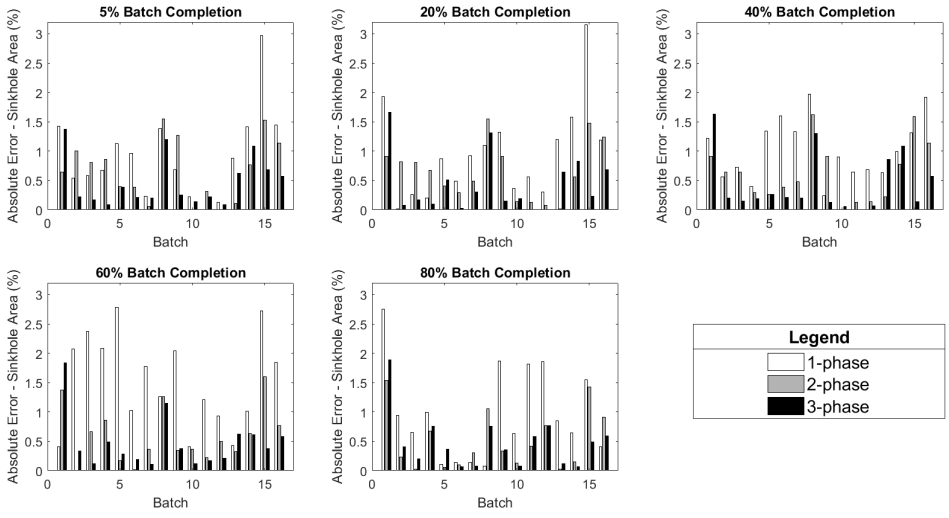

Figure 2.7: Absolute error between the actual and predicted sinkhole area coverage quality values ($\mathbf{q_1}$) for the one-phase, two-phase, and three-phase models of all 16 testing batches.

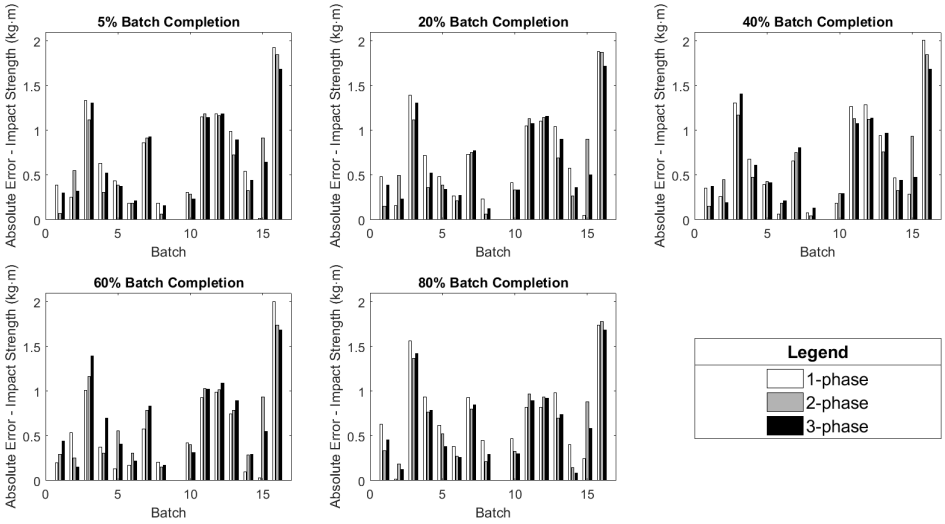

Figure 2.8: Absolute error between the actual and predicted impact strength quality values ($\mathbf{q_2}$) for the one-phase, two-phase, and three-phase models of all 16 testing batches. Note that there was no observed measurement for batch 9.

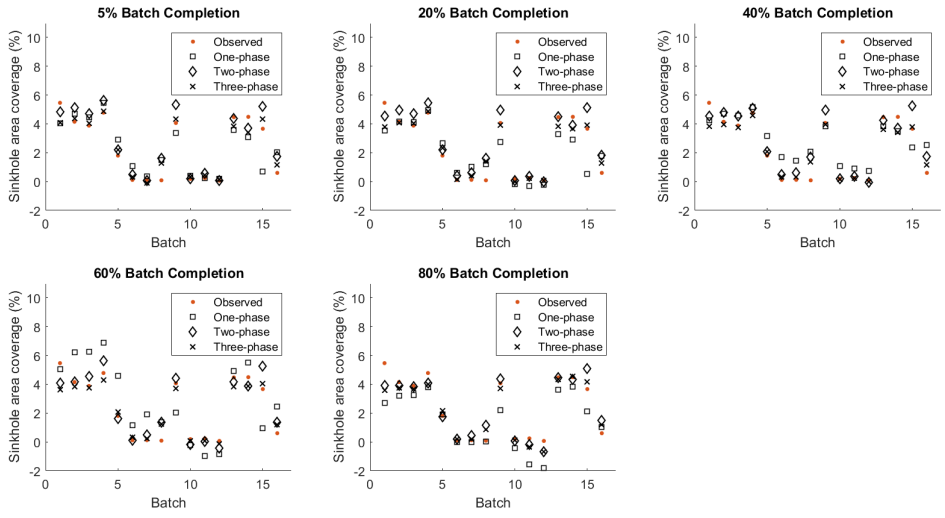

Figure 2.9: Sinkhole area quality values ($\mathbf{q_1}$) for the one-phase, two-phase, and three-phase models of all 16 testing batches plotted against the observed measurements.

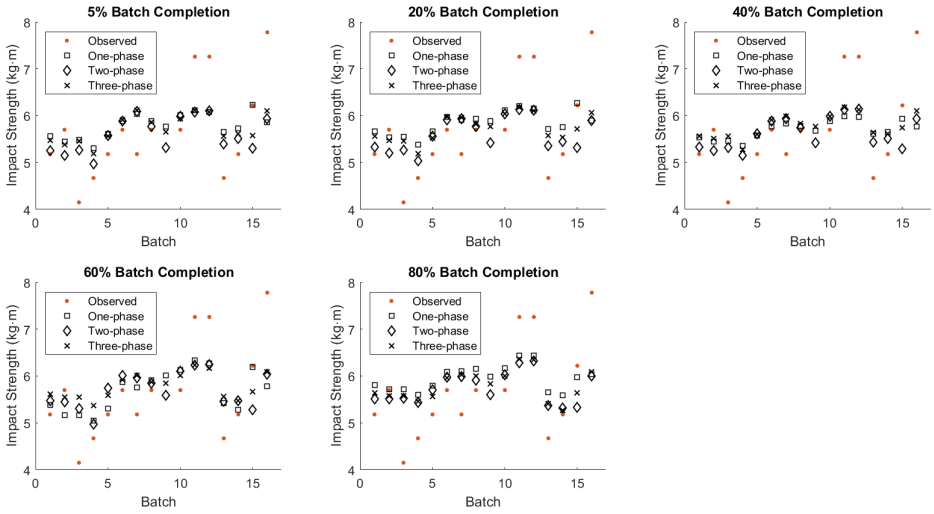

Figure 2.10: Impact strength quality values ($\mathbf{q_2}$) for the one-phase, two-phase, and three-phase models of all 16 testing batches plotted against the observed measurements. Note that there was no observed measurement for batch 9.

2.6 Conclusion

In this work, a new, MPSSID model was introduced to handle multi-phase batch processes. This model was tested on historical data from a uni-axial rotomoulding batch process. Optimal one-phase, two-phase, and three-phase models were determined from a training set and these models were then applied to a testing set for validation and comparison. It was shown that the the three-phase model best predicted the temperature trajectory of the process, which translated to product quality prediction improvements as well. An improvement in the predictive model for batch processes should result in more robust MPC formulations and make it easier to transition control schemes from lab to industrial sized processes.

2.7 Acknowledgement

The authors acknowledge the contribution of data for the purpose of analysis from Professor Michael Thompson's lab, and are grateful for the contributions from Hassan Abdulhussain, Vladimir Gritsichine and Aswin Chandrasekar. Financial support from the McMaster Advanced Control Consortium and the Ontario Graduate Scholarship Award is gratefully acknowledged.

Bibliography

[1] D. Bonvin, B. Srinivasan, and D. Hunkeler, "Control and optimization of batch processes," *IEEE Control Systems Magazine*, vol. 26, no. 6, pp. 34–45, 2006.

[2] D. Bonvin, "Optimal operation of batch reactors—a personal view," *Journal of Process Control*, vol. 8, no. 5, pp. 355 – 368, 1998. ADCHEM '97 IFAC Symposium: Advanced Control of Chemical Processes.

[3] S. Aumi and P. Mhaskar, "Integrating data-based modeling and nonlinear control tools for batch process control," *AIChE Journal*, vol. 58, no. 7, pp. 2105–2119, 2012.

[4] S. Aumi, B. Corbett, P. Mhaskar, and T. Clarke-Pringle, "Data-based modeling and control of nylon-6, 6 batch polymerization," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 1, pp. 94–106, 2013.

[5] B. W. Bequette, "Nonlinear control of chemical processes: a review," *Industrial & Engineering Chemistry Research*, vol. 30, no. 7, pp. 1391–1413, 1991.

[6] D. Shi, N. H. El-Farra, M. Li, P. Mhaskar, and P. D. Christofides, "Predictive control of particle size distribution in particulate processes," *Chemical Engineering Science*, vol. 61, no. 1, pp. 268 – 281, 2006. Advances in population balance modelling.

[7] M. Nayhouse, A. Tran, J. S.-I. Kwon, M. Crose, G. Orkoulas, and P. D. Christofides, "Modeling and control of ibuprofen crystal growth and size distribution," *Chemical Engineering Science*, vol. 134, pp. 414–422, 2015.

[8] J. Valappil and C. Georgakis, "Accounting for batch reactor uncertainty in the nonlinear mpc of end-use properties," *AIChE Journal*, vol. 49, no. 5, pp. 1178–1192, 2003.

[9] D. Bonvin and B. Srinivasan, "On the role of the necessary conditions of optimality in structuring dynamic real-time optimization schemes," *Computers & Chemical Engineering*, vol. 51, pp. 172–180, 2013. CPC VIII.

[10] M. Rafizadeh, R. Solgi, and M. Abbaszadeh, "Constrained model predictive control of mma polymerization reactor based on genetic algorithm optimization," in *Proceedings of 2003 IEEE Conference on Control Applications, 2003. CCA 2003.*, vol. 1, pp. 464–469 vol.1, 2003.

[11] S. Aumi and P. Mhaskar, "Robust model predictive control and fault handling of batch processes," *AIChE Journal*, vol. 57, no. 7, pp. 1796–1808, 2011.

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

[13] M. Nayhouse, J. Sang-Il Kwon, P. D. Christofides, and G. Orkoulas, "Crystal shape modeling and control in protein crystal growth," *Chemical Engineering Science*, vol. 87, pp. 216–223, 2013.

[14] R. Berber, "Control of batch reactors-a review (reprinted from methods of model based process control, 1995)," *Chemical engineering research & design*, vol. 74, no. 1, pp. 3–20, 1996.

[15] C. Undey and A. Cinar, "Statistical monitoring of multistage, multiphase batch processes," *IEEE Control systems magazine*, vol. 22, no. 5, pp. 40–52, 2002.

[16] K. Ogila, M. Shao, W. Yang, and J. Tan, "Rotational molding: A review of the models and materials.," *Express Polymer Letters*, vol. 11, no. 10, pp. 778–798, 2017.

[17] J. Sjöberg and M. Agarwal, "Trajectory tracking in batch processes using neural

controllers," *Engineering Applications of Artificial Intelligence*, vol. 15, no. 1, pp. 41 – 51, 2002.

[18] M. A. Hosen, M. A. Hussain, and F. S. Mjalli, "Control of polystyrene batch reactors using neural network based model predictive control (nnmpc): An experimental investigation," *Control Engineering Practice*, vol. 19, no. 5, pp. 454 – 467, 2011.

[19] J. Flores-Cerrillo and J. F. MacGregor, "Iterative learning control for final batch product quality using partial least squares models," *Industrial & Engineering Chemistry Research*, vol. 44, no. 24, pp. 9146–9155, 2005.

[20] J. Flores-Cerrillo and J. F. MacGregor, "Control of batch product quality by trajectory manipulation using latent variable models," *Journal of Process Control*, vol. 14, no. 5, pp. 539 – 553, 2004.

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

[22] P. Nomikos and J. F. MacGregor, "Multivariate spc charts for monitoring batch processes," *Technometrics*, vol. 37, no. 1, pp. 41–59, 1995.

[23] C. Ündey, S. Ertunç, and A. Çınar, "Online batch/fed-batch process performance monitoring, quality prediction, and variable-contribution analysis for diagnosis," *Industrial & Engineering Chemistry Research*, vol. 42, no. 20, pp. 4645–4658, 2003.

[24] T. Kourti, "Multivariate dynamic data modeling for analysis and statistical process control of batch processes, start-ups and grade transitions," *Journal of Chemometrics*, vol. 17, no. 1, pp. 93–109, 2003.

[25] J. F. Cerrillo and J. F. Macgregor, "Latent variable mpc for trajectory tracking in batch processes," *Journal of Process Control*, vol. 15, no. 6, pp. 651–663, 2005.

[26] H. Yu and J. Flores-Cerrillo, "Latent variable model predictive control for trajectory tracking in batch processes: Internal model control interpretation and design methodology," *Industrial & Engineering Chemistry Research*, vol. 52, no. 35, pp. 12437–12450, 2013.

[27] M. Golshan, J. F. MacGregor, M.-J. Bruwer, and P. Mhaskar, "Latent variable model predictive control (lv-mpc) for trajectory tracking in batch processes," *Journal of Process Control*, vol. 20, no. 4, pp. 538 – 550, 2010.

[28] M. Golshan, J. F. MacGregor, and P. Mhaskar, "Latent variable model predictive control for trajectory tracking in batch processes: alternative modeling approaches," *Journal of Process Control*, vol. 21, no. 9, pp. 1345 – 1358, 2011.

[29] J. Flores-Cerrillo and J. F. MacGregor, "Control of particle size distributions in emulsion semibatch polymerization using mid-course correction policies," *Industrial & Engineering Chemistry Research*, vol. 41, no. 7, pp. 1805–1814, 2002.

[30] B. Corbett, B. Macdonald, and P. Mhaskar, "Model predictive quality control of polymethyl methacrylate," *IEEE Transactions on Control Systems Technology*, vol. 23, pp. 687–692, March 2015.

[31] J. Wan, O. Marjanovic, and B. Lennox, "Disturbance rejection for the control of batch end-product quality using latent variable models," *Journal of Process Control*, vol. 22, no. 3, pp. 643–652, 2012.

[32] A. Kassidas, J. F. MacGregor, and P. A. Taylor, "Synchronization of batch trajectories using dynamic time warping," *AIChE Journal*, vol. 44, no. 4, pp. 864–875, 1998.

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

[34] P. Van Overschee and B. De Moor, "N4sid: Subspace algorithms for the identification of combined deterministic-stochastic systems," *Automatica*, vol. 30, no. 1, pp. 75–93, 1994. Special issue on statistical signal processing and control.

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

[36] R. Shi and J. F. MacGregor, "Modeling of dynamic systems using latent variable and subspace methods," *Journal of Chemometrics*, vol. 14, no. 5-6, pp. 423–439, 2000.

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

[38] B. Corbett and P. Mhaskar, "Subspace identification for data-driven modeling and quality control of batch processes," *AIChE Journal*, vol. 62, no. 5, pp. 1581–1601, 2016.

[39] B. Corbett and P. Mhaskar, "Data-driven modeling and quality control of variable duration batch processes with discrete inputs," *Industrial & Engineering Chemistry Research*, vol. 56, no. 24, pp. 6962–6980, 2017.

[40] A. Garulli, S. Paoletti, and A. Vicino, "A survey on switched and piecewise affine system identification," *IFAC Proceedings Volumes*, vol. 45, no. 16, pp. 344–355, 2012.

[41] M. J. Oliveira, M. C. Cramez, and R. J. Crawford, "Structure-properties relationships in rotationally moulded polyethylene," *Journal of Materials Science*, vol. 31, no. 9, pp. 2227–2240, 1996.

[42] F. P. Gomes, A. Garg, P. Mhaskar, and M. R. Thompson, "Data-driven advances in manufacturing for batch polymer processing using multivariate nondestructive monitoring," *Industrial & Engineering Chemistry Research*, vol. 58, no. 23, pp. 9940–9951, 2019.

[43] D. I. Abu-Al-Nadi, D. I. Abu-Fara, I. Rawabdeh, and R. J. Crawford, "Control of rotational molding using adaptive fuzzy systems," *Advances in Polymer Technology*, vol. 24, no. 4, pp. 266–277, 2005.

[44] A. Hamidi, S. Farzaneh, F. Nony, Z. Ortega, S. Khelladi, M. Monzon, F. Bakir, and A. Tcharkhtchi, "Modelling of sintering during rotational moulding of the thermoplastic polymers," *International Journal of Material Forming*, vol. 9, no. 4, pp. 519–530, 2016.

[45] A. Garg, F. P. Gomes, P. Mhaskar, and M. R. Thompson, "Model predictive control of uni-axial rotational molding process," *Computers & Chemical Engineering*, vol. 121, pp. 306–316, 2019.

[46] A. Garg, H. A. Abdulhussain, P. Mhaskar, and M. R. Thompson, "Handling constraints and raw material variability in rotomolding through data-driven model predictive control," *Processes*, vol. 7, no. 9, 2019.

[47] L. Ljung, *System Identification Theory For The User*. Prentice Hall information and system sciences series, Prentice Hall PTR, 2nd ed., 1999.

[48] P. V. Overschee and B. D. Moor, "A unifying theorem for three subspace system identification algorithms," *Automatica*, vol. 31, no. 12, pp. 1853 – 1864, 1995. Trends in System Identification.

[49] K. R. Muske, J. B. Rawlings, and J. H. Lee, "Receding horizon recursive state estimation," *1993 American Control Conference*, pp. 900–904, 1993.

[50] S. Wold, M. Sjöström, and L. Eriksson, "Pls-regression: a basic tool of chemometrics," *Chemometrics and intelligent laboratory systems*, vol. 58, no. 2, pp. 109–130, 2001.

Chapter 3

Model Predictive Control Using Subspace Model Identification

The previous chapter presented a new subspace identification modelling approach that can be used for multi-phase processes. In this chapter, alternate formulations of this approach are explored and compared. The multi-phase model is also compared to a subspace "reidentification" method that has already been implemented experimentally with promising results. Finally, a model predictive control (MPC) optimization program is presented for use with Multi-Phase Subspace Identification Models.

3.1 Abstract

This chapter presents a comparative analysis for different strategies of implementing Multi-Phase Subspace Identification (MPSSID) and presents a mixed integer linear program to integrate MPSSID into model predictive control (MPC). In some cases, the Multi-Phase model has difficulty maintaining stability at the phase transition point. Implementing a multi-phase subspace model is only useful if the additional model complexity does not jeopardize the model's ability to stay within operating control limits. Models can be kept within control limits by adding constraints to the control program, but better performances can be achieved when the underlying model is sound. Results demonstrate the superior predictive capabilities of multi-phase models and outline limitations to be considered in model construction. A MPC framework for experimental implementation is presented.

3.2 Introduction

Batch processes are popular in the food, pharmaceutical, and specialty chemical industries due to their ability to produce products on-demand, their ability to isolate failures to a single batch, and their suitability for low-volume processes where the final product quality is important [1]. Black-box, data-driven models are becoming increasingly popular in the control of batch processes. Technological advances have improved computational speeds and sensor capabilities so that vast amounts of process data can now be tracked with inexpensive sensors [2, 3]. This includes ARX models [4], Partial Least Squares (PLS) models [5, 6, 7], Neural Networks [8, 9], and Subspace Identification (SSID) models [10, 11, 12, 13, 14].

SSID models are valued for their ability to track the state of the process, which

allows them to better reject process disturbances than traditional trajectory tracking approaches. The calculation of the process state also makes it easy to integrate SSID models into MPC programs [15, 16]. An important consideration for SSID model applications is to investigate whether or not the nonlinearities of the process warrant additional modelling. In [17], a multi-phase subspace identification (MPSSID) model was introduced to address the nonlinearity of multi-phase processes. The predictive capabilities of the model were tested using historical data from a rotational moulding process. It was shown that the multi-phase models outperformed the one-phase model; however, further investigation was required to improve the stability at the phase switching points. In this chapter, alternative methods of MPSSID implementation are explored before an optimization program is introduced to use MPSSID within MPC.

3.3 MPSSID Formulation Analysis

In [17], the predictive capability of the one-phase, two-phase, and three-phase MPSSID models was compared using inputs that had already been determined from experimental runs. The authors plugged these inputs into a one-phase, two-phase, and three-phase model simulation and compared the model-predicted outputs to those that had actually been observed. This was done for varying levels of "batch completion:" 5%, 20%, 40%, 60%, and 80%. "Batch completion" referred to the amount of time before predictions were made without the use of an observer; the observer helps bring the state of the system closer to its true state. This simulates what occurs during an experiment. During an experiment, the next input values set by the controller are chosen by an optimization program that is dependent on the predicted final product quality, which is determined by the *final* predicted state of the process. Every time an output measurement is taken, the controller determines the next input by predicting

a state trajectory all the way to the end of the batch and then using that to generate a predicted product quality. At the beginning of an experiment only a few output measurements are available, and so many model predictions must be made without the use of an observer. Closer to the end of an experiment, many output measurements are available and so less observer-free predictions are required. A model that is better at predicting the trajectory of the batch without an observer should perform better.

The limitation to this analysis is that it is only a retrospective theoretical test. Instead of building controllers based on the different SSID models and comparing the final product quality that is obtained from doing actual experiments, this analysis simulates the ability of each model to accurately predict the result of experiments that already occurred. It does not test the decision-making ability of the new models if they were to be implemented into the experimental control scheme. Still, it is a good way to determine if new methods are worth attempting experimentally so that material is not wasted.

This analysis was performed on a set of historical batches that were semi-randomly divided into a representative set of 39 training batches and 16 testing batches. The randomization was done semi-randomly because three different MPC programs were used for the 55 experiments. First, batches were separated into three groups based on the MPC program that was used. Batches from these groups were randomly assigned to the training or testing set. The following sections make comparisons by utilizing the same sets of training and testing batches.

3.3.1 Original MPSSID Formulation: Adjusting Inputs

In [17], a comparison was made between the temperature and quality predictions of a one-phase, two-phase, and three-phase subspace identification (SSID) model. The

heater temperatures of the process were used as the process inputs for this analysis, while the internal mould temperature was the process output. In the experimental setup, the heater temperature measurements were taken from thermocouples placed on either side of an oven that was placed over the mould to heat it. PID feedback control can be used to bring the heater temperatures to selected setpoints. It is possible to use the heater temperature readings within an MPC control system; however, in practice the experiment was set up for MPC control by utilizing the heater powers and air compressor power as the controlled inputs to the process based on the feedback from the internal mould temperature as the output. In table 3.1 a comparison between the one-phase and three-phase models is presented for the heater power and heater temperature input cases. The results demonstrate the superior performance of the heater temperature case over the heater power case. They also show that in both cases, the three-phase model outperforms the one-phase model. One additional note is that the heater temperature results differ slightly from the previous work. In [17], the observer poles were selected by tracking the convergence of the outputs for validation data and then selecting pole values within the unit circle that produced stable responses to the state estimates. In this work, pole values were selected by first running the input/output data through the "tfest" MATLAB function, which can be found in the System Identification Toolbox [18]. Real number pole estimates that were within the unit circle were used. This method of pole selection was a more standardized approach that allowed for a more objective comparison between methods.

For the heater power case, the improved performance of the three-phase model comes with a caveat (denoted with a * in table 3.1). For the three-phase case, an extra constraint was needed to ensure the phase switch would occur for the batch completion scenarios where many observer-free predictions are required (e.g. 5%, 20%). It was observed that for some of the test batches, the predicted outputs never reached the phase-switching temperature. In some cases the predicted temperature grew very close

Table 3.1: One-phase and Three-phase heater temperature inputs and heater power inputs model prediction comparison metrics for differing batch completion scenarios. The $\mathbf{q}_1$ and $\mathbf{q}_2$ quality variables represent the Sinkhole Area and Impact Strength, respectively. Note that the last column reports the number of times a model predicted quality values that are physically impossible. In practice, this is handled with control constraints. A better model makes less use of the constraints.

	Batch Completion	Temperature SSPE $(^{\circ}C)^2$	$\mathbf{q}_1$ SSPE (% area coverage)^2	$\mathbf{q}_2$ SSPE $(kg \cdot m)^2$	$\mathbf{q}_1$ and $\mathbf{q}_2$ predictions < 0
Heater Powers One-Phase	5%	472,626	106.98	14.66	6
	20%	1,168,434	234.12	24.33	8
	40%	645,036	194.80	16.94	3
	60%	257,937	208.51	12.93	7
	80%	14,478	31.45	9.89	6
	TOTAL	2,558,511	775.86	78.75	30
Heater Powers Three-Phase*	5%	1,026,381	186.31	14.63	5
	20%	794,535	154.36	12.73	6
	40%	204,527	44.09	8.70	3
	60%	39,366	26.74	8.78	3
	80%	10,174	28.14	8.58	6
	TOTAL	2,074,982	439.64	53.42	23
	% improvement from one-phase	18.90	43.34	32.17	23.33
Heater Temperatures One-Phase	5%	209,718	23.85	10.85	0
	20%	208,978	27.25	10.71	3
	40%	208,227	22.98	11.21	0
	60%	165,676	48.76	8.05	3
	80%	12,875	26.92	10.67	5
	TOTAL	805,474	149.75	51.50	11
	% improvement from Heater Powers	68.52	80.70	34.61	63.33
Heater Temperatures Three-Phase	5%	110,354	6.12	10.22	0
	20%	46,731	6.08	10.02	0
	40%	44,352	6.72	10.14	0
	60%	18,859	6.63	10.03	1
	80%	8,094	6.76	9.41	2
	TOTAL	228,390	32.31	49.82	3
	% improvement from one-phase	71.65	78.42	3.26	72.73
	% improvement from Heater Powers	88.99	92.65	6.74	86.96

to the phase switching point, but then it started to decrease. This caused an error in the code as the model never progressed to the next phase. To correct for this, the phase switch was initiated automatically if the output predictions began to plateau. In figure 3.1 the outputs of testing batch 15 are plotted for the unconstrained and

constrained case to illustrate an example of this occurrence. The outputs for testing batch 15 are shown in figure 3.2.

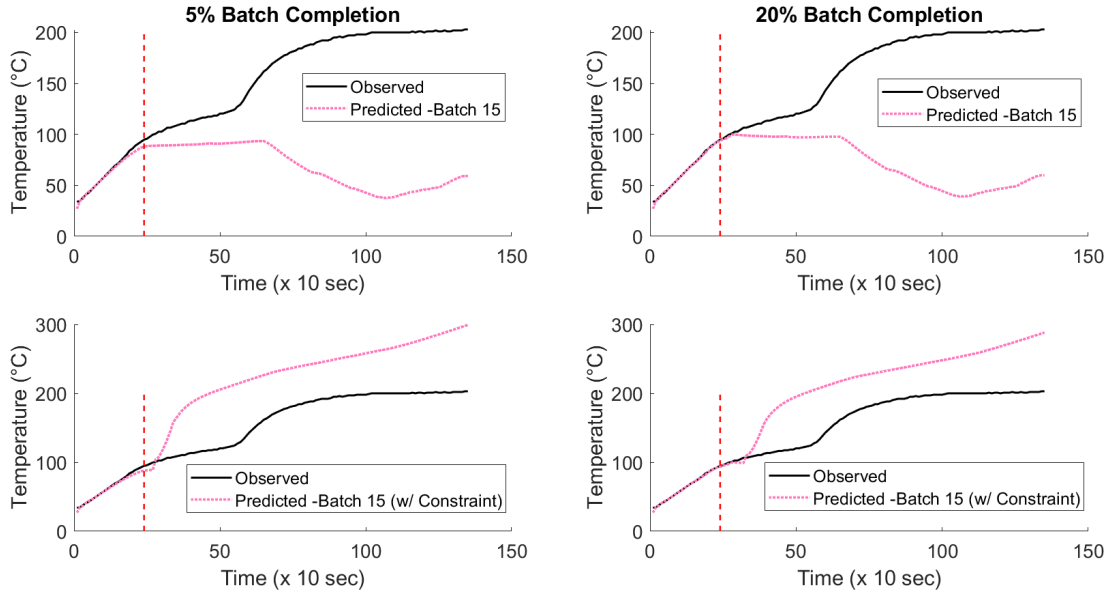

Figure 3.1: Predicted and observed output profiles for testing batch 15 in the constrained (top) and unconstrained (bottom) cases. The predictions are shown for the 5% and 20% batch completion scenarios. For the 40%, 60%, and 80% cases, the observer is left on long enough to avoid the error. The vertical red line indicates the time where the control was switched from PID to MPC for comparison with figure 3.2.

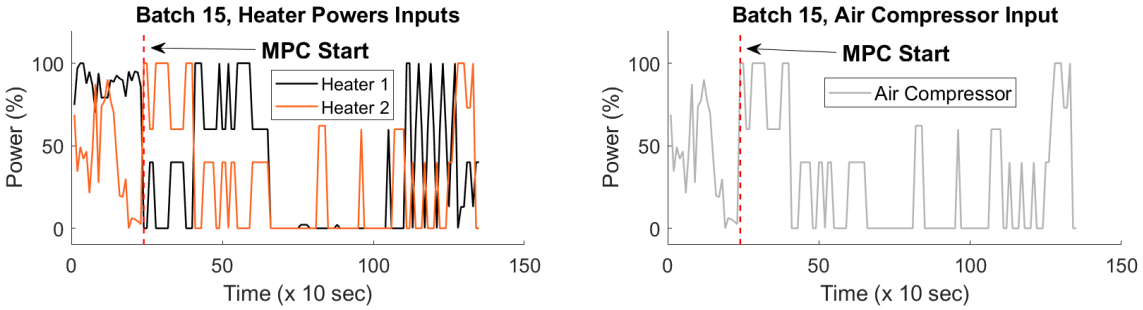

Figure 3.2: Heater power (left) and air compressor (right) input profiles for testing batch 15.

The main cause for this issue seemed to be the sensitivity of the multi-phase model to power input readings of 0%. The test batches where the errors occurred started with PID control to generate a good state estimate before switching to an "Economic" MPC

optimization program. This means the control objective was to minimize the total energy usage during an experiment, and so the controller was pressured to keep the heater powers at 0% as often as possible. Figures 3.1 and 3.2 show how the predicted temperature starts to decrease once the MPC is implemented. In practice, this may not affect a multi-phase implementation. A multi-phase MPC incorporates the phase switch into its calculation, so it would not select input values that do not enable the switch to occur. A multi-phase optimization program for MPC is introduced in subsection 3.5.2.

Making use of the heater temperatures within MPC control would require an indirect implementation. First, each MPC calculation would produce a new temperature setpoint for the two heaters. The PID controllers tuned to the heater temperatures would then determine the heater and air compressor powers to bring the heaters to their new setpoint. When the MPC directly controls the heater powers, the PID controllers are not used. A diagram of this difference is shown in figure 3.3.

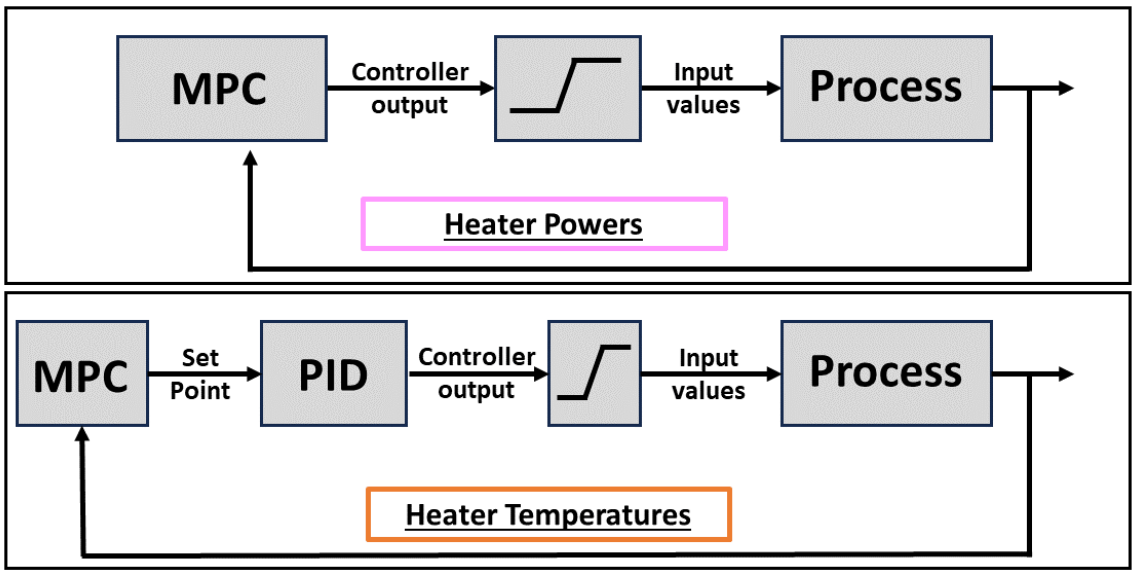

Figure 3.3: The difference between the experimental setup with heater powers as the MPC inputs (top) and the proposed setup using heater temperatures as the inputs to the MPC (bottom).

One reason that may explain the improved performance of the heater temperature input case is that using the temperatures as the inputs may capture the physical relationships of the process more effectively. Conductive, convective, and radiant heat transfer occur during the rotomoulding process. All three are based on relationships with the temperature gradient (linear for conduction and convection, fourth order for radiant heating). The heater temperatures may do a better job approximating these relationships and their impact on the internal mould temperature than the heater powers do.

3.3.2 Propagation-Free MPSSID

To start, a quick recap of the MPSSID formulation in [17] is presented.

The identifying Hankel matrices take the following form for each phase of each batch (note that only the output matrices are shown, but the structure of the input Hankel matrices are the exact same):

$$\mathbf{Y}_{1|i}^{b,p} = \begin{bmatrix} \mathbf{y}^{b,p}[1] & \mathbf{y}^{b,p}[2] & \cdots & \mathbf{y}^{b,p}\left[j^{b,p}\right] \\ \vdots & \vdots & & \vdots \\ \mathbf{y}^{b,p}[i^p] & \mathbf{y}^{b,p}[i^p+1] & \cdots & \mathbf{y}^{b,p}\left[i^p+j^{b,p}-1\right] \end{bmatrix} \tag{3.1}$$

$$\mathbf{Y}_{i+1|2i}^{b,p} = \begin{bmatrix} \mathbf{y}^{b,p}[i+1] & \mathbf{y}^{b,p}[i+2] & \cdots & \mathbf{y}^{b,p}\left[i+j^{b,p}\right] \\ \vdots & \vdots & & \vdots \\ \mathbf{y}^{b,p}[2i] & \mathbf{y}^{b,p}[2i+1] & \cdots & \mathbf{y}^{b,p}\left[2i+j^{b,p}-1\right] \end{bmatrix} \tag{3.2}$$

Where $\mathbf{y}^{b,p}\left[N^{b,p}\right] = \mathbf{y}^{b,p}\left[2i+j^{b,p}-1\right]$

Next, the submatrices are horizontally concatenated:

$$\mathbf{Y}_{1|i}^{p} = \begin{bmatrix} \mathbf{Y}_{1|i}^{1,p} & \mathbf{Y}_{1|i}^{2,p} & \cdots & \mathbf{Y}_{1|i}^{B,p} \end{bmatrix} \tag{3.3}$$

When the SSID is done for each phase it results in P sets of system matrices: $\mathbf{A}^1$, $\mathbf{B}^1$, $\mathbf{C}^1$, $\mathbf{D}^1$... $\mathbf{A}^P$, $\mathbf{B}^P$, $\mathbf{C}^P$, $\mathbf{D}^P$. It also results in P sets of identified state sequences for each training batch,

$$\mathbf{X}_{i+1}^{b,p} = \begin{bmatrix} \mathbf{x}^{b,p}[i+1] & \mathbf{x}^{b,p}[i+2] & \cdots & \mathbf{x}^{b,p}\left[i^p + j^{b,p}\right] \end{bmatrix} \tag{3.4}$$

The next step is to calculate the first and last states of each training batch by propagating through the first i^p states and last $i^p - 1$ states. These states are required for the PLS models that must be set up to connect phases, but the SSID regression mechanism does not calculate them as they are sacrificed to calculate the system states. In [17], the last $i^p - 1$ values were calculated using the identified system matrices:

$$\mathbf{x}^{b,p}\left[i^p + j^{b,p} + \kappa\right] = \mathbf{A}^p \mathbf{x}^{b,p}\left[i^p + j^{b,p}\right] + \mathbf{B}^p \mathbf{u}^{b,p}\left[i^p + j^{b,p}\right] \quad \forall \kappa \in 1, 2, ..., i^p - 1 \tag{3.5}$$

The first i^p values require backwards propagation to be calculated. In [17], this was done using a linear optimization program. This could also be done without the use of an optimization program by making use of the inverse system matrices; however, it was found that this method gave highly unstable state estimates which are troublesome for the PLS steps that are used to connect phases. The structure of the following optimizer is presented in [11].

$$\min_{\mathbf{x}^{b,p+1}[k]} \left(\mathbf{y}^{b,p+1}[k] - \hat{\mathbf{y}}^{b,p+1}[k]\right) \mathbf{R} (\mathbf{y}^{b,p+1}[k] - \hat{\mathbf{y}}^{b,p+1}[k]) + \omega_k \mathbf{Q} \omega_k \quad \forall k \in i^p, i^p - 1, ..., 1 \tag{3.6}$$

subject to

$$\omega_k = \mathbf{x}^{b,p+1}[k+1] - \left(\mathbf{A}^{p+1}\mathbf{x}^{b,p+1}[k] + \mathbf{B}^{p+1}\mathbf{u}^{b,p+1}[k]\right) \tag{3.7}$$

$$\hat{\mathbf{y}}^{b,p+1}[k] = \mathbf{C}^{p+1}\mathbf{x}^{b,p+1}[k] + \mathbf{D}^{p+1}\mathbf{u}^{b,p+1}[k] \tag{3.8}$$

In [17], it was observed that even though the optimization program improved state stability during backwards propagation, and thus improved the stability of the PLS step that is used to switch from one phase to the next, there was still a tendency for the output predictions to "jump" during the phase switch. Figures 3.4, 3.5, and 3.6 illustrate this. In figure 3.4, the 3-phase MPSSID model predictions are shown for the heater temperature input case. Testing batches 3 and 9 are shown for the 5%, 20%, and 40% batch completion scenarios. Figure 3.5 shows this for the heater power input case. The temperature jumps downward when the switch from phase 1 to phase 2 occurs (in the neighbourhood of the 60th observation point). The model does better in the heater temperature inputs case, but the jump still occurs. These figures also show how the prediction improves from left to right as the model is able to make more predictions with the help of an observer. Figure 3.6 shows the 80% completion scenario for testing batch 9. The output prediction jumps at the phase switch before the observer helps it converge back to the observed trajectory. In practice, it is not a problem if the observer jumps for cases when the batch has progressed past the phase switch since the observer helps the output quickly converge again.

One idea to further improve the stability at the phase switch is to adjust the indexing of the phase data to include i^p values before the start of the phase and $i^p - 1$ values

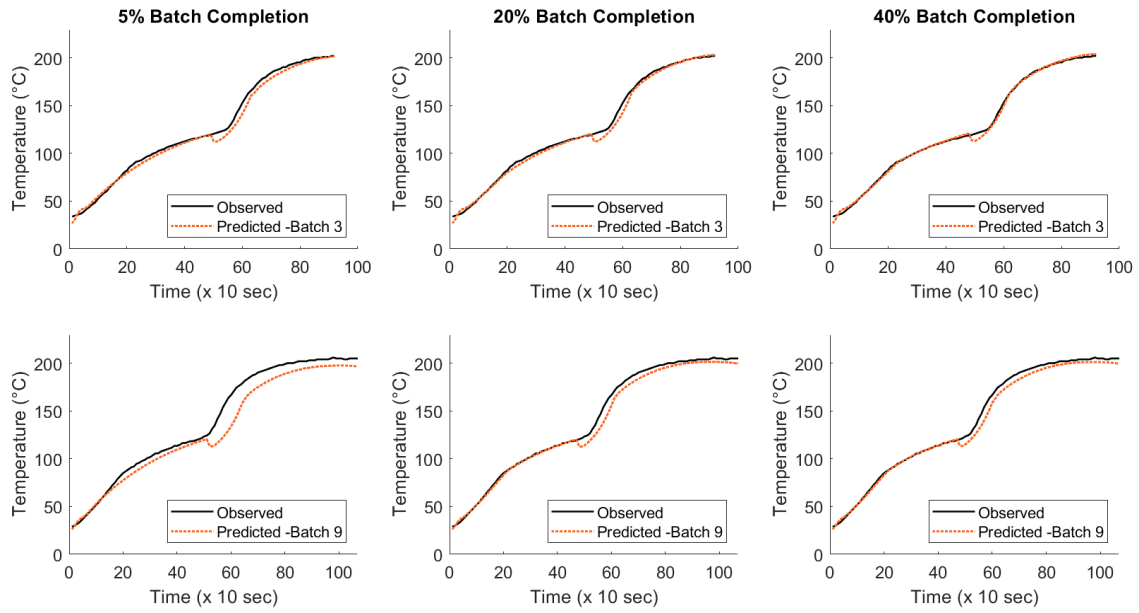

Figure 3.4: Predicted outputs for testing batches 3 and 9 for the heater temperature input case. The jumps are most evident in the 5% completion scenario where the most predictions are made without the help of an observer.

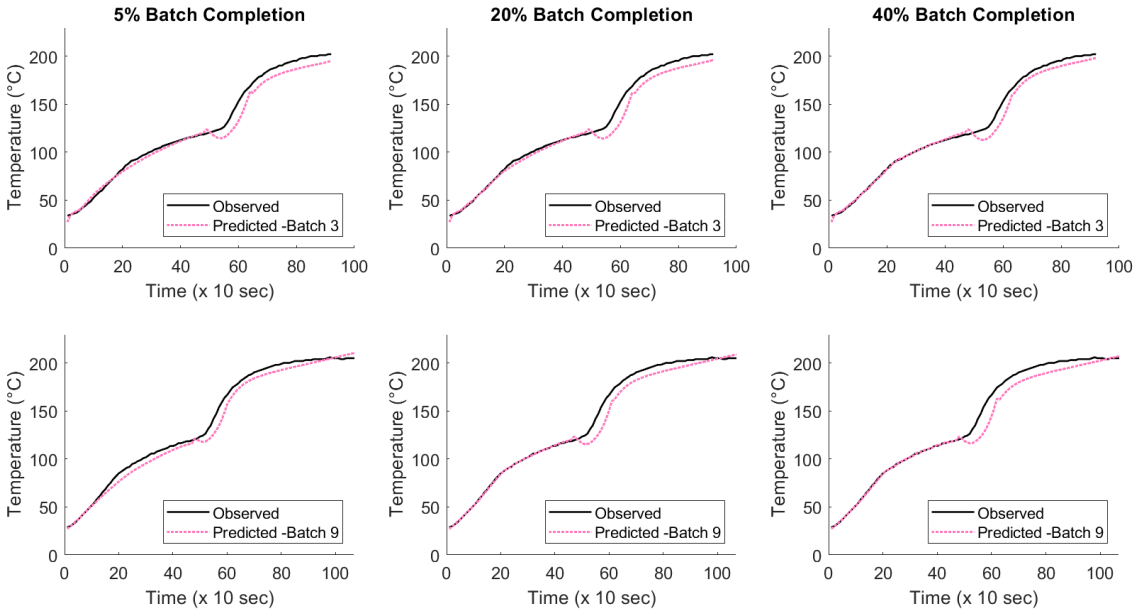

Figure 3.5: Predicted outputs for testing batches 3 and 9 for the heater power input case. The jumps are generally more evident when the heater powers are used as inputs than they are when the heater temperatures are used.

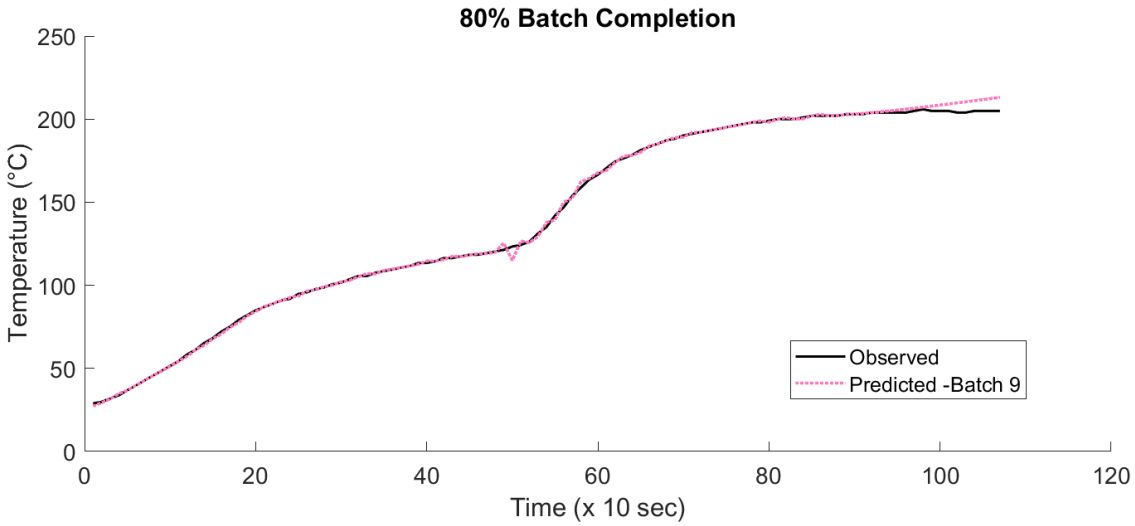

Figure 3.6: 80% batch completion scenario using the heater powers as inputs for batch 9 of the testing set. Note the jump that occurs between the 40th and 60th observation. The observer helps the output prediction converge back to the observed trajectory.

after the end of the phase. This way, the states at the phase switch points are directly calculated when the SSID step is performed and there is no need for additional forward or backward propagation. Tables 3.2 and 3.3 show the batch data that is included in each phase for standard MPSSID, while tables 3.4 and 3.5 show the extra data points that have to be included so that the values of $\mathbf{y}^{b,p}\left[N^{b,p}\right]$ and $\mathbf{y}^{b,p+1}[1]$ are calculated directly from the SSID step. Note that the same indexing is used for the input data.

The difference between this method and the original is that more data points are attributed to two phases. In the original formulation, only the data point where the phase switch occurs exists in the Hankel matrices of two phases. The drawback to the propagation-free method is that values beyond the scope of the phase must be included in the calculation of the system matrices. Note that the vertical lines in tables 3.2, 3.3, 3.4, and 3.5 are used to indicate the observation that connects one phase to the next.

Table 3.2: Previous phase partitioning: first and last phase.

Phase 1	$\mathbf{y}^{b,1}[1] \quad \mathbf{y}^{b,1}[2] \quad \cdots$	$\mathbf{y}^{b,1}\left[N^{b,1}\right]$	
Phase P		$\mathbf{y}^{b,P}[1]$	$\mathbf{y}^{b,P}[2] \quad \cdots \quad \mathbf{y}^{b,P}\left[N^{b,P}\right]$

Table 3.3: Previous phase partitioning for phase $p \mid 1 < p < P$.

Phase p		$\mathbf{y}^{b,p}[1]$	$\cdots \quad \cdots$	$\mathbf{y}^{b,p}\left[N^{b,p}\right]$	

Table 3.4: Proposed phase partitioning to bypass the need for propagation: first and last phase.

Phase 1	$\mathbf{y}^{b,1}[1] \quad \cdots \quad \cdots \quad \cdots$	$\mathbf{y}^{b,1}\left[N^{b,1}\right]$	$\cdots \quad \mathbf{y}^{b,1}\left[N^{b,1}+i-1\right]$
Phase P	$\mathbf{y}^{b,P}[1-i] \quad \cdots$	$\mathbf{y}^{b,P}[1]$	$\cdots \quad \cdots \quad \cdots \quad \mathbf{y}^{b,P}\left[N^{b,P}\right]$

Table 3.5: Proposed phase partitioning to bypass the need for propagation for phase $p \mid 1 < p < P$.

Phase p	$\mathbf{y}^{b,p}[1-i] \quad \cdots$	$\mathbf{y}^{b,p}[1]$	$\cdots \quad \cdots$	$\mathbf{y}^{b,p}\left[N^{b,p}\right]$	$\cdots \quad \mathbf{y}^{b,p}\left[N^{b,p}+i-1\right]$

Note that for a two-phase identification, tables 3.3 and 3.5 are not needed. When the identification is done for three or more phases, the middle phases require an extra i values to be included at the front of the phase identification *and* an extra $i-1$ phases to be included at the end of the phase as shown in tables 3.3 and 3.5. This is because the middle phases are connected to two instances of phase switching, whereas the first and last phases are only connected to one. A more visual representation of this is shown for a sample batch in figures 3.7 and 3.8. In these figures, the rectangular bars for each of the phases represent the entire set of batch data. The enclosed boxes show

the data that pertains to each phase based on the output criteria used to separate the phases (one overlap point where the phase switch occurs). The dashed lines indicate the amount of data that is used in more than one phase. The brace brackets show the data that is sacrificed to determine the states of each phase in the SSID regression step and so must be calculated through propagation after the states have been identified.

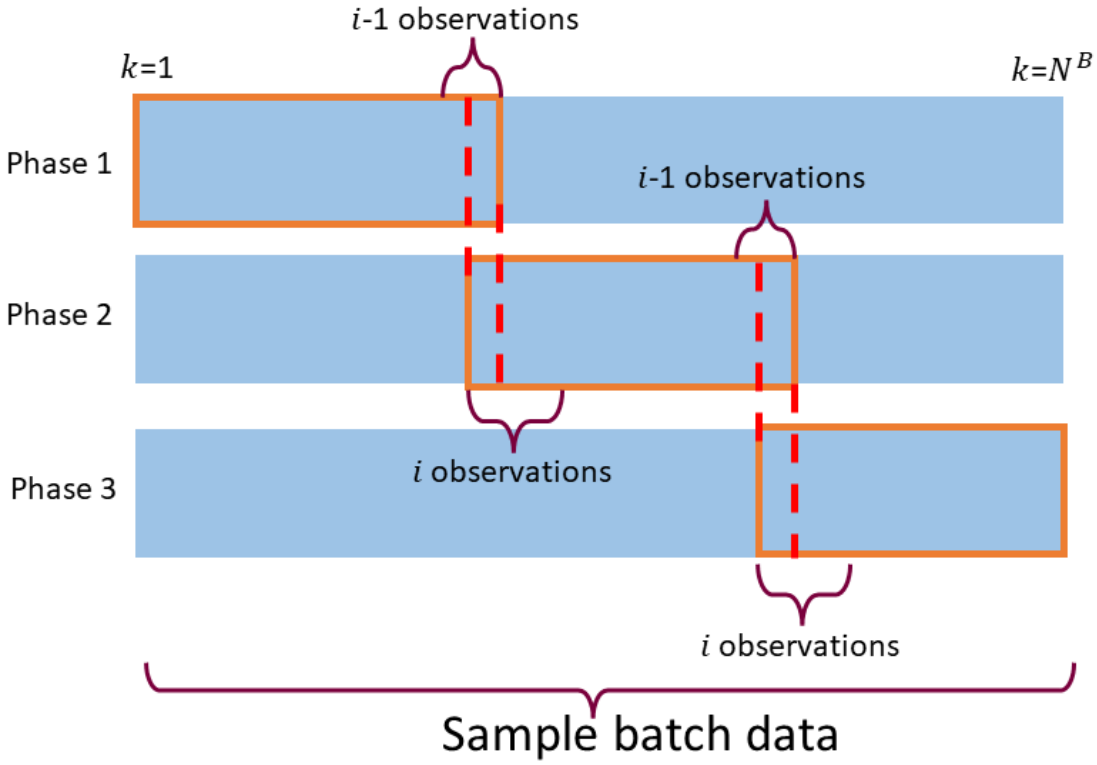

Figure 3.7: Visual representation showing how the data for a sample batch is grouped into phases for the propagation case. Note N^B represents the number of times at which observations are taken for the batch.

A comparison between the original MPSSID and the propagation-free version is shown in table 3.6. These results show that the propagation-free model does not outperform the original multi-phase formulation, especially for the metric for quality predictions less than 0. Including more data points that go beyond the scope of the phase has a negative impact on the predictive capability of the model. Visually though, the "jumping" effect was observed to disappear. This is shown in figure 3.9. These results

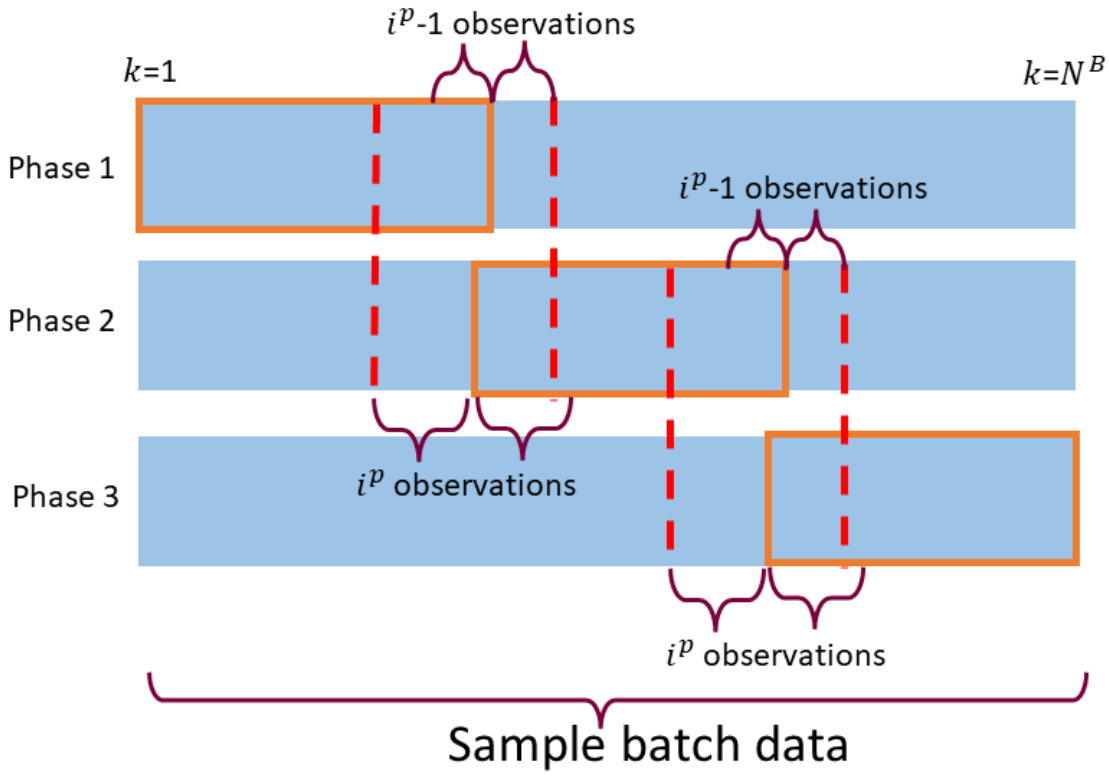

Figure 3.8: Visual representation showing how the data for a sample batch is grouped into phases for the *propagation-free* case. Note N^B represents the number of times at which observations are taken for the batch.

demonstrate that the instability of the states during the backwards propagation step is a key factor in producing the jumping effect that is observed during the phase switching in the MPSSID model. The propagation-free approach yields worse quality predictions, and so a better approach to solve this problem may be to experiment with the back-propagation optimizer described in equations 3.6, 3.7, and 3.8 to obtain a more stable result.

Table 3.6: Original three-phase, and Propagation-Free three-phase model prediction comparison metrics for differing batch completion scenarios. Heater temperatures were used as the inputs. The $\mathbf{q}_1$ and $\mathbf{q}_2$ quality variables represent the Sinkhole Area and Impact Strength, respectively. Note that the last column reports the number of times a model predicted quality values that are physically impossible.

	Batch Completion	Temperature SSPE $(^\circ C)^2$	$\mathbf{q}_1$ SSPE (% area coverage)^2	$\mathbf{q}_2$ SSPE $(kg \cdot m)^2$	$\mathbf{q}_1$ and $\mathbf{q}_2$ predictions < 0
Heater Temperatures Three-Phase	5%	110,354	6.12	10.22	0
	20%	46,731	6.08	10.02	0
	40%	44,352	6.72	10.14	0
	60%	18,859	6.63	10.03	1
	80%	8,094	6.76	9.41	2
	TOTAL	228,390	32.31	49.82	3
Heater Temperatures Propagation-Free	5%	89,572	7.74	10.54	5
	20%	49,843	7.54	10.20	4
	40%	51,045	8.51	10.69	4
	60%	17,360	7.13	9.42	4
	80%	5,966	6.96	9.60	4
	TOTAL	213,786	37.89	50.45	21
	% improvement from original	6.39	-17.26	-1.27	-600

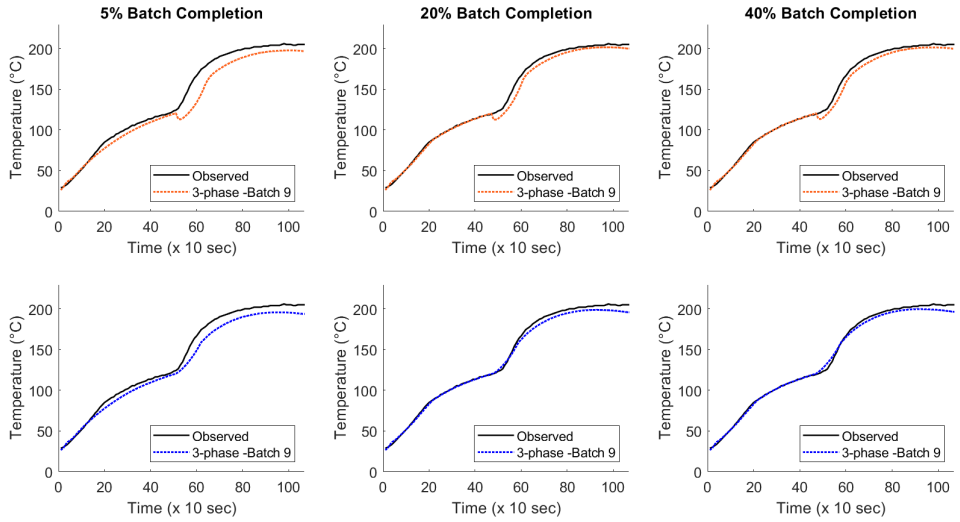

Figure 3.9: Predicted outputs for testing batch 9 for original MPSSID case (top, orange) and propagation-free model (bottom, blue). The jump is observed to disappear when the propagation-free model is used, but the quality predictions are less accurate.

3.3.3 Manual Phase Partitioning

Another method that was used to try and address the phase switch jumping was to organize the phase data for the training batches by visual inspection. Instead of determining the start and the end of a phase based on a chosen output temperature, the start and end were chosen manually by selecting the observations located closest to the inflection points observed on the graph. These inflection points are shown visually in figure 3.10.

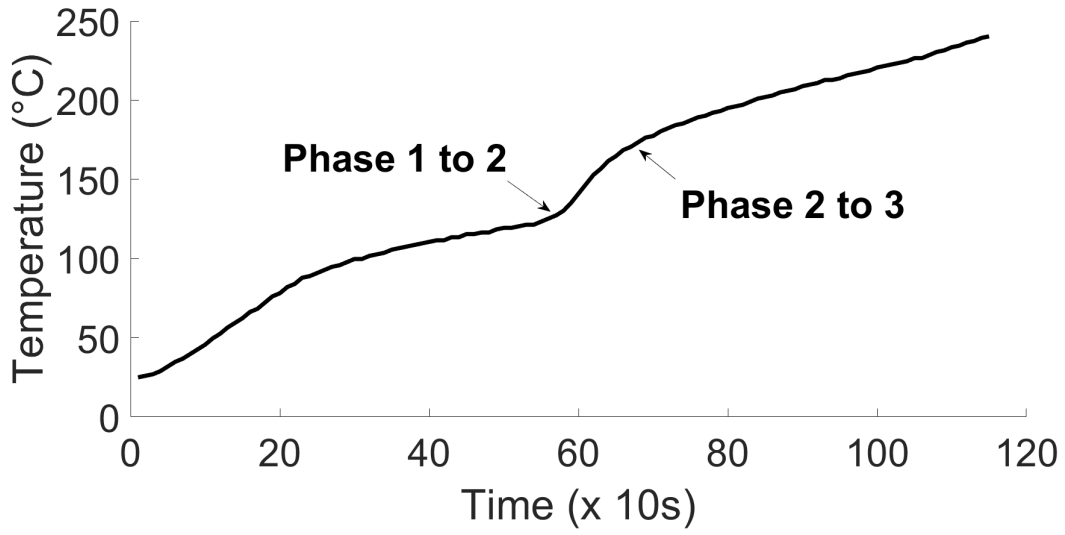

Figure 3.10: An example of where the start and end points of the phases are manually chosen based on the inflection points of the observed output temperature.

A comparison between the original MPSSID and the manual-partitioning version is shown in table 3.7. These results show that the manual-partitioning model does not outperform the original multi-phase formulation. One reason for this could be an inconsistency between the training and testing data. The model is trained based on phase partitioning that is done by visual inspection, but in testing the phase switches still occur at designated output temperatures. If the phase switching criteria was changed to depend on a numerical method inflection point indicator that tracked the second derivative of the data at each observation, the results may improve; however, it

is much simpler, especially when designing an MPC program, to use output-dependent switching criteria and so the manual method is not recommended.

Table 3.7: Original three-phase and Manual-Phase Partitioning three-phase model prediction comparison metrics for differing batch completion scenarios. Heater temperatures were used as the inputs. The $\mathbf{q}_1$ and $\mathbf{q}_2$ quality variables represent the Sinkhole Area and Impact Strength, respectively. Note that the last column reports the number of times a model predicted quality values that are physically impossible.

	Batch Completion	Temperature SSPE $(^{\circ}C)^2$	$\mathbf{q}_1$ SSPE (% area coverage)^2	$\mathbf{q}_2$ SSPE $(kg \cdot m)^2$	$\mathbf{q}_1$ and $\mathbf{q}_2$ predictions < 0
Heater Temperatures Three-Phase	5%	110,354	6.12	10.22	0
	20%	46,731	6.08	10.02	0
	40%	44,352	6.72	10.14	0
	60%	18,859	6.63	10.03	1
	80%	8,094	6.76	9.41	2
	TOTAL	228,390	32.31	49.82	3
Heater Temperatures Three-Phase Manual Entry	5%	144,603	6.79	10.21	1
	20%	110,858	6.61	9.78	0
	40%	108,238	6.64	10.07	0
	60%	65,271	9.57	9.57	0
	80%	7,162	7.33	9.06	4
	TOTAL	436,133	36.94	48.69	5
	% improvement from original	-90.96	-14.34	2.25	−66.67

3.3.4 Cascade MPSSID

As shown in subsection 3.3.1, using the heater temperatures as the process inputs yields better predictability of the rotomoulding process than using the heater powers. However, heater powers were used as the controlled inputs to generate the experimental data that was analyzed. A "cascade" multi-phase SSID algorithm was tested because of its ability to maintain the existing experimental setup while incorporating the heater temperatures. This was constructed as follows:

1. System matrices of a data set were calculated for each phase by performing MPSSID with heater powers as the inputs and heater temperatures as the out-

puts.

2. A second set of system matrices for the same data set were calculated for each phase by performing MPSSID but with heater temperatures as the inputs and the internal mould temperature as the output.

During the testing analysis, the internal mould temperature was predicted as follows:

1. The heater powers were input to the first MPSSID to calculate a heater temperature prediction.

2. The heater temperature prediction was input to the second MPSSID to calculate an internal mould temperature.

The results of this implementation are presented in table 3.8. The results show that the cascade method yields very poor results and so is not worth testing in lab experiments. The temperature and sinkhole area SPE's are in the neighbourhood of an order of magnitude greater than even the SPE reported for the one-phase heater power case. The impact strength SPE is over five times greater than the one-phase heater power case.

Table 3.8: Cascade model prediction comparison metrics for differing batch completion scenarios. Heater temperatures were used as the inputs. The $\mathbf{q}_1$ and $\mathbf{q}_2$ quality variables represent the Sinkhole Area and Impact Strength, respectively. Note that the last column reports the number of times a model predicted quality values that are physically impossible.

Batch Completion	Temperature SSPE $(^\circ C)^2$	$\mathbf{q}_1$ SSPE (% area coverage)^2	$\mathbf{q}_2$ SSPE $(kg \cdot m)^2$	$\mathbf{q}_1$ and $\mathbf{q}_2$ predictions < 0
5%	6,412,343	3,261.45	128.18	0
20%	6,409,336	3,246.50	126.33	0
40%	6,467,357	3,200.95	124.57	0
60%	1,955,220	979.97	25.33	0
80%	6,722	7.16	9.28	2
TOTAL	21,250,978	10,696.02	413.69	2

3.3.5 Reidentification

In [19], the authors formulate a subspace "reidentification" approach where the system matrices are periodically updated during experimental runs based on the most recent measured output value. This approach works by performing on-line SSID on subsets of historical batch input/output data, whereas traditionally, SSID is performed off-line. For MPSSID, the data is partitioned into phases based on predetermined output values and SSID is performed off-line for each phase.

In reidentification, the subset for each historical batch is determined by selecting the data that spans from the location of current measured output to the end of the batch. These subsets are then organized side by side into the appropriate Hankel matrices so that batch SSID, which is introduced in [10], can be used to determine the new system matrices. The advantage of reidentification is that it does not require any PLS steps to connect phases like MPSSID does. It also does not require pre-determined output criteria; however, it still does not provide an accurate prediction of the temperature trajectory (and thus the final product quality) from the early stages of an experimental run. This is demonstrated in figure 3.11, where the predictions of a representative set

of batches from the test set (3 and 9) are plotted for the 5%, 20%, and 40% batch completion scenarios. Usually, the reidentification approach contains additional logic so that the system matrices are updated only if the predicted output deviates far enough from the observed output. This reduces the computation time by preventing the reidentification algorithm from running at each time step.

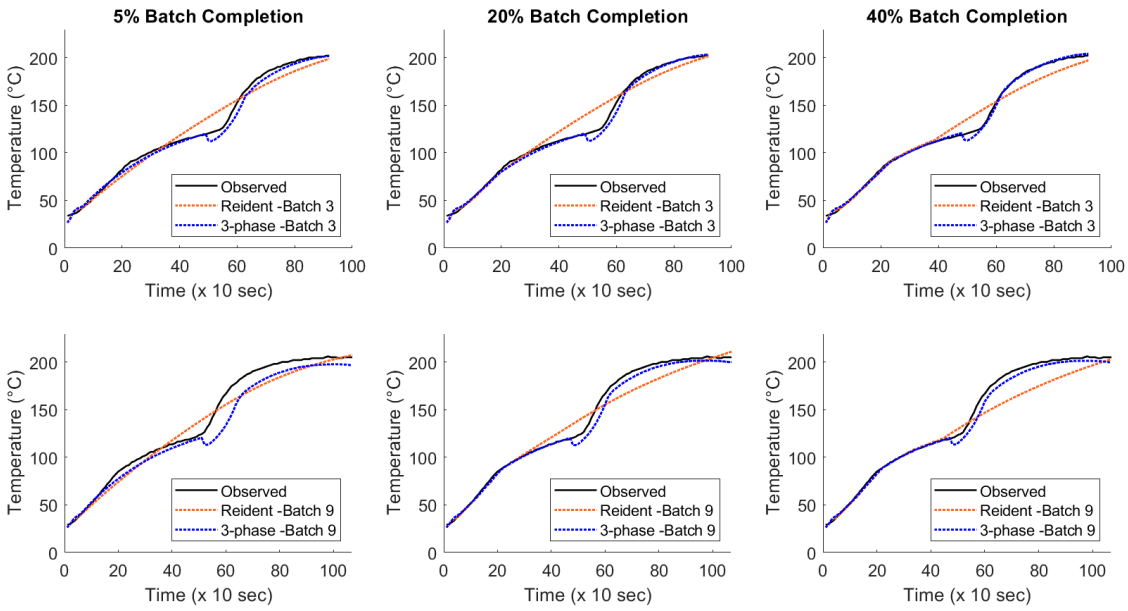

Figure 3.11: A comparison between the predictive ability for the reidentification and 3-phase MPSSID models for the 5%, 20%, and 40% batch completion scenarios. At the begging of the batch the 3-phase model outperforms the reidentification model.

In table 3.9 a comparison between the one-phase, three-phase, and reidentification models is presented for the heater temperature input cases. The results demonstrate the superior performance of the three-phase model over the one-phase and reidentification models, while reidentification only gives minor improvements over the one-phase model. In [19], the average of the sinkhole area and impact strength product qualities for the two lab experiments done with reidentification control were an improvement over the averages of the one-phase method by 59.26% and 27.26%, respectively. Only two experiments were performed for each case, so this is not a statistically significant result, but the improvement should not be ignored. The results in table 3.9 indi-

cate that the multi-phase model could improve the final product quality even more, although there may be diminishing returns.

Table 3.9: One-phase, Three-phase, and Reidentification prediction comparison metrics for differing batch completion scenarios. Heater temperatures were used as the inputs. The $\mathbf{q}_1$ and $\mathbf{q}_2$ quality variables represent the Sinkhole Area and Impact Strength, respectively. Note that the last column reports the number of times a model predicted quality values that are physically impossible.

	Batch Completion	Temperature SSPE $(^\circ C)^2$	$\mathbf{q}_1$ SSPE (% area coverage)^2	$\mathbf{q}_2$ SSPE $(kg \cdot m)^2$	$\mathbf{q}_1$ and $\mathbf{q}_2$ predictions < 0
Heater Temperatures One-Phase	5%	209,718	23.85	10.85	0
	20%	208,978	27.25	10.71	3
	40%	208,227	22.98	11.21	0
	60%	165,676	48.76	8.05	3
	80%	12,875	26.92	10.67	5
	TOTAL	805,474	149.75	51.50	11
Heater Temperatures Reidentification	5%	208,613	29.61	12.56	0
	20%	212,309	36.33	11.92	3
	40%	126,847	12.56	9.41	3
	60%	25,787	48.51	16.46	0
	80%	11,952	14.27	10.54	0
	TOTAL	585,374	137.40	61.14	6
	% improvement from one-phase	27.33	8.25	-18.72	45.45
Heater Temperatures Three-Phase	5%	110,354	6.12	10.22	0
	20%	46,731	6.08	10.02	0
	40%	44,352	6.72	10.14	0
	60%	18,859	6.63	10.03	1
	80%	8,094	6.76	9.41	2
	TOTAL	228,390	32.31	49.82	3
	% improvement from one-phase	71.65	78.42	3.26	72.73
	% improvement from Reidentification	60.98	76.48	18.52	50.00

3.4 Considerations for Multi-Output Processes

The methods discussed in section 3.3 are fairly easy to implement with single-output processes. An extension for future work could investigate methods of implementation for multi-output processes.

One possibility is to set switching criteria for each output. For example, in the multi-phase application to the rotomoulding process, the phase switches were set to occur when the temperature first reached 122 °C and 155 °C. In a different process, additional criteria could be established for another variable, like pressure or conversion. The phase would only switch when all specified criteria are met.

Another option is to construct a Principal Component Analysis (PCA) model for the output variables. At each time, the computation steps would occur as follows:

1. The output values are measured.
2. The measured outputs are fed into a PCA model to determine whether the process is still in the operating region for the current phase.
3. If the PCA step finds the process is no longer in the operating region for the current phase, a phase switch is triggered.
4. The MPC SSID code determines the next set of inputs.

A neural network could also be used for this application instead of PCA.

3.5 MPC for MPSSID

In [20], a one-phase SSID MPC formulation was introduced for control of a lab-scale rotomoulding process. In [21], quality constraints were added to the formulation which improved the results, and in [19], the formulation with quality constraints was used for the reidentification case discussed in subsection 3.3.5. The reidentification method determines a new one-phase model each time the system matrices are recalculated, which requires no change to the original MPC code. The only additional step required

is to recalculate the current state when the reidentification occurs. This step is not part of the MPC code, and is done using:

- The new identified system matrices.
- The entire past trajectory of the previous inputs and outputs of the current experiment.

The MPC formulation for MPSSID is more complex. It must incorporate the PLS steps and trigger them when specified output criteria are reached. In this section, the original MPC model is discussed and then a new MPC model is introduced that is compatible with MPSSID.

3.5.1 One-phase MPC

The one-phase MPC formulation for the latest experimental setup is shown below:

$$\min_{\mathbf{U}_f, l_f} \beta \hat{Q}_{t_f}[l_f - l] \tag{3.9a}$$

$$s.t.\ U_{j,min} \leq u_f[k] \tag{3.9b}$$

$$|u_f[0] - u[l-1]| \leq \delta \tag{3.9c}$$

$$|u_f[k] - u[k-1]| \leq \delta \qquad \forall\ 1 \leq k \leq l_f - 1 \tag{3.9d}$$

$$\hat{\mathbf{x}}[0] = \hat{\mathbf{x}}[l] \tag{3.9e}$$

$$l_f \in \{t_{mpc} + t_{run,1}, t_{mpc} + t_{run,2}, ...\} \tag{3.9f}$$

$$\mathbf{\Lambda} \hat{Q}_{t_f} \leq \mathbf{\Gamma} \tag{3.9g}$$

$$\hat{\mathbf{x}}[k+1] = \mathbf{A}\hat{\mathbf{x}}[k] + \mathbf{B}\mathbf{u}_f[k] \qquad \forall\ 0 \leq k \leq l_f - l \tag{3.9h}$$

$$\hat{\mathbf{y}}[k] = \mathbf{C}\hat{\mathbf{x}}[k] + \mathbf{D}\mathbf{u}_f[k] \qquad \forall\ 0 \leq k \leq l_f - l \tag{3.9i}$$

$$\hat{Q}_{t_f} = L_m \hat{x}[l_f] \tag{3.9j}$$

with,

$$\hat{Q} = \begin{bmatrix} \hat{Q}_1 & \hat{Q}_2 \end{bmatrix}^T \tag{3.10a}$$

$$\beta = \begin{bmatrix} parameter_1 & parameter_2 \end{bmatrix} \tag{3.10b}$$

$$\mathbf{\Delta} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.10c}$$

$$\mathbf{\Gamma} = \begin{bmatrix} constraint_1 \\ constraint_2 \end{bmatrix} \tag{3.10d}$$

Where,

- $\mathbf{U}_f = [\mathbf{u}_f[0], \mathbf{u}_f[1], \mathbf{u}_f[l_f - l]]$ is the decision variable consisting the of control inputs.
- l_f is a decision variable for the heating cycle termination time that the program selects from a pre-defined set of discrete values (chosen from experience).
- l represents the current sampling time
- β represents a parameter matrix used to scale the optimization program for each input.
- $\hat{Q}_{t_f}[l_f - l]$ represents the predicted qualities at the chosen termination time for the determined future set of inputs.
- Equation 3.9b sets the bounds of the manipulated inputs.
- Equations 3.9c and 3.9d set the maximum rate of input change for the manipulated inputs.
- Equation 3.9e sets the initial state to be used in the SSID model to predict the inputs over the next $l_f - l$ time steps.

- Equation 3.9f is the pre-defined set of discrete values for the heater termination time. t_{mpc} is the time at which the MPC model turns on, whereas $t_{run,1}$, $t_{run,2}$, etc. are the predefined run time choices.
- Equation 3.9g sets the final product quality constraints.
- Equations 3.9h and 3.9i are the SSID equations that are used to determine the next input value.
- Equation 3.9j is a PLS model that uses the predicted final state calculated from the SSID model to predict the corresponding predicted final product quality. L_m is the PLS transformation matrix.
- Equations 3.10a and 3.10b set the parameters for each product quality to be used in the objective function.
- Equations 3.10c and 3.10d set the quality constraints.

3.5.2 Multi-phase MPC

For the MPSSID, adjustments to the optimization program are required to include the PLS steps that switch the state from one phase to the next. Drawing inspiration from [22], the following mixed integer linear program was constructed in collaboration

with Aswin Chandrasekar:

$$\min_{\mathbf{U}_f, l_f} \left(\beta \hat{Q}_{t_f}[l_f - l] + \alpha_1 \left(\sum_{p=p_l}^{P} \sum_{k=0}^{l_f - l} k \phi_{pk} \right) \right) \tag{3.11a}$$

$$s.t.\ U_{j,min} \leq u_f[k] \tag{3.11b}$$

$$|u_f[0] - u[l-1]| \leq \delta \tag{3.11c}$$

$$|u_f[k] - u[k-1]| \leq \delta \qquad \forall\ 1 \leq k \leq l_f - 1 \tag{3.11d}$$

$$\hat{\mathbf{x}}[0] = \hat{\mathbf{x}}[l] \tag{3.11e}$$

$$l_f \in \{t_{mpc} + t_{run,1}, t_{mpc} + t_{run,2}, ...\} \tag{3.11f}$$

$$\mathbf{\Lambda} \hat{Q}_{t_f} \leq \mathbf{\Gamma} \tag{3.11g}$$

The following equations are applicable $\forall\ 0 \leq k \leq l_f - l,\ \forall\ p_l \leq p \leq P$

$$\hat{\mathbf{x}}[k+1] - \mathbf{A}_p \hat{\mathbf{x}}[k] - \mathbf{B}_p \mathbf{u}_f[k] - \mathbf{M}\rho_{pk} - \mathbf{M}\phi_{pk} \leq 0 \tag{3.12a}$$

$$\hat{\mathbf{x}}[k+1] - \mathbf{A}_p \hat{\mathbf{x}}[k] - \mathbf{B}_p \mathbf{u}_f[k] + \mathbf{M}\rho_{pk} + \mathbf{M}\phi_{pk} \geq 0 \tag{3.12b}$$

$$\hat{\mathbf{y}}[k] - \mathbf{C}_p \hat{\mathbf{x}}[k] - \mathbf{D}_p \mathbf{u}_f[k] - \mathbf{M}\rho_{pk} - \mathbf{M}\phi_{pk} \leq 0 \tag{3.12c}$$

$$\hat{\mathbf{y}}[k] - \mathbf{C}_p \hat{\mathbf{x}}[k] - \mathbf{D}_p \mathbf{u}_f[k] + \mathbf{M}\rho_{pk} + \mathbf{M}\phi_{pk} \geq 0 \tag{3.12d}$$

$$\hat{\mathbf{x}}[k+1] - \mathbf{A}_p \big(\hat{\mathbf{x}}[k]\, \mathbf{W^*}_p \mathbf{C^T}_p \big) - \mathbf{B}_p \mathbf{u}_f[k] - \mathbf{M}(1 - \phi_{pk}) \leq 0 \tag{3.12e}$$

$$\hat{\mathbf{x}}[k+1] - \mathbf{A}_p \big(\hat{\mathbf{x}}[k]\, \mathbf{W^*}_p \mathbf{C^T}_p \big) - \mathbf{B}_p \mathbf{u}_f[k] + \mathbf{M}(1 - \phi_{pk}) \geq 0 \tag{3.12f}$$

Additionally, we require extra equations to handle the binary variables.

$$\hat{\mathbf{y}} - \mathbf{y}_s(1,p) \leq \mathbf{M}\rho_{pk} \qquad \forall\ 0 \leq k \leq l_f - l,\ \forall\ p_l \leq p \leq P \tag{3.13a}$$

$$\mathbf{y}_s(2,p) - \hat{\mathbf{y}} \leq \mathbf{M}\rho_{pk} \qquad \forall\ 0 \leq k \leq l_f - l,\ \forall\ p_l \leq p \leq P \tag{3.13b}$$

$$\sum_{p=p_l}^{P} \rho_{pk} = P - p_l \qquad \forall\ 0 \leq k \leq l_f - l \tag{3.13c}$$

$$\phi_{ik} - \mathbf{M}(1 - \rho_{pk}) \leq 0 \qquad \forall\ 0 \leq k \leq l_f - l,\ \forall\ p_l \leq p \leq P \tag{3.13d}$$

$$\phi_{ik} + \mathbf{M}(1 - \rho_{pk}) \geq 0 \qquad \forall\ 0 \leq k \leq l_f - l,\ \forall\ p_l \leq p \leq P \tag{3.13e}$$

$$\sum_{k=0}^{l_f - l} \phi_{pk} = 1 \qquad \forall\ p_l \leq p \leq P \tag{3.13f}$$

$$\hat{Q}_{t_f} = L_m \hat{x}[l_f] \tag{3.14a}$$

with,

$$\hat{Q} = \begin{bmatrix} \hat{Q}_1 & \hat{Q}_2 \end{bmatrix}^T \tag{3.15a}$$

$$\beta = \begin{bmatrix} parameter_1 & parameter_2 \end{bmatrix} \tag{3.15b}$$

$$\mathbf{\Delta} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.15c}$$

$$\mathbf{\Gamma} = \begin{bmatrix} constraint_1 \\ constraint_2 \end{bmatrix} \tag{3.15d}$$

$$y_s = \begin{bmatrix} s_1 & s_2 & \cdots & \cdots & \mathbf{M} \\ 0 & s_1 & s_2 & \cdots & s_{P-1} \end{bmatrix} \tag{3.15e}$$

$$\mathbf{W^*}_p \mathbf{C^T}_p = \mathbf{I} \qquad \forall\ p = p_l \tag{3.15f}$$

The variables that were added to the formulation are as follows:

- α_1 is a parameter for objective function tuning.
- $\mathbf{M}$ is a large number.
- ρ_{pk} and ϕ_{pk} are binary variables that have a value of 0 or 1 for each phase, p, at each time point, k. ρ_{pk} is used to turn phases on and off, whereas ϕ_{pk} is used to turn the PLS step on and off. When ρ_{pk} is 0, it means phase p is active at time k. When ϕ_{pk} is 0, it means the PLS step for phase p is *inactive* at time k. Put simply: for ρ_{pk}, 0/1 = on/off; for ϕ_{pk}, 1/0 = on/off.
- $\mathbf{W^*}_p$ and $\mathbf{C^T}_p$ are PLS matrices that enable the transition from one phase to the next. Note that the $\mathbf{W^*}_p$ matrix can sometimes be ill-conditioned. In these cases, an element-by-element calculation of the new states is required which makes used of the $\mathbf{P}_p$ matrix that is derived from PLS. This calculation is described in more detail in [17] and [23], but $\mathbf{W^*}_p$ and $\mathbf{C^T}_p$ are used here for simplicity. $\mathbf{W^*}_p\mathbf{C^T}_p = \mathbf{I}$ for the first phase as no PLS step occurs (see equation 3.15f).
- p_l represents the current phase at time l.
- y_s is a 2xP matrix. The first row is used for equation 3.13a and the second row is used for equation 3.13b. $\mathbf{M}$ is a large number and s_1, s_2, $\cdots$, s_{P-1} are the phase switching indicators (i.e. for the rotomoulding data set, s_1 would be the output temperature that triggers a switch from phase 1 to 2).

Lastly, the function of the equations that were added are described below:

- Placing ϕ_{pk} in the objective function sets its value to 0 (PLS step off) unless constraints force it to be 1.
- Equations 3.12a, 3.12b, 3.12c, and 3.12d (termed here as the "SSID equations") are inequality pairs that exist for each phase. They are turned on and off by the

ρ_{pk} and ϕ_{pk} variables as the process progresses through its phases. When $\rho_{pk} = 0$ and $\phi_{pk} = 0$, the inequality pair turns into an equality constraint for phase p at time k. When either ρ_{pk} or ϕ_{pk} are 1, the inequality pair is unconstrained for phase p, time k. Only the equations that are constrained at time k are used in the MPC calculation of the new input.

- Equations 3.12e and 3.12f implement the PLS step at the phase switch. When ϕ_{pk} is 1, the inequality pair turns into an equality constraint while simultaneously unconstraining equations 3.12a and 3.12b. The PLS step is implemented by multiplying the state, $\hat{\mathbf{x}}[k]$, by the relevant PLS matrices to transform it to the state of the next phase.

- Equation 3.13a sets $\rho_{pk} = 1$ when the predicted output is outside of the range of a phase. As an example, if the switch from phase 1 to 2 in the rotomoulding process occurs at 122 °C, then $\mathbf{y}_s(1,1) = 122$. If the predicted output, $\hat{\mathbf{y}}_s$ surpasses 122, then $\hat{\mathbf{y}}_s - \mathbf{y}_s(1,1) > 0$ and so the value of ρ_{1k} is forced to 1 to satisfy equation 3.13a. This makes sure the phase 1 SSID equations are not used when the predicted output is in the range of phase 2 or 3.

- Equation 3.13b also sets $\rho_{pk} = 1$ when the predicted output is outside the range of a phase. It is necessary to ensure the correct phase switches on when the predicted output reaches a phase switching indicator. Continuing with the rotomoulding example, the switch from phase 1 to 2 occurs at 122 °C and the switch from phase 2 to 3 occurs at 155 °C, so we have $\mathbf{y}_s(2,2) = 122$ and $\mathbf{y}_s(2,3) = 155$. If the predicted output surpasses 122 but is less than 155, $\mathbf{y}_s(2,2) - \hat{\mathbf{y}}_s < 0$ and so ρ_{2k} can be 0 or 1. However, $\mathbf{y}_s(2,3) - \hat{\mathbf{y}}_s > 0$, so ρ_{3k} is forced to 1 to satisfy equation 3.13b. ρ_{2k} is then forced to 0 to satisfy equation 3.13c. This makes sure *only* the phase 2 SSID equations can be used when the predicted output is in the range of phase 2. Without equation 3.13b, the MPC could use the equations for phase 3 instead.

- Equation 3.13c requires that at each time point, k, the SSID equations are only turned on for one phase. For example, if the MPC is being implemented for a three-phase system ($P = 3$) and the current MPC calculation occurs in phase 1, $\Sigma\rho_{pk} = 2$. This means that at any k, the SSID equations will be switched off for 2 of the 3 phases.

- Equations 3.13d and 3.13e set $\phi_{pk} = 0$ when $\rho_{pk} = 1$. This ensures that the PLS equations will not be used for phases outside the range of the current predicted output (recall that for ρ_{pk}, 0/1 = on/off; for ϕ_{pk}, 1/0 = on/off).

- Finally, equation 3.13f guarantees that for each phase, ϕ_{pk} is set to 1 only one time. Practically, this means that the PLS step will only occur once per phase. To ensure the PLS step occurs at the first time step of the phase (where the switch occurs), $\alpha_1(\Sigma\Sigma k\phi_{pk})$ is placed in the objective function to be minimized. Since ϕ_{pk} is multiplied in this term by the time step, k, the optimization program will set $\phi_{pk} = 1$ at the smallest possible k to minimize the value of the term. Further, due to equations 3.13d and 3.13e, ϕ_{pk} can be 1 only if $\rho_{pk} = 0$. Therefore, the smallest k chosen by the objective function for $\phi_{1k} = 1$, $\phi_{2k} = 1$, ... , $\phi_{Pk} = 1$ will occur at the first time point in each phase.

This MPC formulation is much more complex than the original one-phase SSID MPC program. A nonlinear program was avoided, but solving a mixed integer linear program can still take significantly more time than a simple linear program. In the rotomoulding application, the MPC is run every time a measurement is taken, which occurs every 10s. Future work could investigate how well the mixed integer linear program works for the multi-phase setup and if similar results can be obtained if the sampling time frequency is reduced to allow for the longer computation times that may be required.

3.6 Conclusion

In this chapter, limitations to the original MPSSID model were discussed and efforts were made to address these limitations. First, the heater power and heater temperature input cases were compared, with the results showing that it may be more beneficial to utilize heater temperature control than heater power control. Secondly, a propagation-free model was suggested to address the phase switch "jumping." This model successfully removed the jumping, but resulted in worse quality predictions, indicating that alternative methods to stabilize the back propagation step should be explored instead, such as adjustments to the back propagation optimizer. Third, a manual phase partitioning model and cascade model were tested but did not achieve good results. Next, the multi-phase model was shown to outperform the predictive ability of reidentification models that had achieved good experimental results, indicating the MPSSID model is worth testing in the lab. Lastly, an MPC formulation that is compatible with MPSSID was introduced for experimental applications.

3.7 Acknowledgement

The contribution of data for the purpose of analysis from Professor Michael Thompson's lab is gratefully acknowledged, specifically the contributions from Hassan Abdulhussain and Vladimir Gritsichine. Aswin Chandrasekar is gratefully acknowledged for his contributions to the coding side of the equation. Finally, financial support from the McMaster Advanced Control Consortium and the Ontario Graduate Scholarship Award is gratefully acknowledged.

Bibliography

[1] D. Bonvin, B. Srinivasan, and D. Hunkeler, "Control and optimization of batch processes," *IEEE Control Systems Magazine*, vol. 26, no. 6, pp. 34–45, 2006.

[2] D. Bonvin, "Optimal operation of batch reactors—a personal view," *Journal of Process Control*, vol. 8, no. 5, pp. 355 – 368, 1998. ADCHEM '97 IFAC Symposium: Advanced Control of Chemical Processes.

[3] G. Elicabe and G. Meira, "Estimation and control in polymerization reactors. a review," *Polymer Engineering & Science*, vol. 28, no. 3, pp. 121–135, 1988.

[4] S. Aumi and P. Mhaskar, "Integrating data-based modeling and nonlinear control tools for batch process control," *AIChE journal*, vol. 58, no. 7, pp. 2105–2119, 2012.

[5] J. Flores-Cerrillo and J. F. MacGregor, "Control of particle size distributions in emulsion semibatch polymerization using mid-course correction policies," *Industrial & Engineering Chemistry Research*, vol. 41, no. 7, pp. 1805–1814, 2002.

[6] B. Corbett, B. Macdonald, and P. Mhaskar, "Model predictive quality control of polymethyl methacrylate," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 687–692, 2014.

[7] J. Wan, O. Marjanovic, and B. Lennox, "Disturbance rejection for the control of batch end-product quality using latent variable models," *Journal of Process Control*, vol. 22, no. 3, pp. 643–652, 2012.

[8] J. Sjöberg and M. Agarwal, "Trajectory tracking in batch processes using neural controllers," *Engineering Applications of Artificial Intelligence*, vol. 15, no. 1, pp. 41–51, 2002.

[9] M. A. Hosen, M. A. Hussain, and F. S. Mjalli, "Control of polystyrene batch reactors using neural network based model predictive control (nnmpc): An experimental investigation," *Control Engineering Practice*, vol. 19, no. 5, pp. 454–467, 2011.

[10] B. Corbett and P. Mhaskar, "Subspace identification for data-driven modeling and quality control of batch processes," *AIChE Journal*, vol. 62, no. 5, pp. 1581–1601, 2016.

[11] B. Corbett and P. Mhaskar, "Data-driven modeling and quality control of variable duration batch processes with discrete inputs," *Industrial & Engineering Chemistry Research*, vol. 56, no. 24, pp. 6962–6980, 2017.

[12] P. Van Overschee and B. De Moor, *Subspace identification for linear systems: Theory—Implementation—Applications*. Springer Science & Business Media, 2012.

[13] P. Van Overschee and B. De Moor, "N4sid: Subspace algorithms for the identification of combined deterministic-stochastic systems," *Automatica*, vol. 30, no. 1, pp. 75–93, 1994.

[14] R. Shi and J. F. MacGregor, "Modeling of dynamic systems using latent variable and subspace methods," *Journal of Chemometrics*, vol. 14, no. 5-6, pp. 423–439, 2000.

[15] M. L. Darby and M. Nikolaou, "Mpc: Current practice and challenges," *Control Engineering Practice*, vol. 20, no. 4, pp. 328–342, 2012.

[16] H. Zhao, M. Harmse, J. Guiver, and W. M. Canney, "Subspace identification in industrial apc applications - a review of recent progress and industrial experience," *IFAC Proceedings Volumes*, vol. 39, no. 1, pp. 1074–1079, 2006. 14th IFAC Symposium on Identification and System Parameter Estimation.

[17] E. Ubene and P. Mhaskar, "Data-driven modeling for multiphase processes: Application to a rotomolding process," *Industrial & Engineering Chemistry Research*, vol. 62, no. 18, pp. 7058–7071, 2023.

[18] "System identification toolbox: 9.4 (r2022b)," *The MathWorks Inc.*, 2022, https://www.mathworks.com/help/ident/ref/tfest.html.

[19] A. Chandrasekar, H. A. Abdulhussain, V. Gritsichine, M. R. Thompson, and P. Mhaskar, "Adaptive predictive control algorithm for batch processes: Application to a rotational molding process," *Industrial & Engineering Chemistry Research*, vol. 61, no. 48, pp. 17572–17581, 2022.

[20] A. Garg, F. P. Gomes, P. Mhaskar, and M. R. Thompson, "Model predictive control of uni-axial rotational molding process," *Computers & Chemical Engineering*, vol. 121, pp. 306–316, 2019.

[21] A. Garg, H. A. Abdulhussain, P. Mhaskar, and M. R. Thompson, "Handling constraints and raw material variability in rotomolding through data-driven model predictive control," *Processes*, vol. 7, no. 9, 2019.

[22] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.

[23] S. Wold, M. Sjöström, and L. Eriksson, "Pls-regression: a basic tool of chemometrics," *Chemometrics and intelligent laboratory systems*, vol. 58, no. 2, pp. 109–130, 2001.

Chapter 4

Conclusions and Recommendations

This thesis focuses on data-driven modeling and control methods for batch processes using subspace identification. In Chapter 2, a Multiphase Subspace Identification approach was introduced as a way to handle the nonlinearity of multi-phase processes. The multi-phase model is built by grouping subsets of historical batch process data into phases based on predefined output values. A subspace identification model is constructed for each phase, and then inner PLS models are used to connect the final states from one phase to the first state of the next phase. Determining the first and last states of each phase in the training batches to enable construction of the PLS models requires careful propagation techniques. This model was applied to a rotational moulding plastics manufacturing process and the results demonstrated the superior predictive capability of the multi-phase model. Limitations to this chapter included "jumps" that occurred at the phase switching points and poor impact testing quality data. The model was also not a direct representation of the experimental set up.

Chapter 3 further explored the use of the Multi-Phase Subspace Identification (MPSSID) model by conducting a comparative analysis for alternative multi-phase model formulations that could be applied to the rotomoulding setup. First, a comparison between using the heater temperatures and heater powers as the process inputs was tested. The heater power case is the direct representation of the experimental set up, but the heater temperatures case performed best. Next, a propagation-free formulation was introduced and was observed to eliminate the phase switch "jumps" that occurred for the original formulation, but the model resulted in worse quality predictions, suggesting that adjustments to the back-propagation optimizer in the original formulation may be more effective than the propagation-free model. A manual phase switching and a cascade model were attempted without, but no improvements were observed. Finally, MPSSID was shown to outperform the predictive capabilities of a subspace "reidentification" approach. Beyond the comparitive analysis, considerations to use MPSSID for multi-output processes were discussed and a mixed integer linear pro-

gram that can handle MPSSID for model predictive control was introduced. The MPSSID model is now ready to test in a lab setting.

4.1 Future Work

Based on the data presented in this thesis, areas for future work include:

1. A rotomoulding experimental setup that uses the heater temperatures as the inputs to the MPC program.
2. Investigating the results of a controller designed with MPSSID in the lab.
3. Adjusting the back-propagation optimizer to prevent jumps from occurring at phase switch points.
4. Analyzing the computation time of the proposed mixed integer MPC and setting smaller batch frequencies if the time is significant.
5. Test the generalizability of the controller to recycled material or other, similar plastics material.
6. Implement MPSSID on new processes, including multi-output processes.

Future work will focus on more fine-tuning of the MPC model, investigating its MPC applications, and investigating its usefulness for a variety of other materials and processes. As discussed in the introductory chapter, researchers in this field must exercise due diligence and consider if the work they are doing is getting at the root cause of the plastics proliferation problem. Researchers should refuse to work with plastics production companies that do not implement waste management protocols and/or focus exclusively on recycling. Recycling pathways still require virgin material to produce

the final product, which means they require continuous raw material inputs from the oil companies responsible for the environmental crisis we face. Only when enough people are willing to take a stand to fix this problem will governments do what is right.