Defect Minimizing Control of Low Pressure Die Casting

by

XinMei Shi

M.S., Northeastern University, China,2000

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

in

THE FACULTY OF GRADUATE STUDIES
(Electrical and Computer Engineering)

The University Of British Columbia
(Vancouver)

July 2012

Abstract

Controlling and eliminating defects, such as macro-porosity, in die casting processes is an on-going challenge for manufacturers. Current strategies for eliminating macro-porosity focus on the execution of pre-set casting cycles, die structure design or the combination of both. To respond to process variability and mitigate its negative effects, advanced process control methodology has been developed to dynamically drive the process towards optimal dynamic or static operational conditions, hence minimizing macro-porosity in the casting.

In this thesis, a Finite Element heat transfer model has been developed to predict the evolution of temperatures and the volume of encapsulated liquid in a casting with a high propensity to form macro-porosity. The model was validated by comparison to plant trial data. A virtual process has then been developed based on the model to simulate the continuous operation of a real process, for use as a platform to evaluate a controller's performance.

Since macro-porosity cannot be measured during casting, die temperature has been used as an indirect indicator of this defect. A model-based methodology has been developed to analyze the correlation between die temperature and encapsulated liquid volume, a precursor to the formation of macro-porosity. This methodology is employed to assess the suitability of different in-cycle die temperatures for use as indicators of macro-porosity formation. The optimal locations have then been determined to monitor die temperatures for the purpose of minimizing macro-porosity.

A nonlinear state-space model, based on data from the virtual process, has been developed to provide a reliable representation of this virtual process. The control variable-driven portion exhibits linear dynamic behavior with nonlinear static gain.

The resulting MIMO state-space model facilitated the design of a controller for this process.

Finally, the performance of the nonlinear model-based predictive controller was evaluated using the virtual process. Independent of the initial state of the process - i.e. steady state or startup, the controller exhibited the capability to automatically adjust the process toward the dynamic or static optimal operational condition during disturbances examined. The advanced control methodology developed for LPDC provides a novel solution to improve the operational conditions in die casting process.

Preface

A number of journal and conference papers have arisen from the work presented in this dissertation. With the exception of my supervisors, Dr. Guy Dumont and Dr. Daan Maijer, who offered detailed advice on experimental procedures, analysis methods, results interpretation, and editorial suggestions, I am principal author of the text and the figures of the following publications:

1. X. Shi, E. Harinath, D.M. Maijer and G. Dumont, “Nonlinear Model Predictive Control in a Low Pressure Die Casting Process”, prepared for Journal of Process Control, 2012.

2. X. Shi, D.M. Maijer and G. Dumont, “Determination of the Optimal Location to Monitor Temperature in a Low Pressure Die Casting Process”, Materials Science and Technology, UK, Vol 27, No 6, p1073-1083, 2011.

3. X. Shi, D.M. Maijer and G. Dumont, “Nonlinear Identification for Control of Low Pressure Die Casting”, accepted, American Control Conference 2012, Montreal, Canada, June 27-29, 2012.

4. X. Shi, D.M. Maijer and G. Dumont, “Identification for Control of Low Pressure Die Casting”, MSEC2010, Erie, Pennsylvania, USA, p3-13, Oct 12-15, 2010.

5. X. Shi, D.M. Maijer and G. Dumont, “Determination of Optimal Location to Monitor Temperature During Low Pressure Die Casting”, Light Metals 2010 Proceedings, Advances in Materials and Processes, COM2010, Vancouver, BC, p.3-13, Oct 2010.

This paper won the 2011 Best Paper Award from the Light Metals Section of the Metallurgical Society of CIM.

6. X. Shi, D.M. Maijer and G. Dumont, “Assessing Controllability of Low Pressure Die Casting using Process Simulation”, Proceedings of the 7th Pacific Rim International Conference on Modeling of Casting and Solidification Processes 2007, Dalian, China, p635-642, August 2007.

Table of Contents

List of Tables

List of Figures

Acknowledgements

I would like to acknowledge the central role of both my research supervisors, Dr. Guy Dumont and Dr. Daan Maijer, for their continuous guidance and support during the course of my research program. I wish to express my deepest gratitude for their continuous trust, encouragement, patience and guidance throughout this project. They are mentors and friends whose opinions and insights are highly valued and often sought.

I would like to thank all of my colleagues and officemates, especially Jaime Sossa, Masita Mohamad and Francisco Fernandez, for providing a friendly study environment that I was always pleased to work in. My special thanks to Eranda Harinath and Fazel Farahmand for invaluable technical discussions on my academic research.

I wish to thank the members of NRCan's CANMET-MTL for their help in conducting the industrial low pressure die casting experiments that are presented in this thesis. The financial support of NCE AUTO 21 is greatly appreciated.

Special thanks are owed to my family members, my parents, my brother Xinkuan Shi, my husband YiMing Zhang and my beloved son SiChun Zhang who have been my source of motivation, inspiration and unconditional love. I would never have reached my goals, if there had not been the steady support of my family members.

Chapter 1

Introduction

The automotive industry continues to search for and to exploit opportunities to replace steel and cast iron components or assemblies with light-weight aluminum castings. Examples include cylinder heads, engine blocks, suspension components, brake components and wheels. High volume casting methods that operate in batch mode where parts are produced cyclically, such as low pressure die casting (LPDC), have facilitated this conversion [1, 2]. The evolution of the LPDC process and its development as a major manufacturing process have been discussed by a number of authors [3–5]. Currently, the LPDC process plays an increasingly important role in the foundry industry as a low-cost and high-efficiency precision forming technique with new applications beyond its typical use in the production of automotive wheels [2].

In the LPDC process minimizing defects, including macro- and micro-porosity, is an on-going production challenge. Defects cause castings to be rejected because of their deleterious effects in two main areas: aesthetic appearance and mechanical performance [6]. In the case of macro-porosity or shrinkage porosity, these defects form in locations where there is insufficient feeding of liquid metal to offset the volumetric shrinkage associated with the solid-to-liquid transformation. The current philosophy to reduce the formation of macro-porosity is to promote progressive solidification and thereby eliminate hot spots. In practice, this can be accomplished by proper die structure design [7] and the execution of a pre-set casting cycle that does not vary from cycle to cycle, e.g. using programmable logic controllers (PLCs) [8]. Progressive solidification refers to an ideal solidification pattern, which starts and proceeds in continuous fashion, where the solidification front is fed with liquid metal until solidification of the entire casting is complete. A die designed with suitable cooling channels along with a pre-determined sequence for activating and operating these cooling channels can encourage progressive solidification but its is rarely achieved for complex castings.

In principle, these two approaches to control defect formation are straightforward to implement. However, they lack the ability to adjust to the variations in the casting process, resulting in defective products during common process disturbances such as varying incoming metal temperature or varying die open time. The incoming metal temperature of the molten aluminum alloy can vary depending on the holding furnace charging schedule and the temperature of the charged metal. Die open time refers to the length of time that the dies remain open after ejecting a casting. In the industrial process, this length of time is usually constant from cycle to cycle, but may vary when an operator has to perform maintenance on the die or must spend time clearing a blockage or releasing a casting when it clings to the die. These process disturbances cause the process to deviate from its intended steady state operational condition and may result in more macro-porosity in the products. To respond to the process variability, research is needed to develop advanced process control methodologies to help compensate for industrial process dynamics and mitigate their negative effects on the product. By dynamically adjusting the operational parameters of a cyclic casting process in the presence of process disturbances, it is expected that the overall performance of the process would be improved.

1.1 Low Pressure Die Casting

Die casting is a high-volume metal casting process which operates by forcing molten metal into a mold cavity. This process is capable of producing metal parts with smooth surface finish and dimensional consistency [3]. Die casting technologies include high pressure die casting, low pressure die casting, gravity die casting, vacuum die casting, squeeze die casting among others. The low pressure die casting(LPDC) process, a form of die casting, is the dominant method for the production of automotive aluminum wheels. The LPDC process plays an increasingly important role in the foundry industry with new applications being developed, over and above its traditional usage for wheels. The LPDC process accounts for about 20% of light alloy casting production.

1.1.1 LPDC Process Operation

A typical LPDC process, shown schematically in Figure 1.1, is comprised of a die assembly containing one or more die cavities located above an electrically heated furnace, which contains a reservoir of molten metal. A casting cycle begins when the die is closed, creating the casting cavity. Metal enters the cavity through a joint pipe / sprue when the air above the liquid metal in the furnace is pressurized. The casting solidifies as heat is transferred to the die and then to the environment surrounding the die or to cooling media (air or water) circulating through the die with pre-determined cooling cycles. Once solidification is complete, the die is opened and the casting is removed. Following a brief delay to allow the operators to perform intermittent maintenance such as touching up the protective coating on the die surface or cleaning off a piece of flashing, the die closes and the next cycle begins. Following the casting operation, the castings are typically rough-machined, heat treated, finish-machined and painted.

During casting, if progressive solidification cannot be achieved, areas of liquid metal may be encapsulated by solid metal. This cuts off the flow of liquid metal necessary to compensate for the volume contraction that occurs during the solid to liquid phase change. Once encapsulated, these pockets of liquid metal form macroporosity which may result in a defective casting depending on their location and severity.

1.1.2 Heat Transfer and Temperature Evolution in LPDC

In the LPDC process, heat is transferred from the casting to the die and then to the surrounding environment. Dies are fabricated with cooling channels to facilitate localized cooling. In practice, a variety of different fluids, including air, water and/or oil, are used to cool the surface of the cooling channels. The thermophysical properties of the fluid and the die and their relative temperature combine to have

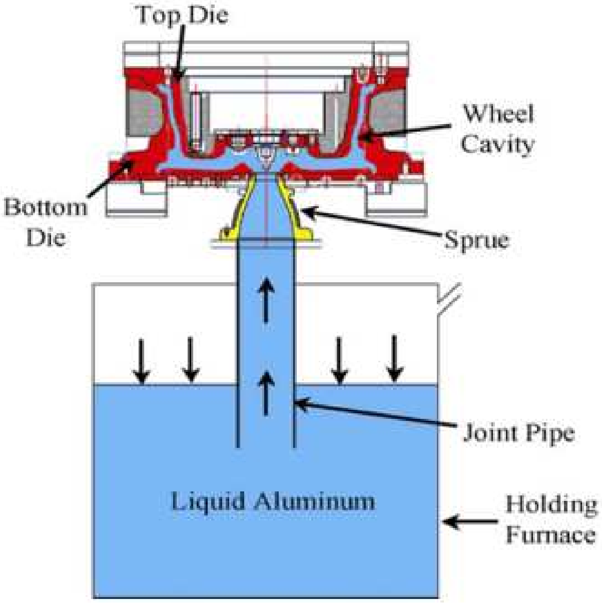
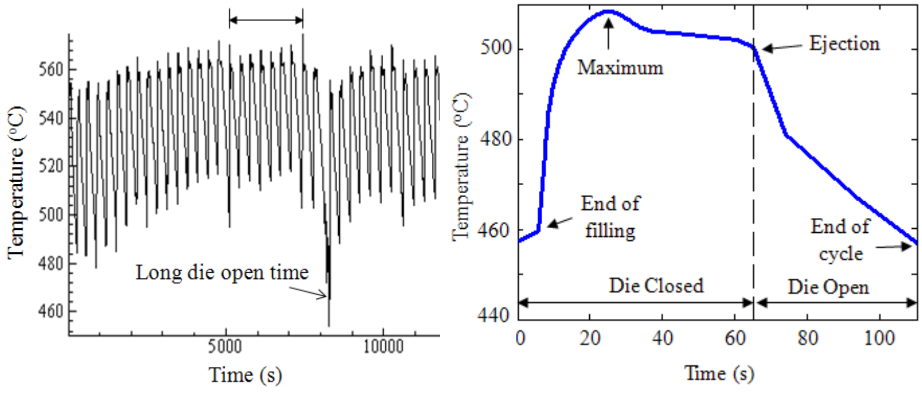


Figure 1.1: A Cross-Section of a Typical LPDC Process used to Produce Wheels [1].

different effects on die temperature. The duration of cooling within a cycle (i.e. length of time the fluid is flowing in the channel) and the number of cycles where the cooling program is active also affect the die temperature. In general, cooling with air has a relatively small cooling effect on the die in close proximity to the cooling channel in each cycle. However, over many cycles, the effects of air cooling can accumulate, causing die temperatures variation over a large region of the die. In contrast to air cooling, for the same cooling duration within a cycle, cooling with water has a strong influence on the die temperature surrounding the cooling channel in each cycle due to its higher cooling intensity [9].

The temperature measured in an operating industrial low pressure die during a plant trial is shown in Figure 1.2a. Each casting cycle results in an increase in temperature as the die heats followed by a decrease as the heat is conducted away from the casting. The combined cyclic temperature history exhibits an oscillatory response. The cycle-to-cycle variation of the temperature shown in Figure 1.2a indicates that the process never reaches cyclic steady state where the starting and final temperatures of a cycle are equal. Initially, this variation is attributed to the start-up transient in the process, but at longer times there are intermittent disturbances that affect the cycle-to-cycle temperatures. At 8000s, a large disturbance may be observed which is due to an extended die open time necessary for maintenance. The somewhat regular variations of cyclic temperatures (every 8 - 10 cycles) that occur throughout the casting campaign are due to metal transfer to the holding furnace. The period for adding metal to the holding furnace is demonstrated in Figure 1.2a and marked with a double-headed arrow. The period starts at 5000s with a low cycle-to-cycle temperature caused by the new metal that is added into the furnace. The metal in the holding furnace gradually heats up, resulting in the subsequent increase in cycle-to-cycle die temperature via heat transfer from casting to die. The period ends at 7500s when new metal is added to the holding furnace.



(a) (b)

Figure 1.2: (a) The Variation of Multi-cycles Die Temperature Measured in an Operational Wheel Die and (b) the Variation of Die Temperature at a Location for a Single Casting Cycle under Steady state.

These results highlight the sensitivity of the die temperature to variations in the process parameters that are linked to the dynamic industrial environment. The operational conditions associated with nominal steady state are determined in industry based on historical values. Ideally, the process operates where minimal defects occur, but this is not necessarily the case.

The die temperature variation at a location for a single casting cycle is presented in Figure 1.2b. The characteristic temperatures, as marked in Figure 1.2b, are Maximum Temperature and Ejection Temperature. The former is the peak temperature reached at a location within a cycle while the latter is the temperature in the die upon opening and ejecting the casting. Both temperatures are highly relevant to solidification in the casting. The time when the Maximum Temperature arises is a complex function of the process. It is a turning point where heat-in rate equals to heat-out rate. When heat-in rate is greater than heat-out rate, die temperature at a location will go up, otherwise it would go down. heat-in rate here is the heat transfer rate conducted from the hot liquid metal entering the die to the location while heat-out rate is the heat transfer rate caused by cooling convection and die conduction. The Ejection Temperature is a specific time in the casting cycle and should correspond to the time when complete solidification of the casting is achieved. When cyclic steady state is reached, the temperature at the start of a cycle has the same value as the temperature at the end of the cycle, just as shown in Figure 1.2b. To achieve steady state, the LPDC operational parameters must remain unchanged cycle-to-cycle and the process must not experience any disturbance in the surrounding environment. In practice, the ideal steady state operational conditions are impossible to achieve due to continuous process disturbances.

1.1.3 Porosity

Porosity is a common defect that forms during the casting operation. Based on the size of the pores, porosity is classified as either micro- or macro-porosity. Microporosity is generally characterized as small dispersed voids ($< 500\mu m$) which are formed when poor interdendritic feeding causes the exsolution of hydrogen gas bubbles, while macro-, or shrinkage, porosity arises during the solid-to-liquid transformation when pockets of liquid metal are encapsulated by solidified material. This shrinkage is approximately 5.4% by volume in A356 [10], a common alloy used for wheel castings.

In comparison to macro-porosity, micro-porosity is more complex, as the size, amount and distribution are functions of process and alloy variables [11]. Macroporosity may be relatively easy to avoid by ensuring progressive solidification, since feed metal is continually available for the solidification process. Although the conditions leading to the formation of macro-porosity are straightforward to identify, the final shape and distribution of macro-porosity are difficult to determine. Thus, there is a focus on identifying the occurrence of liquid encapsulation rather than macro-porosity [1, 12].

In order to assess the impact of techniques to eliminate macro-porosity, it is first necessary to develop an approach to identify and quantify the volume of macro-porosity formed during casting. However, on-line or in-line characterization of macro-porosity during manufacturing is a major challenge for producers and researchers. Although non-destructive measurement techniques, including ultrasonic inspection and X-ray imaging [13–15] may be used to assess casting quality, the use of these methods in an industrial setting to characterize macro-porosity is costly and the results tend to be qualitative rather than quantitative. Additionally, the time delay associated with performing and processing the measurements will adversely affect an operator's ability to respond in a timely manner.

Liquid encapsulation, a precursor to the formation of macro-porosity, occurs when a volume of liquid is surrounded by solid (or near solid) metal. The formation of encapsulated liquid regions is directly related to the casting's temperature history during solidification [6, 16] which in turn is related to die temperatures. Given this linkage, it is possible to use die temperatures as an indirect indicator of macro-porosity. This concept is supported by industrial experience which indicates that proper control of die temperatures is essential for producing superior quality components and yielding high production rates [17, 18]. However, relatively little work of investigating the quantitative relationship between die temperatures and liquid encapsulation has been published.

1.2 Literature Review

As mentioned previously, the traditional approaches for reducing macro-porosity in LPDC are to optimize die design by promoting progressive solidification and to use preset cycle timing via Programmable Logic Controllers (PLCs) where the control parameters remain static during each cycle of the process. The development of die design and process operational parameters is typically based on operational experience and/or trial-and-error. In recent years, control methodologies are being adapted and applied in the manufacturing industry to improve process performance [17–19].

In order to develop a control methodology for a new process, numerous experiments are performed in an effort to characterize process dynamics and to test the performance of the proposed control scheme. It is costly to conduct experiments directly on industrial processes, hence in-plant trials are not appropriate until the control methodology has been thoroughly tested. As a result, there is a need to develop offline process simulations to develop and optimize industrial process controls.

The following subsections will discuss two main topics: Process Modeling and Control Methodologies.

1.2.1 Process Modeling

Most materials-processing operations involve a complex series of steps or unit operations in which the product shape and mechanical properties are developed. In order to meet increasingly stringent product and processing objectives, detailed knowledge of the relationship between process variables and material properties must be clearly delineated. Mathematical models of materials processing operations have become important tools for engineers to design and troubleshoot process operations and to optimize products [12, 19, 20].

In building a mathematical model of a materials processing operation, it is important to bear in mind that all real processes are complex, and hence any attempt to build an exact description of the process is usually impossible. Therefore, different types of process modelling techniques are applied depending on the specific requirements [21]. The internal details of a process are relevant only to the extent that they are necessary to achieve the desired level of performance. Models whose computational time is longer than the time required for a process operation are of limited value for control-system implementation.

There are many different types of modeling approaches that can be applied to understand casting process dynamics and to predict the formation of defects. Computer-based process models that are based on heat, mass, and momentum balances incorporate descriptions of the inner dynamics of the casting process and provide high-fidelity predictions of relevant solution variables such as temperature. The types of processes being modeled, as well as the complexity of the models, has increased over time. Repeated runs of these models can be used to simulate a cyclic operation. These models can be augmented to predict the formation of defects. Process models for use in control solutions tend to ignore the extremely detailed and complex information occurring inside the process (such as heat transfer in the casting and boundary conditions along the die).

1.2.1.1 High-fidelity Process Modeling

High-fidelity process modelling, a computer based technique, involves the development of a geometrically accurate model that incorporates the relevant physics of a process through first principles calculations. Usually it is applied for simulating complex processes. It can assist in decision making for scale-up, reactor design [22] and reducing energy consumption, for example. The use of high-fidelity process modelling can reduce costs by avoiding expensive process trials.

In the case of die casting, high-fidelity process models can be used to predict temperatures, fluid flow, composition and defects formed during solidification within both the casting and die where appropriate. These types of models have been used for die structure design [20] and to identify approaches for improving product quality from a given process [12]. Numerous studies have been presented in the literature detailing the application of high-fidelity models to predict the progress of solidification in die casting. Zhu et al. [23] developed a modified cellular automaton model and demonstrated its capabilities in modeling the microstructure evolution during solidification of aluminum alloys by simulating dendritic and nondendritic microstructure evolution in semisolid processing of an Al-Si alloy. Tin et al. [24] integrated process models for the various stages of gas-turbine disc manufacture to simulate the physical and microstructure transformations occurring within a nickel-based superalloy throughout the entire manufacturing route. It was shown that the microstructure of the alloy changes significantly throughout the process chain, the final microstructure and defect distribution at each stage being related to those formed in the previous stages.

To develop a high-fidelity model of the heat transfer occurring in LPDC, the governing partial differential equation (PDE) (1.1) for transient heat conduction is solved. The time varying temperature distribution of the process can be obtained by seeking ”analytical solutions” or ”numerical solutions” to Equation (1.1) with relevant heat flux boundary conditions (BC’s) (Equation (1.2)), initial conditions (IC’s), material properties and geometry.

$$\frac{\partial}{\partial x}\left(k_x\frac{\partial T}{\partial x}\right)+\frac{\partial}{\partial y}\left(k_y\frac{\partial T}{\partial y}\right)+\frac{\partial}{\partial z}\left(k_z\frac{\partial T}{\partial z}\right)-\rho C_p\frac{\partial T}{\partial t}+Q=0 \tag{1.1}$$

$$q=-k\frac{\partial T}{\partial n_s}=h(T_s-T_\infty) \tag{1.2}$$

Analytical solutions are techniques for solving PDE’s often based on geometric and BC simplifications. Stefanescu [25] applied analytical solutions to describe non-steady state heat transport and solidification of castings. Non-steady state heat transport is typical for some progressive solidification processes such as chill casting, as well as for the vast majority of casting processes, including sand casting, die casting, etc. The PDE that describes typical progressive solidification processes can be solved analytically only if further simplifying assumptions are involved, such as assuming solidification occurs at a single temperature rather than over a temperature range and assuming the mold is semi-infinite. It is also possible to develop analytical solutions by simplifying the relevant PDE. Shadloo et al. [26] presented an analytical solution for magnetohydrodynamic flows of viscoelastic fluids in converging/diverging channels. A similarity transform was used to reduce the Navier-Stokes and energy equations to a set of non-linear oridinary differential equations that were solved analytically by means of the homotopy perturbation method.

Numerical solutions are widely applied to heat transfer problems which cannot be simplified to analytical solutions due to nonlinearities, complex geometries and/or complicated BC’s. Early models based on the finite difference method (FDM) [27, 28] tended to necessarily adopt simplified geometries and boundary conditions. In 1988, Hwang and Stoehr [29] used the simplified marker and cell method based on the FDM to simulate 2-D molding filling phenomena including the effects of turbulence and wall shear stress. Others employed the finite element method (FEM) to tackle complex geometries [30, 31] and solved problems subject to sophisticated temporally and spatially dependent boundary conditions [7, 32]. FEM has been used to simulate heat transfer processes in casting. Kobryn [33] built thermal FEM models to accurately predict the thermal history of Ti-6Al-4V castings. Interface heat-transfer coefficients $h_o(T)$ were established as a function of casting surface temperature using a calibration-curve technique. The FEM-predicted casting and mold temperatures were found to be insensitive to certain changes in the h0 values but sensitive to others. Recently, the FEM has been applied to solve for fluid flow during mold filling and solidification of a casting. Jeong and Yang [34] employed the marker surface method and the adaptive grid refinement technique in a 3-D FEM analysis of the filling stage in a die-casting process. By a display technique in which the shaded images are sequentially combined into a final image, the molten metal flow field was effectively visualized.

Another application of FEM techniques is to model stress development during the solidification and cooling of a casting. Shabani [35] developed a new approach that combines an artificial neural network (ANN) and an FEM modeling technique to accurately predict the mechanical properties of A356 such as its tensile strength and yield stress.

Chae et al. [36] applied both analytical and numerical techniques on the sheet casting processes in which the flow domain of the system can be separated into two parts based on the flow kinematics. Then they developed a coupled approach for the prediction of the sheet profile, which combines one-dimensional analytical methods on the planar elongational flow region and a three-dimensional numerical method on the other region. As a result, the prediction from the developed coupled approach was as good as that from three-dimensional numerical simulation previously developed.

High-fidelity modelling has been widely used for many applications [37–41]. The availability of increasingly fast and cheap hardware [42] along with the development of modeling software packages such as Phoenics, Fidap, Fluent, Abaqus, and Ansys have made it possible to model virtually every materials-processing operation. Increasingly, current modeling efforts are focused on coupling microstructure and property prediction into the process-modelling framework. The most critical and most difficult stage in the development of high-fidelity mathematical models is their verification. A number of techniques are used to achieve verification, including the use of physical models and the application of process sensors or tracers during plant operation.

The execution time of these high-fidelity models is typically many times longer than the real process. Therefore, they are able to be used for offline process simulations but not feasible for use in a real-time control solution. Usually high-fidelity models are not suited to act as control models because of both their complexity and long running time, but they can be used as a virtual process framework to assess the performance of control methods as a high-fidelity model can run repeatedly to simulate a continuous cyclic process like a virtual process. The higher the expected performance of the control system, the higher the required fidelity of the model on which it is based.

1.2.1.2 Models for Control

Models developed for control purposes usually differ from those intended for other purposes such as process design. These types of models seek to simplify a process to as few equations as possible by only capturing the control-relevant features of the plant dynamics and plant nonlinearities. As one part of control algorithms, models for control usually run fast because of their concise expressions, hence making it possible to quickly test model-based control methodologies.

Sometimes a first-principles nonlinear model of the plant is available to characterize the dynamics of the input-output behaviours. The first-principles model contains the equations obtained from knowledge of the underlying physical processes [43, 44]. If an analytical solution can be obtained to the first-principles model, then this model, called a ’white box’ model, can act as a control model.

For most complex industrial processes where first-principles dynamic models are too difficult and expensive to develop, the problem of building mathematical models of dynamic systems is usually dealt with by system identification based on observed data from the system. Models obtained in this way are ’Black box’ models, which represent only the input-output dynamic behaviour of the plant, and carry no information about its internal structure [45–47]. ’Grey box’ models are constructed by combining knowledge of the system and experimental data despite a lack of specific knowledge of what is going on inside the system. These models have a number of unknown free parameters that must be estimated using system identification [48, 49].

Since real processes are nonlinear and complex, most control models are black box models. A variety of mathematical control models based on system identification have been developed to describe the dynamic behaviours of processes. Models can usually be reduced to differential equations (continuous-time)[50, 51], difference equations (discrete-time)[52, 53], or a combination of these (hybrid or sampled-data systems)[54, 55] according to the data sampling manner. Discrete-time models are usually chosen for die casting processes as they are batch processes.

In the case of die casting where nonlinearities, complex geometries and complicated BC’s are involved, the ’black-box’ discrete time models that are typically employed are not based on the physical properties of the process but rather on correlations between the process inputs and outputs. Compared to FDM and FEM, the execution time of these control models is much faster. Therefore, they are better suited to be used as the prediction models in advanced model-based controllers.

Models formulated through system identification can be classified into linear and nonlinear systems in terms of the input - output relationship. Although almost every real system includes nonlinear features, many systems can be reasonably described, at least within certain operating ranges, by linear models. A linearized model can be developed for a nonlinear system at the neighbourhood of an equilibrium point if the principles of superposition and homogeneity are valid in the operating range of interest [56]. The incentive to try to approximate a nonlinear system by a linear model is that the techniques for linear control are more developed and straightforward to apply than they are for nonlinear systems [45, 57, 58]. The form of linear model that is adopted does not constrain the kind of test that should be applied to the plant.

A good example of a linear discrete-time control model application in die casting is that of Maijer et al. [12], who developed a reduced order state-space model to predict the approximate input-output behaviour of a low pressure die casting process simulation. The state-space model, generated by the ’Numerical Algorithms for Subspace State-Space System Identification’ (N4SID) method, is described by Equation (1.3). In the neighbourhood of the equilibrium operating point, the identified state-space model fit validation data very well and provided good prediction of the input-output behaviour of the casting process simulation for use in the MPC controller.

$$\begin{aligned} x(k+1) &= Ax(k)+Bu(k) \\ y(k) &= Cx(k) \end{aligned} \tag{1.3}$$

where k represents the current discrete time step, A, B, and C are matrices that operate on the vectors $x(k)$ and $u(k)$ to produce the vectors $x(k+1)$ and $y(k)$. The $x(k)$ and $x(k+1)$ vectors represent the state variables of the model at the current and next time steps, respectively. The vector $u(k)$ represents all of the model inputs and $y(k)$ represents all of the model outputs.

Most real-world systems exhibit linear behaviour within a limited operating range. However, for cases where the system’s response has to be predicted over a wide operating range that includes nonlinearity, or where the principles of superposition and homogeneity do not hold in the neighbourhood of an equilibrium point, a nonlinear model rather than linear model must be used to accurately predict the dynamics of the input-output behaviour.

The techniques for building nonlinear models have not been as thoroughly developed as those for linear models. However, as computer technology is increasingly growing, more and more researchers are focusing on the development of nonlinear models to pursue the higher performance which they afford. For nonlinear system models determined via system identification, the state-space modeling technique is commonly used mainly because it can deal with both single-input-single-output (SISO) and multi-input-multi-output (MIMO) conditions. There are different nonlinear forms of state-space models. Isabelle et al. [59] performed neural network black-box modeling to produce a nonlinear neural state-space model, and showed that neural network-based state-space models are potentially more efficient than their conventional input-output counterparts. The Wiener Model [60] has been applied to a number of different processes, such as modeling electromechanical systems and radio frequency components [61, 62]. The popularity of the Wiener Model lies in that it has a convenient block representation, transparent relationship to linear systems, and is easier to implement.

Apart from nonlinear state-space models, there are also reported applications of other forms of nonlinear models. Yang et al. [9] developed a fuzzy model to describe the input-output dynamic behaviour of a high-pressure die casting process. The empirical knowledge of how experts operate the process was modeled and stored in a fuzzy rule database. The experimental results obtained from a laboratory die casting process simulator indicate that the developed fuzzy model-based control system is capable of adjusting the desired supply of cooling water into multiple cooling lines so that the local temperature distribution in a die insert is more homogeneous.

1.2.2 Control Methodologies

Process control is a discipline that incorporates statistics and engineering concepts to deal with system architecture, mechanisms and algorithms in order to maintain the output of a specific process within a desired range. Often the ability to control a process leads to benefits such as improving product yield, reducing energy consumption, increasing capacity and improving product quality. Process control has been extensively used in industry and is an enabling technology for mass production processes such as oil refining, paper manufacturing, chemical production, power plants and many other industries [63–65].

Most control methodologies are model-based. They are applicable for regulating systems to known set points or reference trajectories. To accomplish this, a process control system designer must have a comprehensive understanding of how a process operates. This understanding is usually captured in the form of a mathematical model that is generated by system identification.

The block diagram for the rationale of a model-based controller is illustrated in Figure 1.3.

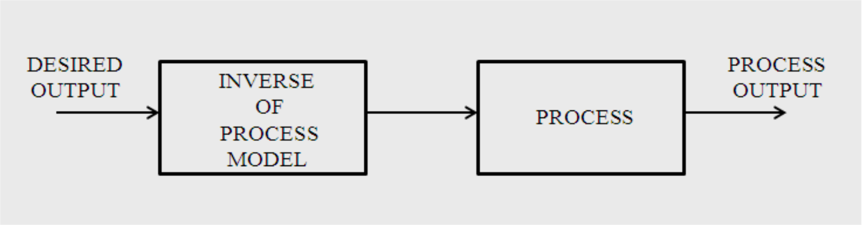


Figure 1.3: Ideal Model-Based Control scheme.

If an accurate model of the process is available, and if its inverse exists, then the process dynamics and static gain can be cancelled by the inverse model. As a result, the output of the process will always be equal to the desired output [21]. Model-based control has the potential to provide perfect control. However, given that there are constraints on process operations, models contain some degree of error, and models are rarely invertible in practice, perfect control is impossible to realize. These are the issues that modern control techniques aim to address, either directly or indirectly.

Technological improvements have led to different types of model-based control methods that have been applied in the manufacturing industry. It has been estimated that in excess of 90% of all controllers currently in operation worldwide are Proportional+Integral+Derivative (PID) controllers [18, 21]. The tuning process for a PID controller requires the development of a series of rules such as the internal model control tuning rules given by Skogestad [66]. The settings of a PID controller are selected such that the controlled response follows a defined trajectory or exhibits stability [67]. However, regular PID controllers can not deal with processes which exhibit complex and mostly nonlinear characteristics. Hence, advanced control methodologies have been developed for cases which require some or all of the following capabilities:

- Control and optimization of MIMO systems
- Automatic adaptive tuning of control parameters while monitoring a process
- Compensation for measured disturbances
- Input and output constraint handling

Model Predictive Control (MPC) is an advanced control methodology that provides many advantages over a PID controller. An example of MPC application is described in the paper [19] to show the superior performance of the MPC approach over PID control. Shang [19] applied MPC to control the temperature of steel strips (hot band) immediately after hot rolling and just prior to coiling. In this process, residence time variation is analogous to variation in transportation lag, which is a destabilizing influence on conventional PID controllers. The MPC approach had no difficulty in these situations as the MPC approach incorporates a prediction phase that easily compensates the residence-time variations. The MPC approach turned out to maintain the target values perfectly while the PID controller had difficulty doing so. Although this is a simple example to illustrate some of the performance advantages of the MPC approach within the context of a typical metal-processing example, there are several other points worth mentioning. The MPC technology can naturally incorporate constraints on the manipulated and/or other process variables. Further the MPC approach, with some simple modifications, can be enhanced to control cooling/heating profiles, whereas, the more conventional control approaches cannot.

MPC has had a significant and widespread impact on a variety of industrial processes including oil & gas, pulp & paper, chemical and batch processes [12, 68–71]. The widespread industrial acceptance of MPC technologies has in turn spawned considerable academic research and industrial development efforts. MPC provides a simple to understand framework that can be easily extended to handle multivariable processes and constraints on the inputs and outputs of a process. it uses the process model to predict the future response of the process at each sampling interval. Using this prediction capability, the MPC controller minimizes a cost function at each sampling interval to produce the optimum process inputs. By using the process model for prediction, an MPC controller can easily integrate compensation for measured process disturbances and constraint handling.

A basic discrete MPC scheme is presented in Figure 1.4. In the simplest case the input trajectory ($\hat{u}(t|k)$) is computed so as to bring the predicted value of the controlled output variable ($\hat{y}(t|k)$) at the end of the prediction horizon 'Hp' to the required value $r(k+H_p|k)$, while only executing 'H_u' control moves. Figure 1.4 illustrates the input assumed to vary over the first three steps of the prediction horizon, but to remain constant thereafter: $\hat{u}(k+2|k) = \hat{u}(k+3|k) = \ldots, \hat{u}(k+Hp-1|k)$, so that there are three 'parameters' to choose: $\hat{u}(k|k), \hat{u}(k+1|k), \hat{u}(k+2|k)$. In Figure 1.4, the algorithm can be shown to split into two phases. In the first phase, a prediction is made Hp control intervals into the future based on past control actions and assuming no more will be made in the future. Based on the free response prediction ($\hat{y}_f(t|k)$), namely the response that would be obtained if the future input trajectory remained at the latest value $u(k-1)$, and the reference trajectory ($r(t|k)$), an uncontrolled error trajectory can be calculated. This error trajectory is an expression of how the output variable (y) is expected to evolve in time should no more control action be taken. The second phase of the MPC approach involves the minimization of this realized error trajectory, usually based on a least-squares calculation and assuming that Hu control actions are taken. Note that $H_u \leq H_p$. When the model on which the MPC calculations are based is linear and a conventional least-squares objective is assumed, the unconstrained optimal solution can be calculated analytically. When the model is nonlinear and/or constraints have been incorporated into the MPC algorithm, then the resulting optimization problem must be solved numerically. The notation $\hat{u}(k+i|k)$ here indicates that at time k a prediction of what the input at time k+i may be; the actual input at that time, $u(k+i)$, will probably be different from $\hat{u}(k+i|k)$. Once a future input trajectory has been chosen, only the first element of that trajectory is applied as the input signal to the plant. That is, $u(k) = \hat{u}(k|k)$, where $u(k)$ denotes the actual signal applied. Then the whole cycle of output measurement prediction, and input trajectory determination is repeated one sampling interval later. For a more thorough introduction to MPC, the reader is referred to the treatments in Maciejowski [72].

Most control methods are model-based, but there also exist non model-based control methods, one of which is extremum seeking control [73], which is a type of adaptive control that does not fit into the classical paradigm of model related schemes. The classical adaptive control approaches for linear [74–76] and nonlinear systems [77] are applicable only for regulating systems to known set points or reference trajectories. In some applications, the reference-to-output map exhibits an extremum (i.e., a maximum or a minimum) which enables the development of a controller to maintain the process output at the extremum value. The uncertainty in the reference-to-output map makes it necessary to use some sort of adaptation to

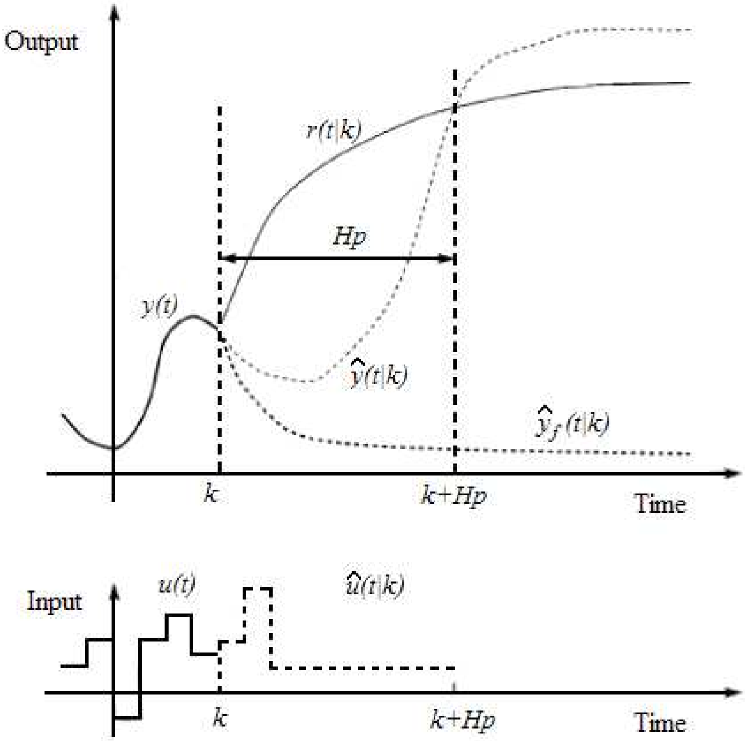


Figure 1.4: A Basic Discrete MPC Scheme.

find the set point which maximizes or minimizes the output. This control is called extremum control or self-optimizing control. The method of sinusoidal perturbation is the most popular of method of seeking the extremum. In fact, it is the only method that permits fast adaptation, going beyond numerically based methods that require a stable process before optimization.

1.2.3 Batch Process Control

Batch processes are as non-continuous processes where ”batches” of product are produced repeatedly from a process. These types of processes are widely applied in many sectors of the chemical, pharmaceutical, food and beverage, polymer, consumer product, and biotechnology industries. Following a thorough literature survey, the application of control methods to batch processes has received considerable attention [63, 78–82]. However, very few studies relevant die casting were found. The following discussion will present the application of control methodologies to batch processes in general. The applications of control methodologies to casting processes will then be discussed in the next subsection.

Xaumier et al. [81] describe the temperature control of a batch reactor for a chemical process where improved product quality was the goal. Huzmezan et al. [80] achieved a significant reduction in cycle time of a batch reactor by reducing the variability of the controlled temperatures. In each case, the in-cycle reference trajectories for temperatures were able to be preset due to the fixed length of each cycle. Temperatures were treated as continuous process variables within a non-continuous batch process. These temperatures were then controlled to follow the in-cycle reference trajectory using model predictive control in both cases.

The dynamic features of some batch processes can be described by a linearized control model when the system runs near optimal operational conditions. Lee et al. [82] assessed the application of Model-based Predictive Control for Batch processes (BMPC) to linear constrained systems, as well as its convergence properties. MPC techniques can also be applied to processes represented by nonlinear control models. Xaumier et al. [81] described the application of nonlinear model predictive control (NMPC) to the temperature control of a semi-batch chemical reactor equipped with a multi-fluid heating/cooling system. The strategy of the nonlinear control system was based on a constrained optimization problem, which was solved repeatedly on-line by a step-wise integration of a nonlinear dynamic model and an optimization strategy.

1.2.4 Die Casting Process Control

In the past, various efforts have been made to develop thermal management systems for dies and a number of temperature control methods have been designed to help control the die temperature. A common control approach in the die casting industry is to define a reference trajectory for die temperature to help achieve the goal of improving the casting quality or reducing the scrap rate [9, 16, 16–18, 83].

A variety of control methods have been developed for HPDC. Bishenden et al. [16] reduced the temperature variation and the scrap rate in an HPDC process by applying a temperature feedback controller to manipulate water flow-rates. In this work, casting defects were traced to the variability of die temperatures in critical die sections. This finding suggests that stabilizing die temperatures should reduce the number of defective casting cycles. Tiebao et al. [9] designed a fuzzy PID controller for a high-pressure die casting process to minimize the temperature differences between two adjacent channels in the die. The control system developed was capable of adjusting the desired supply of cooling water into multiple lines so that the local temperature distribution of the die insert was homogeneous.

Compared to HPDC, there is much less literature on the application of control methodologies in LPDC. Maijer et al. [12] developed and applied MPC control to a simulation of wheel casting. The implemented MPC controller used a linear MIMO state-space model to regulate die temperatures for the purpose of mitigating the negative effects caused from simulated disturbance scenarios, hence improving the casting quality. This study reported that temperature deviations from the optimum values affected liquid encapsulation in the wheel.

In both low pressure and high pressure die casting, the die temperatures have been shown to affect casting quality if they deviate from preset reference trajectories. Note that the reference trajectories for die temperatures are predetermined by either trial and error or empirical knowledge. Therefore, these trajectories can only be qualitatively associated with the ultimate control goal such as casting quality or scrap rate. In addition, controlling temperature, while acceptable, does not provide a direct control of the microstructure and properties of the product. Therefore the best process control is direct product control [19].

1.3 Scope, Objectives and Contributions

1.3.1 Scope of Research Programme

The goal of this research program is to develop a methodology to control an operational LPDC process that is capable of minimizing the negative effects of process disturbances. To achieve this goal, a process control solution that incorporates the dynamics of the process is necessary to drive the system to its optimal operational condition under both dynamic and static conditions. To avoid the cost and difficulties associated with the implementation of control methodologies in an operational industrial process and to enable extensive assessment and testing, a computer simulation-based approach will be employed in this research.

A critical component in developing the control solution for this research was to explore the qualitative and quantitative relationships between die temperature and the volume of liquid encapsulation which is linked to macro-porosity formation. Die temperature will be considered an indirect indicator of the volume of liquid encapsulation that occurs during solidification in an example casting.

A first-principles mathematical model will be developed to predict the evolution of temperature in a low pressure die casting process, using the commercial finite element software ABAQUS. The casting studied will be designed to be susceptible to macro-porosity. The ABAQUS simulation will form the core of a virtual process on which experiments will be performed to explore the relationship between die temperature and the volume of liquid encapsulation and to assess the benefits of implementing an advanced control solution on the casting process. A nonlinear state-space model, based on data from the virtual process, will be developed to predict the input-output behaviour of the virtual process. Using the identified model, a nonlinear model-based predictive controller (NMPC) will then be developed to reject disturbances in the process.

Chapter 2 introduces the design and the validation of the process model within ABAQUS and the transformation of this model into a ’virtual’ process. Chapter 3 presents both correlation method and linear regression method for identifying macro-porosity formed during solidification in LPDC. Chapter 4 describes the preparation and procedure to perform a nonlinear system identification of the casting process. In Chapter 5, a NMPC is designed and implemented on the computing software platform MATLAB. Finally, the performance of the casting process in both the controlled and uncontrolled modes are compared using the virtual process in Chapter 6.

1.3.2 Objectives of Research Programme

The primary objective of the present study is:

- To develop an advanced control methodology to compensate for the negative effects on LPDC caused by process disturbances and to drive the process towards its optimal operating point under both dynamic and static conditions, thereby minimizing macro-porosity in the casting.

To accomplish the primary objectives, the following sub-objectives have been identified:

- To developed and validate a mathematical model of an LPDC process to act as a ’virtual process’ for use in testing an advanced process control solution;
- To formulate a technique that correlates the extent of macro-porosity with die temperatures and use this technique to consider the optimal locations to monitor temperature during LPDC;
- To employ system identification to develop a nonlinear state-space model of the virtual process;
- To assess the performance of advanced control methods on an LPDC process using a virtual process framework.

1.3.3 Contributions

The main contributions of this work are summarized as follows:

1. A mathematical model of an LPDC process, which was designed to be prone to macro-porosity defect formation, has been developed. The model, validated by comparison with experimental data, has been formulated to act as a ’virtual process’ for use in developing and testing an advanced process control solution.
2. A model-based methodology to analyze the correlation between die temperature and volume of encapsulated liquid has been developed. A linear regression (LR) expression to calculate the volume of encapsulated liquid in a casting using die temperatures from selected locations in the die has been formulated and assessed. The method for correlating die temperatures to liquid encapsulation is an original contribution and a paper describing this approach won the 2011 Best Paper Award from the Light Metals Section of the Metallurgical Society of CIM.

3. The correlation method and the LR approach have been used to evaluate and determine the optimal locations to monitor temperatures in the experimental die. This technique provides industry with a quantitative means of selecting and assessing locations to monitor dies and represents an important development over the traditional trial-and-error approach that is typically used.

4. A state-space model with nonlinear dynamic behaviour and linear static gain has been developed to approximate the virtual process through nonlinear system identification (SI). This easy to understand nonlinear SI approach is novel and simple to apply.

5. A nonlinear Model-based Predictive Controller (MPC) was designed to achieve the goal of minimizing the volume of encapsulated liquid. In previous work on die casting, the goal was to control die temperature trajectories, which are determined by experience. The nonlinear MPC controller implemented in the current work, automatically drives the system towards optimal operational conditions both dynamically and statically. This control methodology has been formulated in a general manner to allow application to other cyclic casting processes.

Chapter 2

Virtual Process Model Development[1]

2.1 Overview

Numerous experiments are necessary to develop, implement and test the performance of a control scheme for low pressure die casting. However it would be very costly and impractical to conduct all the experiments directly on a real die casting machine. Therefore, a computer-based die casting process simulator has been built to predict the temperature variation throughout the casting process. The use of a casting process simulator versus performing plant trials provides several advantages including:

- A simulator is inherently flexible allowing a wide variety of operational conditions to be evaluated and it can run concurrently to investigate several phe-

[1] Portions of this chapter have been published in:
• X. Shi, D.M. Maijer and G. Dumont, “Determination of Optimal Location to Monitor Temperature During Low Pressure Die Casting”, Light Metals 2010 Proceedings, Advances in Materials and Processes, COM2010, Vancouver, BC, p.3-13, Oct 2010.
• X. Shi, D.M. Maijer and G. Dumont, “Determination of the Optimal Location to Monitor Temperature in a Low Pressure Die Casting Process”, Materials Science and Technology, UK, Vol 27, No 6, p1073-1083, 2011.
• E. Khajeh, X. Shi, D.M. Maijer, “Modeling the Formation of Porosity during Low Pressure Die Casting (LPDC) of Aluminum Alloy A356”, Shape Casting: 3rd International Symposium 2009, TMS, pp. 297-304, February 2009.

nomena at once.

- A simulator can provide temperature information at any (and/or all) locations within the model domain whereas measured temperature data is limited to the locations where thermocouples were installed in a plant trial.
- It is straightforward to analyze and explore the relationship between die temperatures and the amount of liquid encapsulation in a post-processing operation using a casting process simulator.

To provide a useful tool for investigating an operational process, the simulator needs to accurately represent the real process and it must simulate multiple casting cycles. Functioning in this manner, the simulator can be considered a virtual process. To ensure accuracy, the virtual process must be validated by comparing predicted temperatures to the experimental data for a variety of operational conditions.

This chapter describes the development of a mathematical model of a demonstration casting and the transformation of this model into a virtual process that operates like a real industrial process. The chapter is organized as follows: in Section 2.2, the computer-based die casting process model is introduced. The model validation is presented in Section 2.3. The structure of the virtual process is detailed in the last Section 2.4.

2.2 Model Description

In this research, a computer-based, mathematical model of a casting process was developed and implemented in the commercial finite-element package, ABAQUS. A geometrically simple demonstration die, shown in Figure 2.1, was designed for this study to provide a test platform to assess cyclic casting operations. Contrary to production castings, this casting process was designed with the goal of producing quantifiable defects - specifically two regions of macro-porosity. A transient heat conduction model considering the solidifying LPDC casting and die was developed to predict the temperature variation throughout each casting cycle. The governing equation describing the transient heat transport in the 3-D model is presented in

Equation (2.1):

$$\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right)+\frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right)+\frac{\partial}{\partial z}\left(k\frac{\partial T}{\partial z}\right)-\rho C_p\frac{dT}{dt}+Q=0 \qquad (2.1)$$

where T is the temperature (°C), k is the thermal conductivity (W/m/K), ρ is the density (kg/m^3), C_p is the specific heat (J/kg/K), t is the time (s) and Q is a volumetric heat source term (W/m^3) associated with the latent-heat of solidification in the casting when appropriate, i.e. during the liquid to solid phase change, and x, y and z are directions (m). Equation (2.1) is solved for the temperatures in both the casting and die subject to the definition of model geometry, material properties and appropriate boundary and initial conditions. The model was implemented using the commercial finite element package ABAQUS. Fluid flow within the casting during mold filling has been ignored here; however, when the temperature is above the eutectic solidification temperature, the thermal conductivity, k, of the casting material was intentionally increased to account for enhanced transport of heat due to natural convection. The approach to used to augment k is described below in subsection 2.2.2.

2.2.1 Geometry

By taking advantage of symmetry, the geometry of the casting and die were reduced to a $\frac{1}{4}$ section to shorten computation times. The die geometry employed in the model with overall dimensions of 240mm × 80mm × 80mm is presented in Figure 2.1. Four pairs of cooling channels, also shown in Figure 2.1, were located at different heights in the die. In each pair, coolant (air or water) enters the die along the channel closest to the casting and returns through the other channel. The geometry of the casting and die was meshed using 4-node linear tetrahedral elements with minimum element edge lengths of 4 mm and 7 mm, respectively. The mesh contains 11,773 nodes and 50,115 elements. A more refined mesh was developed to assess the sensitivity of the model predictions to increased mesh density. This entailed comparing the results obtained with the two meshes to assess any difference. The results indicate that the current mesh is adequate for its present use.

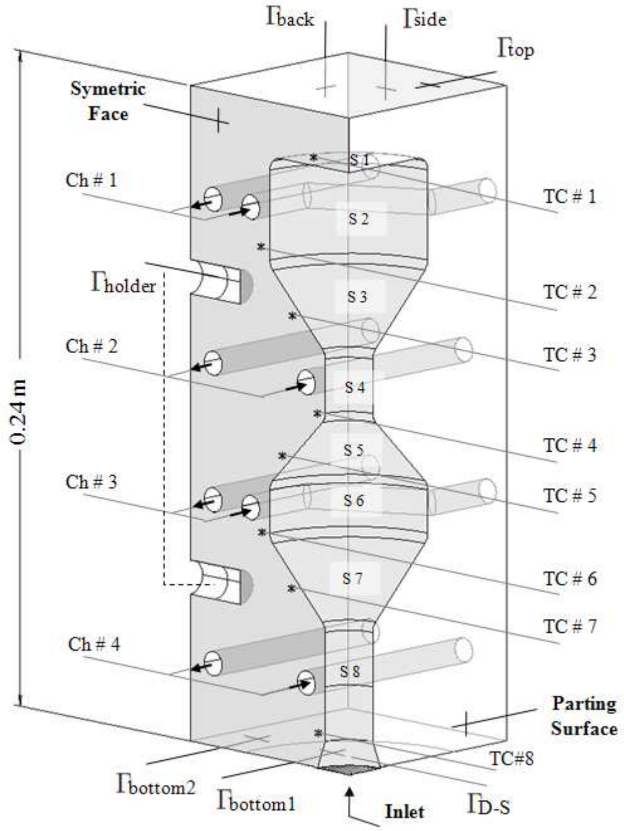


Figure 2.1: Schematic of a $\frac{1}{4}$ Section of the Die with Cooling Channels, Thermocouple Locations, and Interior Surface Partitions Marked.

2.2.2 Thermophysical Properties

The die was fabricated from H13 tool steel and A356 aluminum alloy was used to produce the castings. The nominal compositions of these alloys are given in Table 2.1. The thermophysical properties of A356 and H13, including thermal conductivity, specific heat, density and latent heat (where necessary), used in the model were based on a variety of literature sources and are given in Table 2.2. During the liquid to solid phase transformation in A356, the latent heat of solidification (397.5

Table 2.1: The Nominal Compositions of A356 and H13 [85].

Composition	Si	Cu	Mg	Mn	Fe	Zn	Al
A356	7.0	0.20(max)	0.35	0.10(max)	0.20(max)	0.10(max)	balance

Composition	C	Mn	Si	Cr	Mo	V	Fe
H13	0.32-0.45	0.20-0.50	0.80-1.20	4.75-5.50	1.10-1.75	0.80-1.20	balance

Table 2.2: Thermophysical Properties for A356 and H13 Used in the Thermal Model [86].

Material	T	k	T	C_p	T *range*	L	ρ
	(°C)	(W/m/K)	(°C)	(J/kg/K)	(°C)	(kJ/kg)	(kg/m^3)
A356	133.5	146.54		963	$557 \leq T \leq 568$	198.75	2369
	147.0	153.29			$568 < T \leq 602$	50.4	
	159.1	154.98			$602 < T \leq 610$	148.35	
	203.9	166.80					
	316.4	167.47					
	393.0	166.12					
	557.0	166.12					
	610.0	400.0*					
H13	20	24.60	23	458.80		N/A	7367
	200	26.25	200	518.50			
	500	27.30	400	587.76			
	600	27.76	600	726.20			
	800	28.07	700	905.40			
	850	28.39	760	1151.10			
	900	30.40	800	885.00			
	1000	31.23	850	792.70			
			900	747.90			
			1000	733.00			

* The thermal conductivity of the liquid metal has been increased to approximate the effects of convection[1].

kJ/kg) is released linearly with temperature in three steps. The temperature ranges and the amount of latent heat released in each step are based on the experimental results reported by Thompson et al. [84].

2.2.3 Initial Conditions

The effects of filling on the initial temperature of the casting and die have been neglected in the model based on previous process measurements which showed little temperature change during filling. Thus, the casting is initialized at a uniform temperature of 690 $^\circ$C at the start of each cycle. In the die, a uniform initial temperature of 400-440 $^\circ$C (specified based on the measured initial die temperature) is assumed for the first cycle and in subsequent cycles, the temperature distribution at the end of the previous cycle is used as the initial temperature.

2.2.4 Boundary Conditions

A variety of boundary conditions are needed to properly describe the flow of heat from the casting to the die and then to the surrounding environment. Moving outward from the casting, the first boundary condition encountered is at the interface between the casting and the die. The heat flux at this interface is defined as [1]:

$$-k_{cast}\frac{\partial T}{\partial n}|_{cast} = h_i(T_{cast} - T_{die}) = k_{die}\frac{\partial T}{\partial n}|_{die} \tag{2.2}$$

where n is the outward pointing normal vector of the surface and h_i is the interfacial heat transfer coefficient (W/m^2/K) describing the resistance to heat flow across the interface. Based on surface orientation (i.e. horizontal, vertical, or angled), the casting/die interface was partitioned into 8 sections marked as S #$1-8$ in Figure 2.1. For each section, the maximum and minimum values of the interfacial heat transfer coefficients were defined along with a temperature range over which a linear transition occurs. A trial and error procedure was employed to determine the initial contact heat transfer coefficient (h_{max} = 4000 W/m^2/K) and the subsequent reduction in interfacial thermal resistance due to solidification shrinkage (h_{min} = 400 to 1200 W/m^2/K) for each section. The minimum interfacial heat transfer coefficient was increased on those sections that were expected to maintain improved contact because of the effects of gravity and fluid flow. Table 2.3 summarizes the heat transfer coefficients and temperature ranges over which the heat transfer coefficient was ramped from h_{max} to h_{min} for the eight interface sections.

The heat transfer conditions summarized in Table 2.3 were defined through an

Table 2.3: Heat Transfer Coefficients and Linear Ramp Temperature Range Applied Along the Casting/Die Interface Sections.

Interface Section	*T range* (°C)	h_{min} (W/m^2/K)	h_{max} (W/m^2/K)
1 (trial3*)	540-560	400	4000
1 (trial4*)	565-585	400	4000
1 (trial6*)	540-560	400	4000
2	564-584	1200	4000
3	568-588	1200	4000
4	540-560	1200	4000
5	567-587	400	4000
6	540-560	1200	4000
7	540-560	1200	4000
8	540-560	1200	4000

*trial 3,4,6 represent trials with different operational conditions respectively, detailed in section 2.3

iterative process based on comparison between predicted and measured temperatures and considering the effects of gravity, casting shrinkage, and casting surface orientation. The finish and topology of the casting surfaces were also considered. For example, the lowest h_{min} was assigned to interface sections 1 and 5 based on the expectation that casting contraction and gravity effects will combine to produce significant air gaps along these interfaces. The heat transfer coefficients were ramped at a higher temperature range for interface sections 2 and 3 compared to 6 and 7 because of vertical location with heat transfer at higher locations decreasing earlier (i.e. with higher temperatures). Interface sections where the casting was expected to rest on-, or to bulge out toward-, and maintain contact (sections 2, 3, 4, 6, 7, and 8) were defined with a higher minimum heat transfer coefficient (1200 W/m^2/K).

To account for the additional heat supplied by the liquid metal contained in the riser tube, a temperature constraint was set on the surface representing the inlet from the sprue. In the process, the casting remains in contact with this liquid metal after the die is filled because the filling pressure is maintained. The casting machine used to generate data for comparison with the model was a re-purposed unit with a robotic system that removed the die from the holding furnace each cycle. When this occurs, heat is no longer supplied to the metal in the sprue and cooling occurs to the ambient environment. To represent these conditions, a time-based temperature constraint was applied to the bottom surface of the metal in the sprue, starting at the initial casting temperature (i.e. 690°C), and linearly decreasing to 600 °C between 3s (after the filling process is complete) and 37s (when the furnace pressure is released) during the casting cycle. The temperature decrease and the time over which this occurs were based on the trends observed in the temperature measurements.

Although the flow of liquid metal to the die during filling are not included in the model, the heat transfer along the casting / die interface was initiated as a function of height in the die. The height of the liquid metal in the die was assumed to vary linearly during filling based on the linear pressure ramp applied during the plant trials. Thus the heat transfer across the casting/die interface was activated based on whether or not the liquid metal height in the die cavity exceeded the interface height.

Continuing outward from the casting, boundary conditions were defined on the surfaces defining the internal cooling channels in the die to include the effects of forced convective (air or water) cooling. Heat transfer coefficients and cooling media temperatures were defined based on temperature measurements and trial and error fitting. For comparison with the plant trial data, the starting time for cooling and the duration that cooling is active in the cooling channels are defined based on the experimental conditions. On the exterior surface of the die, the boundary conditions used to describe the heat transfer to the ambient environment surrounding the die, considering both convective and radiative effects, has the form [1]:

$$-k\frac{\partial T}{\partial n}|_{\Gamma} = [h_{conv} + h_{rad}](T_{surf} - T_{amb}) \tag{2.3}$$

where Γ refers to the surface where the BC is applied, h_{conv} is the convective heat transfer coefficient, h_{rad} is an equivalent radiative heat transfer coefficient, T_{surf} is the surface temperature and T_{amb} is the ambient environment temperature. The

Table 2.4: Air Cooling and Exterior Boundary Condition Parameters.

Surface	h (W/m^2/K)	Ambient T(°C)	Surface emissivity	Time in a cycle(t)
Γ_{side}	20	50	0.7	$t > 0$
Γ_{back}	20	75	0	$t > 0$
Γ_{top}	40	20	0.7	$t > 0$
$\Gamma_{bottom1}$	20	75	0.7	$t < t_{fill}$ or $t > t_{sep}$
	1000	620	0	$t_{fill} \leq t \leq t_{sep}$
$\Gamma_{bottom2}$	20	75	0.7	$t < t_{fill}$ or $t > t_{sep}$
	100	700	0	$t_{fill} \leq t \leq t_{sep}$
$\Gamma_{cooling}$	20	200	0	when not active
	200	100	0	when active
Γ_{holder}	1000	400	0	$t > 0$
$\Gamma_{D-C}(S\#1-8)$	20	50	0.5	$t > t_{eject}$
Γ_{D-S}	20	50	0.7	$t > t_{sep}$
Γ_{C}	20	50	0.7	$t > t_{eject}$

Notes:
Surfaces marked with Γ are indicated in Figure 2.1;
$\Gamma_{cooling}$ are the cooling channels, $Ch\#1-4$;
t_{fill}, t_{sep} and t_{eject} refers to filling, separation, and ejection times which are marked in Figure 2.2.

equivalent radiative heat transfer coefficient is calculated using [1]:

$$h_{rad} = \sigma\varepsilon(T_{surf}^2 + T_{amb}^2)(T_{surf} + T_{amb}) \tag{2.4}$$

where σ is the Stefan-Boltzmann constant (5.669×10^{-8}W/m^2/K^4) and ε is the emissivity of the surface. The parameters of the forced-air cooling and exterior boundary conditions are summarized in Table 2.4. Note that the parameter of ambient temperature needs to be converted into the Kelvin scale (K) when used in Equation (2.4). The cycle timing parameters referenced in Table 2.4 are defined on the schematic plot of the die temperature variation during an experimental casting cycle shown in Figure 2.2.

2.2.5 Liquid Encapsulation Prediction

The predicted casting temperature can be used to calculate if / when liquid encapsulation occurs in each casting cycle and the volume of liquid that is encap-

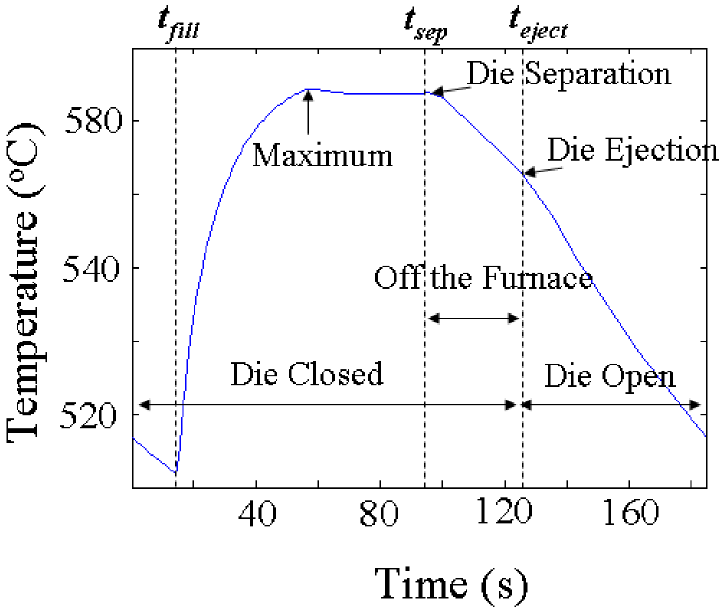


Figure 2.2: Schematic of Single Cycle Die Temperature Variation in the Trials.

sulated. A post-processing program was developed to determine when a portion of the casting is isolated from the liquid metal supply of the sprue. This involves performing a search to identify all of the nodes in the geometry representing the casting, with temperatures higher than a critical temperature for feeding, which are not connected to the sprue(liquid metal source) by a path of interconnected nodes. The critical temperature for feeding was assumed to be 568°C based on a critical solid fraction of 0.8 [86]. The nodes identified in this manner represent a pocket of encapsulated liquid metal which have the potential to form macro-porosity as the isolated liquid metal solidifies.

Liquid metal encapsulated is dependent on the both the casting and die temperature. In the casting used for this study, encapsulation has been observed to occur in the upper section, in the lower section, and concurrently in both sections. An example of liquid encapsulation occurring in the upper section is presented in Figure 2.3. In Figure 2.3, the red regions represent metal with temperatures greater than the critical temperature and the blue regions represent metal at temperatures below the critical temperature. The sequence of liquid metal encapsulated evolves during a casting cycle and may (or may not) include: i) the encapsulation of a volume of liquid in the upper section (refer to Figure 2.3a; and ii) as solidification proceeds, encapsulation of a volume of liquid in the lower section (refer to Figure 2.3b. The total volume of encapsulated liquid is the sum of the maximum volumes in the upper and lower sections. The volume of each encapsulated liquid region as well as the summation are calculated for use as an indicator of the potential extent of macro-porosity.

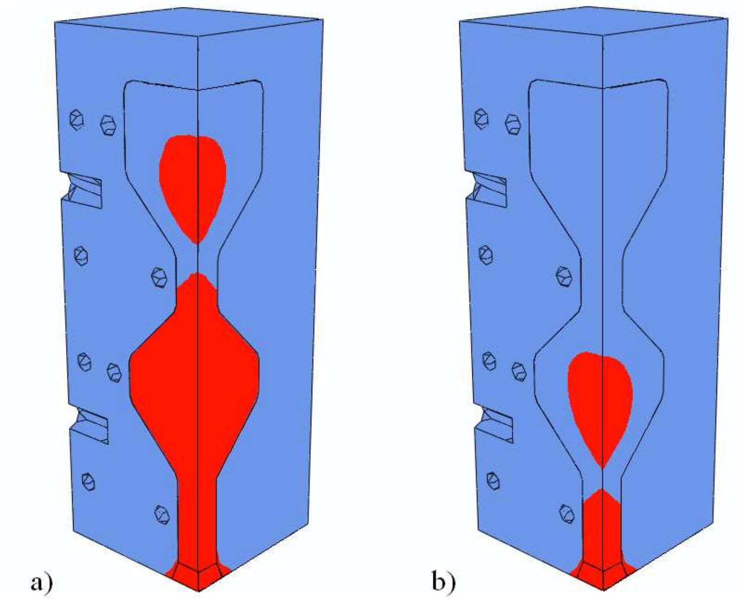


Figure 2.3: Predicted Liquid Encapsulation Evolution (a) with Initial Passage Block in the Upper Section and (b) with Latter Passage Block in the Lower Section.

2.3 Model Validation

Model validation is an essential task that was completed at an early stage in the research to tune the boundary conditions of the FE model and to assess the accuracy of the predictions for temperatures and liquid encapsulation. Once validated, execution of the process model was automated through the use of a Perl script wrapper to create a ’virtual process’ for use in designing and testing an advanced process control solution.

2.3.1 Low Pressure Die Casting Trials

To provide the data necessary to fit the boundary conditions of the model and validate the temperature predictions, a plant trial was conducted at NRCan’s CANMET-MTL LPDC facility, previously located in Ottawa, Canada. The LPDC machine at this facility was adapted for casting light metals and its operation is atypical in that the die lifts off the holding furnace each cycle and rotates to the side prior to ejection. A casting cycle begins when the die closes and rotates; placing it in contact with the holding furnace. Metal is forced up a joint pipe and into a sprue to fill the die when the air above the liquid metal in the holding furnace is pressurized. The casting solidifies as heat is transferred to the die. After a predetermined time, the pressure in the holding furnace is released and the die is then rotated away from the holding furnace. The die is opened and the casting is removed manually. Prior to the start of the next cycle, the die remains open and cools. Four cooling channels were placed in the die, in the locations shown in Figure 2.1, to provide cooling to the die. Air or water could be used as coolant in the cooling channels depending on desired cooling rates.

A LabView program was developed to monitor and record thermocouples and to control the cycle timing for die cooling. The die was instrumented with Type-E thermocouples at various locations to measure temperatures during each casting cycle. A coolant distribution system, shown in Figure 2.4, was fabricated by connecting 4 coolant supply lines to a manifold. Each supply line was split to provide coolant to each half of the die. The flow of coolant (cycle start time and duration of cooling) in the supply lines was controlled with solenoid valves that were linked via a relay board to the control program. Coolant flow rate in the cooling channel was preset by regulating a ball valve installed in-line with the solenoid valve.

The plant trials conducted for this investigation used a die based on the geometry shown in Figure 2.1. The die temperatures were measured at 8 locations in the die via Type-E thermocouples mounted 5 mm below the die surface with a sam-

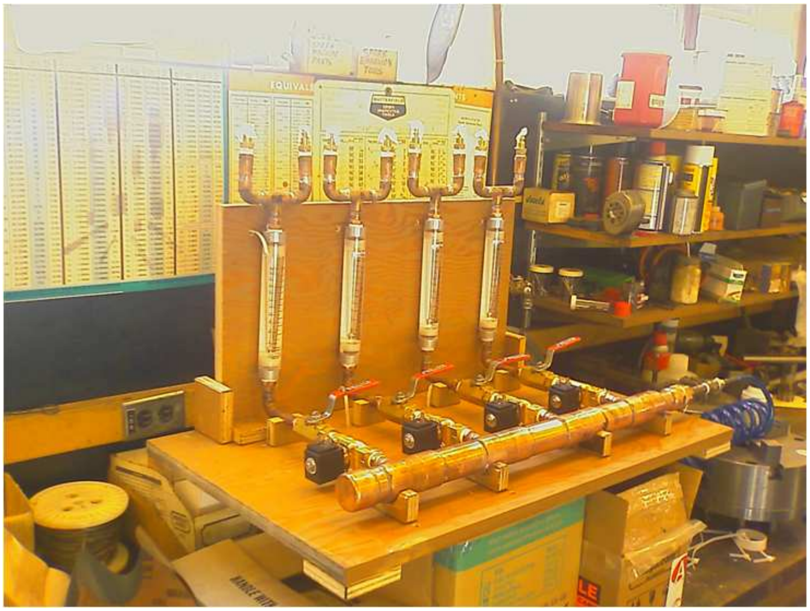


Figure 2.4: The Manifold Used in the Trials.

pling rate of 1 Hz. A series of 12 casting trials were conducted in open loop mode with the casting process-related cycle timing performed manually. Each casting trial condition was run until cyclic steady state was achieved: when the measured die temperatures at the start and end of a casting cycle were within 1°C (a minimum of 10 shots). When planning the plant trial, a start and end temperature difference of 0.1°C was targeted to define steady state operation. However, in practice, the ideal steady state was not achieved. The requirement to perform cycle timing manually and the frequent process disturbances which occurred during each trial made this goal impractical. Figure 2.5a shows the die temperature history from one thermocouple. Figure 2.5b presents an enlarged plot of the 13th cycle's die temperature variation which met the steady state condition requirement. Careful observation of the data shown in Figure 2.5a reveals a small cycle-to-cycle fluctuation in the Maximum Temperature as this location approaches steady state. This variation was likely caused by cycle time variability caused by manual operation.

From the 12 trials, 3 representative trial conditions, summarized in Table 2.5,

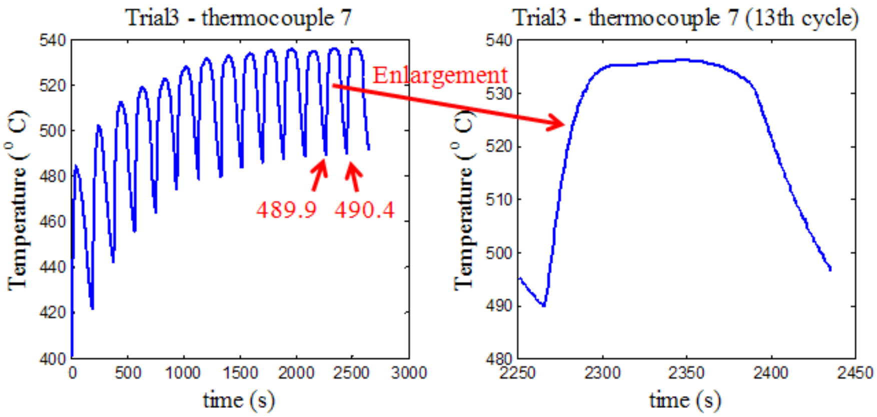


Figure 2.5: (a) A Sample of Die Temperature's Evolution in a Thermocouple Until Steady State and (b) an Enlarged Plot of a Single Cycle Die Temperature Variation at Steady State.

Table 2.5: Experimental Process Parameters.

Trial No.	Die Closed Time(s)		Die Open Time (s)	Cooling Channel 1		Shots No.
	Pressure* on	Pressure off		Time (s)	Flow rate (L/min)	
3	30	90	45	-	-	20
4	30	90	60	-	-	13
6	30	90	45	120	600	15

* When Pressure is on, the bottom of the casting was connected with the liquid metal supply of the sprue, and vice versa.

were selected for use in this investigation to examine the effects of Die Closed Time, Die Open Time and Cooling Duration (only Cooling Channel 1 was active).

2.3.2 Results

The predicted (red lines) and measured (blue symbols) die temperatures at 3 of the 8 thermocouple locations are compared in Figure 2.6 for representative steady state casting cycles from casting trials 3, 4 and 6 (refer to Table 2.5 for details of process parameters). Among the results for each trial condition, the temperature at TC#2 was the lowest of the three thermocouple locations reported because this location was both closest to the exterior mounting surface of the die where heat is lost to the casting machine and closest to the cooling channel activated in Trial 6. Conversely, the highest temperatures were measured by TC#5 because it was furthest from the external surface of the die and in a location surrounded by considerable thermal mass. Considering the effect of operational conditions for a given die location, the subplots in Figure 2.6 indicate that increased Die Open Time (Trial 4) and the activation of forced-air cooling (Trial 6) result in decreased die temperatures. The largest decrease in die temperature, and the most localized effect, was measured at TC#2 when forced-air cooling activated because this location was nearest to the active cooling channel.

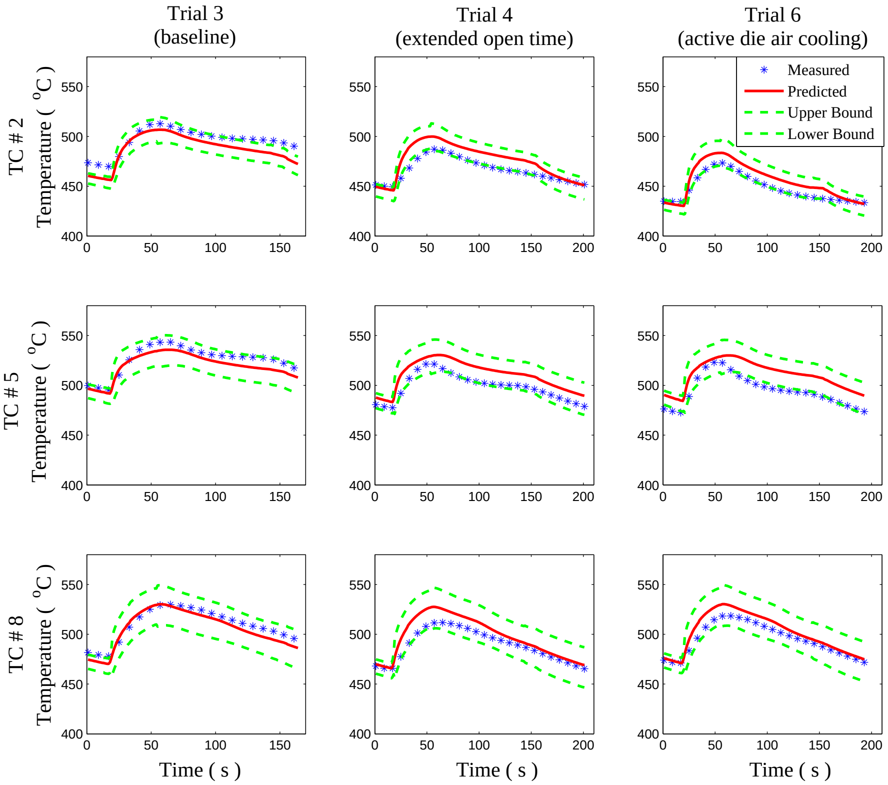


Figure 2.6: Graphical Comparison of Predicted and Measured Data with the Consideration of Uncertain Factors.

Despite attempts to alter the boundary conditions to improve the temperature predictions, differences still exist between the predicted (red line) and measured temperatures (blue symbols) presented in Figure 2.6. The discrepancy between the predicted and measured data may be due to uncertainty and/or variability in the operational conditions achieved in the trial which have not been accounted for in the model. In an attempt to assess the sensitivity of the model predictions, an uncertainty analysis has been performed. After systematically varying the model input parameters, potential errors contributing to discrepancy between the predicted and measured temperatures are:

1. *Random errors*

   (a) μ_1: the error from varying forced-air cooling heat transfer coefficient (h = 10-20 W/m^2/°C).

   (b) μ_2: the error from varying thermocouple position (uncertainty in each direction $\pm$ 2mm).

   (c) μ_3: the error from varying contact time between casting and die during the process.

2. *Systematic error*

   (a) μ_4: the error from cycle timing uncertainty ($\pm$ 2s).

Following a statistical approach to estimate the propagation of uncertainty in this system [87], the combined error due to uncertainty in the identified parameters can be calculated as the Root Mean Square (RMS) of the three independent random error factors, expressed as follows:

$$\mu_{1-3} = \sqrt{\mu_1^2 + \mu_2^2 + \mu_3^2} \tag{2.5}$$

where $\mu_2 = \sqrt{\mu_{2x}^2 + \mu_{2y}^2 + \mu_{2z}^2}$, and μ_{2x}, μ_{2y}, μ_{2z} are the directional components of μ_2 along x, y, and z directions. Based on the composite errors of both the random and systematic errors [87], upper (T(t)$^+$) and lower (T(t)$^-$) bounds of the model predictions have been defined as follows:

$$T(t)^{\pm} = T(t)_{pred} \pm \mu^{\pm}_{1-3(t)} \pm \mu^{\pm}_{4(t)} \tag{2.6}$$

where $T(t)_{pred}$ is the die temperature predicted by the model, $\mu^+_{i(t)}$ and $\mu^-_{i(t)}$ are the positive and negative effects of μ_i on the die temperature, as a function of time t, respectively.

The variation of the temperature predictions due to the identified uncertainty was calculated with Equation (2.6). The lower and upper bounds of the temperature predictions are plotted in Figure 2.6 as green dashed lines. For the most part, the measured temperatures fall within the temperature range defined by these curves indicating that identified uncertainty factors may explain the differences between the predicted and measured temperature data. Based on this analysis and the comparison between the predicted and measured temperatures, the model is assumed to adequately describe the casting process enabling its use in further analysis.

Following the casting trials, sample castings collected during the trials were sectioned in half vertically, mounted in epoxy, and the cut surface was ground. The macro-porosity in these samples was quantified by imaging the ground surface. Figure 2.7 shows the cross-section of a casting sample with overlaid contours of the measured porosity distribution for a casting sample from Trial 6. The majority of the porosity observed in this casting and the others observed occurred due to liquid encapsulation and shrinkage effects. The size of the encapsulated region and the extent of porosity depend on the coherency of the solidified region which block off the incoming liquid and the extent of solidification in the overall casting. The asymmetric distribution of macro-porosity in the upper region of the casting shown in Figure 2.7 may be caused be uneven shell thickness formation due to non-symmetric heat transfer and process variability. The macro-porosity distribution is symmetric in the low section. In the example casting shown, a surface depression, caused by the internal pressure in the casting, was observed. The formation of the defect may have influenced the macro-porosity distribution in the upper section.

2.4 Virtual Process

The ABAQUS model developed in section 2.2 can be used to predict the evolution of the casting and die temperature distribution for a single casting cycle. This section outlines the system developed to transform the ABAQUS simulation into a virtual process. A Perl language-based wrapper script was written to automate the

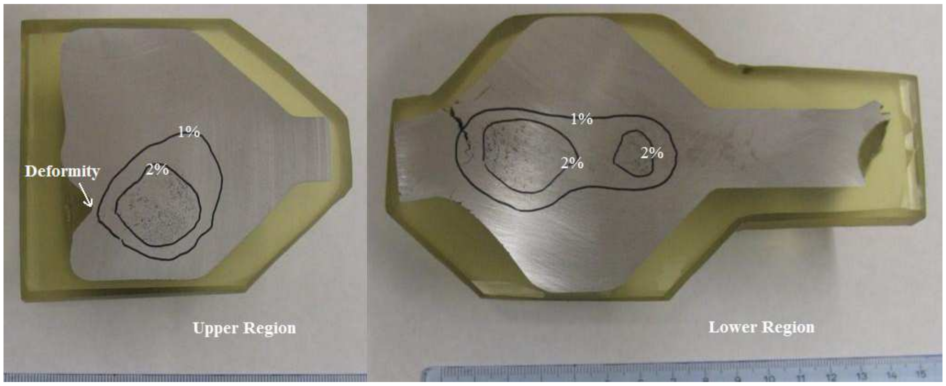


Figure 2.7: Section View of the Asymmetric Distribution and the Extent of Macro-porosity for the Experimental Casting.

model allowing it to run continuously like an operating casting process. The wrapper communicates with and controls the model in a manner similar to an industrial controller and a casting machine. Operating in this manner, the process model acts as a virtual process which runs continuously and can be programmed with varying input conditions.

2.4.1 Virtual Process Operation Mode

The core component of the system is a Perl script that manages the virtual process's communications while repeatedly running the single cycle ABAQUS simulation. The virtual process can be operated in two modes: open-loop (uncontrolled) and closed-loop (controlled). The open-loop functionality has been used to assess the influence of process variables and to perform system identification, while closed-loop operation provides a platform to verify the performance of the designed controller. In closed-loop, based on virtual process data, the controller computes the manipulated (controllable) process inputs and applies them to the current cycle's simulation. The virtual process operational cycle and data communications are illustrated in Figure 2.8.

As is shown in Figure 2.8, during each cycle the Perl script executes the following three major tasks which are described in the subsequent sections:

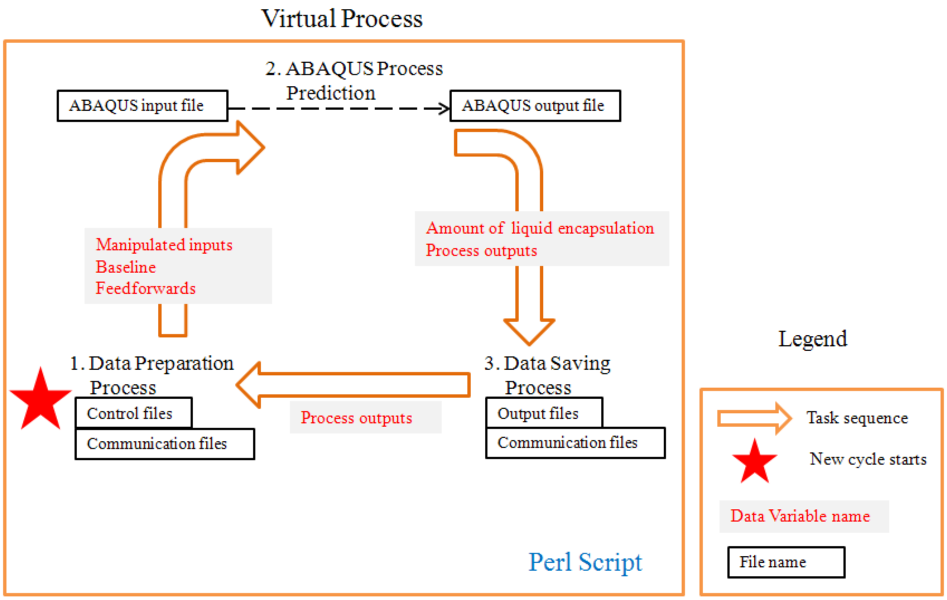


Figure 2.8: Data Flow in a Typical Virtual Process Simulation Cycle.

- Retrieve new cycle's process data for data preparation.
- Compile the input file and run the ABAQUS simulation.
- Extract virtual process data and save the ABAQUS simulation output files.

The perl wrapper script and examples of the control files are provided in Appendix A.

2.4.2 Data Preparation Process

During the data preparation process, the Perl script collects the process parameters necessary to run the simulation for the current cycle. The data collection modes for both open loop and closed loop are a slightly different. The common operation for both is that, at the beginning of each cycle, data is read from the following separate control files:

1. $Simulationmode.input$ – specifies whether the virtual process operation mode is open loop or closed loop during each simulation cycle.

2. $Feedforward.input-$ specifies the values of the measured disturbances during each simulation cycle.

3. $Baseline.input-$ specifies the baseline values of the process outputs during each simulation cycle.

These control files are text files that list the cycle number and that cycle's associated parameters on the same lines. At the beginning of each cycle the Perl script parses the control files to determine the parameters for the current cycle. If there is no definition for the current cycle, the last cycle's data is used. Thus, input data is only required for cycles that are different from a previous cycle.

The value returned from the simulationmode.input file specifies whether the virtual process operation mode is open or closed loop and then determines the source from which the manipulated (controllable) process parameters are read. A manipulated parameter is defined as the independent variable subject to the controller action.

The data flow in the data preparation process is presented in Figure 2.9. When the simulation mode is open loop, there is no controller action involved. Therefore manipulated process inputs are directly read from the predefined controlvariables.input file. When the simulation mode is closed loop, the manipulated process inputs in the current cycle are computed by the controller. At the end of the last cycle, the Perl script has written the previous cycle's process outputs that were extracted from ABAQUS simulation files to the communication file. When the new cycle starts, the Perl script extracts the previous cycle's process outputs from the communication file and retrieves the new cycle's baseline and feedforward inputs from control files. Once the data are ready, the Perl script writes a trigger to notify the controller running in MATLAB to compute the manipulated process inputs.

2.4.3 ABAQUS Process Prediction

ABAQUS employs a text-based input file to describe a single casting cycle's conditions. The input file contains model information about part geometries, material properties and initial conditions. It also contains a series of analysis steps that define the boundary conditions and contact conditions for each stage in the casting cycle such as die closes or die opens.

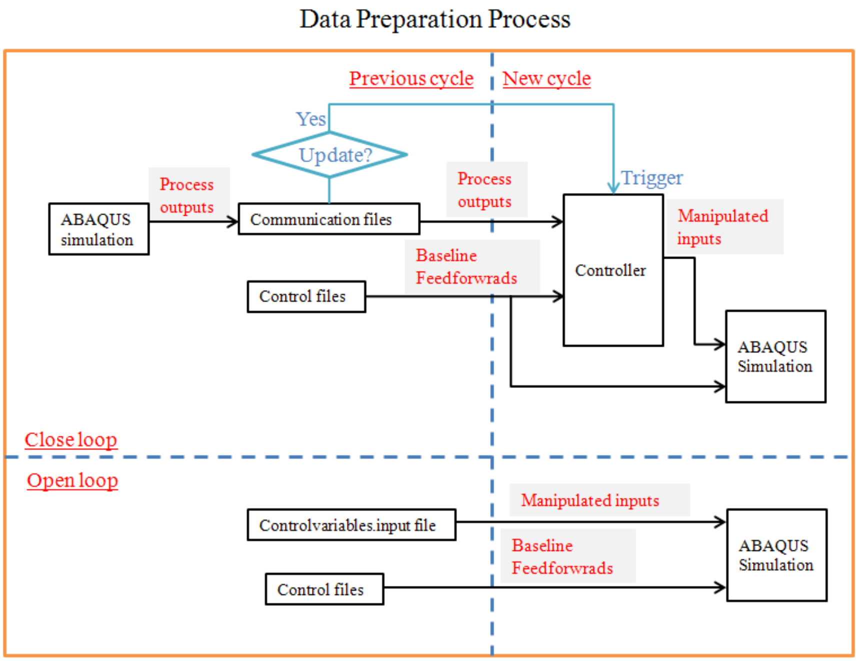


Figure 2.9: Data Flow in the Data Preparation Process.

In order to simulate continuous cycling with the ABAQUS model, the simulation parameters in the input file may be changed from cycle to cycle according to the task's requirement. To facilitate these changes, input parameters are replaced with identification ”tags” that are replaced when the input file is compiled prior to running the simulation. A new input file is compiled for each cycle with the specified simulation parameters. Table 2.6 lists general names of tags found in the ’cycle.tag’ file and their corresponding definitions in the simulation.

Another tag file, named ’main.tag’, contains the input parameters that define the cooling channel boundary conditions. The tags for the user subroutine **FILM**, as shown in Table 2.7, set the timing for coolant flow in the cooling channels.

2.4.4 Data Extraction Process

At the end of the ABAQUS process simulation for each cycle, a corresponding results file with a suffix of ’fil’ is generated from the ABAQUS process predictions. It

Table 2.6: Names of Tags and Their Definitions in the Cycle.tag File.

Tag Name	Definition
$<$ CASTDIETEMPS $>$	Initial conditions for casting and die temperatures
$<$ SIMPLEBDFORSPRUE $>$	Temperature constraint for bottom surface of the sprue
$<$ BOUNDARYCASTTEMP $>$	Incoming casting temperature during the filling time
$<$ DIECLOSEDTIME $>$	Die closed time
$<$ DIEOPENTIME $>$	Die open time

Table 2.7: Names of Tags and Their Definitions in the Main.tag File.

Tag's Name	Definition
$<$ HEATCOEFFICIENT1 $>$	Starting time and duration for activated Cooling Channel1
$<$ SHEATCOEFFICIENT2 $>$	Starting time and duration for activated Cooling Channel2
$<$ HEATCOEFFICIENT3 $>$	Starting time and duration for activated Cooling Channel3
$<$ HEATCOEFFICIENT4 $>$	Starting time and duration for activated Cooling Channel4

contains the temperature history for all nodes in the analysis at all times in a cycle. By running a user-written extraction program, the process outputs are extracted from the above results file and saved to output files for later use in system identification or for input to a controller. The extraction program also computes the extent of liquid encapsulation based on the casting temperature evolution during solidification. The ABAQUS process prediction file is also archived to facilitate additional post-processing.

Chapter 3

Developing and Applying Die Temperature - Liquid Encapsulation Correlations[1]

In the LPDC process minimizing macro-porosity can be a important production issue since this defect can cause castings to be rejected due to its deleterious effect on the mechanical properties and surface quality. The ability to identify and quantify the volume of macro-porosity formed during casting is a pre-requisite for conducting research on methods to reduce its formation. Non-destructive measurement techniques such as X-ray imaging have not been used in this research due to their costly uses and inaccurate measures of macro-porosity as is described in Chapter 1. Instead die temperatures will be used as an indirect indicator of macro-porosity because of the relative ease with which temperature data can be obtained from an operational LPDC process. Most industrial casting operations monitor temperature at a variety of locations in the die as part of the standard operating

[1] Portions of this chapter have been published in:
• X. Shi, D.M. Maijer and G. Dumont, "Determination of Optimal Location to Monitor Temperature During Low Pressure Die Casting", Light Metals 2010 Proceedings, Advances in Materials and Processes, COM2010, Vancouver, BC, p.3-13, Oct 2010.
• X. Shi, D.M. Maijer and G. Dumont, "Determination of the Optimal Location to Monitor Temperature in a Low Pressure Die Casting Process", Materials Science and Technology, UK, Vol 27, No 6, p1073-1083, 2011.

procedure. It is assumed that the die temperatures will bear close relationship to liquid encapsulation, a precursor to the formation of macro-porosity.

This chapter uses the virtual process, described in Chapter 2, to develop correlation method between die temperature and the volume of encapsulated liquid and then to analyze its application. The correlation method will then combine with a Linear Regression (LR) method to determine the optimal die locations to monitor temperature for the purpose of minimizing macro-porosity through controller design. The virtual process, which simulates an operational casting process, predicts the temperature distribution of the casting and die in the LPDC process. A post-processing program has been developed to calculate the amount of liquid that is encapsulated using the predicted casting temperature history. This framework can be used to conduct multiple simulations with various sets of process parameters at a time.

3.1 Data Generation

Before generating data for use in developing correlations for liquid encapsulation, it was necessary to determine the largest contributing parametric factors to liquid encapsulation and then to identify the associated operational ranges through which these factors can be varied. Following a brief sensitivity analysis, nine process parameters that affect the volume of liquid encapsulation were selected: Die Closed Time and the operational parameters (Cooling Start Time and Cooling Duration) for the 4 cooling channels. The commonly used techniques to assess the effects of manipulated parameters on the process are trial-and-error [16] or factorial design [88]. However, full factorial design at 2 levels for 9 parameters would require 2^9 different combinations to be simulated with the virtual process. Each combination takes more than 10 simulated cycles to reach cyclic steady state. Since one simulated cycle consumes one hour using the current computational resources and version of ABAQUS, the total computational time to perform a full factorial analysis with the virtual process would be at least 5120 hours.

To save on simulation time, the Taguchi method [89–91], an experimental optimization method, was applied. In this study, an orthogonal array containing 16 combinations was designed to investigate the influence of the manipulated variables on the volume of encapsulated liquid. Each combination of parameters was simulated sequentially in the virtual process and until the cyclic steady state was achieved before moving to the next parameter combination. The volume of liquid encapsulation was computed for each cyclic steady state condition using the post-processing program described in subsection 2.2.5. The results indicated that turning on the cooling channels $2-4$ increased the volume of liquid encapsulation. Since cooling from these channels was not effective in minimizing liquid encapsulation, they were not considered further. Two variables were found to greatly affect the magnitude of the encapsulated liquid volume: Die Closed Time and Cooling Duration for Channel 1 (when water cooling is considered). Hence, these two variables were identified as control inputs with potential to aid in reducing macroporosity. The remaining discussion and analysis presented in this chapter will be limited to these variables.

To explore the relationship between die temperatures and the volume of encapsulated liquid, it was necessary to generate detailed data over the operational range of the control parameters identified through the Taguchi analysis. The virtual process was run for a range of process operational conditions to provide this data. The process inputs of Die Closed Time and Cooling Duration were varied from 105 to 205 s in increments of 5 s and from 0 to 24 s in 3 s increments, respectively. Die Open Time was held constant at 60s. The combination of these parameters represents 189 (21×9) operational conditions. For each condition, the virtual process was run until cyclic steady state was achieved. For this portion of the study, cyclic steady state was defined as less than a 0.01°C difference between a cycle's start and finish temperatures throughout the die.

From the cyclic steady state result of each process condition, the volume of encapsulated liquid was calculated and the temperature history at each thermocouple location was extracted in a post-processing operation. The volume of encapsulated liquid, defined as the maximum total amount of encapsulated liquid that occurred during solidification, was used as an indicator of the potential extent of macroporosity. The die temperature data was further processed to extract two characteristic temperatures per cycle, the Maximum Temperature and the Ejection Temperature (refer to Figure 1.2b), for correlation with the volume of encapsulated liquid. The Maximum Temperature is the peak temperature reached at the die location being considered during a cycle, while the Ejection Temperature is the temperature measured when the die opens to release the casting. These temperatures were selected for evaluation because they are representative points in the casting cycle that are easy to identify. The extracted data were used for further analysis on the relationship between die characteristic temperatures and the volume of encapsulated liquid.

3.2 Correlation Method

A methodology has been developed to analyze the correlation between die temperatures and the volume of encapsulated liquid. A Correlation Index (*CI*) can be calculated from the die temperature and volume of encapsulated liquid data at a particular die location which represents the average of the correlation value over the operational range. The Standard Deviation (*STD*) of the *CI* has been calculated as an auxiliary measure of the variability between the die temperature and the volume of encapsulated liquid over the operational range. The algorithms for both approaches are detailed as follows.

In Figure 3.1, the response surfaces of the volume of encapsulated liquid and Maximum Temperature are overlaid for an upper and a lower die location (TC #2 and TC #8 in Figure 2.1). The response surface of a parameter is made up of the contours of the parameter over the operational range. The response surface of the volume of encapsulated liquid, which is location independent, exhibits a minimum of volume when the Die Closed Time is about 190 s and Cooling Duration is 0 s - i.e. no cooling. The Maximum Temperature at both thermocouple locations generally increases with decreasing Cooling Duration. At the lower die location (TC #8), the response surfaces of the Maximum Temperature and the volume of encapsulated liquid show better correlation in the upper right operational region above the diagonal compared to the bottom left operational region below the diagonal. However, the response surface of Maximum Temperature for the upper die location (TC #2) exhibits less curvature. Thus, there is only a small region (along the diagonal defined by the minimum volume of encapsulated liquid vs. Die Closed Time) of good correlation at this location.

The correlation at a specific operating point is calculated between the gradient

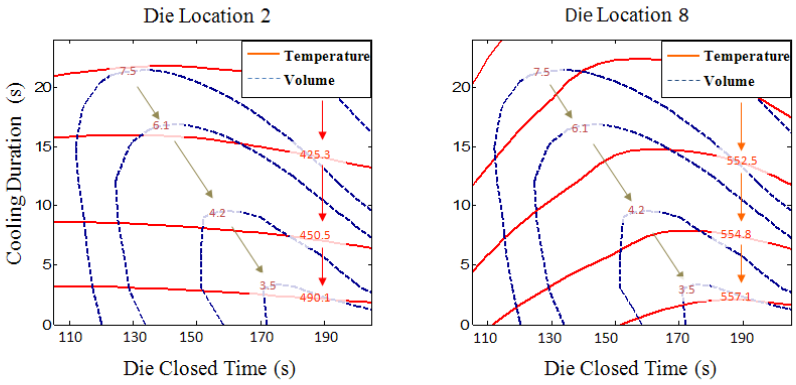


Figure 3.1: Response Surfaces (Contours) of Maximum Temperature and Volume of Encapsulated Liquid for Die Locations Corresponding to TC #2 and TC#8.

of Maximum Temperature and the negative gradient of the volume of encapsulated liquid, where the gradient of a scalar field at one point is mathematically defined as the direction in which the parameter rises most quickly.

The methods for computing the gradient of both Maximum Temperature and the volume of encapsulated liquid are the same. For example, the gradient angle of Maximum Temperature at an operating point is cacluated as follows,

- Locate the operating point to be considered and its adjacent points on X-Y coordinates that is covered by the operational range (refer to Figure 3.2), and then label the variables of $(T_{y1}, T_{y2}, T_{x1}, T_{x2})$ for the Maximum Temperatures of the adjacent operating points as shown in Figure 3.2.

- Identify which quadrant contains the gradient direction of Maximum Temperature response surface at the operating point by simple comparison between T_{y1} and T_{y2} and between T_{x1} and T_{x2}. For example, if $T_{y2} \geq T_{y1}$ and $T_{x2} \geq T_{x1}$, the gradient direction will fall into the first quadrant.

- Define the temporary angular variable θ_{abs} as $\tan^{-1} | \frac{T_{y2}-T_{y1}}{T_{x2}-T_{x1}} |$, where $\theta_{abs} \in [0, 90]$.

- The gradient angle $\theta_T, \in [0,360]$, is computed as follows:

$$\theta_T = \begin{cases} \theta_{abs}, & \text{if } \theta_T \in \text{the first Quadrant} \\ 180-\theta_{abs}, & \text{if } \theta_T \in \text{the second Quadrant} \\ 180+\theta_{abs}, & \text{if } \theta_T \in \text{the third Quadrant} \\ 360-\theta_{abs}, & \text{if } \theta_T \in \text{the fourth Quadrant} \end{cases}$$

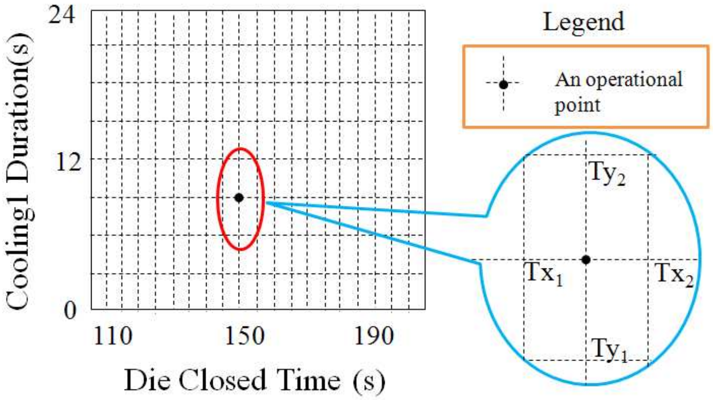


Figure 3.2: Schematic Pot of an operating Point and Its Adjacent Points.

A metric to quantify the correlation between the responses of both parameters at an operating point has been developed. *Correlation* is computed based on the differences in the local normal of the two response surfaces at this operating point according to:

$$Correlation = (1 - \frac{\theta_d^i}{180}) \times 100 \qquad (3.1)$$

where θ_d^i is the angular difference of θ_v^i and θ_T^i, $\theta_d^i \in$`[0,180]` . θ_v^i is the angle of negative gradient of the volume of encapsulated liquid response surface at the ith operational condition and θ_T^i is the angle of gradient of the die temperature response surface at the ith point, θ_v^i and $\theta_T^i \in$`[0,360]`. Specifically, θ_d^i is computed as follows:

$$\theta_d^i = \begin{cases} \mid \theta_T^i - \theta_v^i \mid, & \text{if } \mid \theta_T^i - \theta_v^i \mid \ < 180 \\ 360 - \mid \theta_T^i - \theta_v^i \mid, & \text{if } \mid \theta_T^i - \theta_v^i \mid \ \geq 180 \end{cases}$$

The quantities calculated are shown graphically in Figure 3.3. The correlation will result in a calculated value of 100 when the response surfaces are identical at an operating point, *i.e.* where there are no differences between the gradient of Maximum Temperature and the negative gradient of the volume of encapsulated liquid.

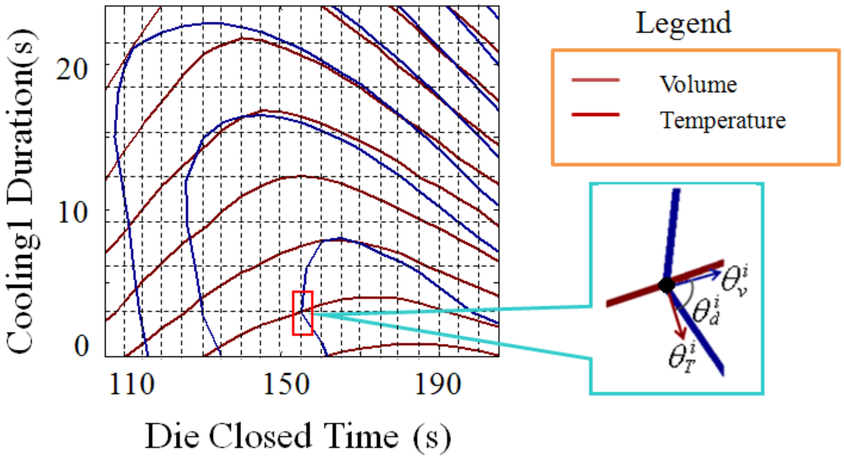


Figure 3.3: Schematic Plot of the Response Surfaces for Maximum Temperature and the Volume of Encapsulated Liquid for the Operational Region and the Normals of the Two Response Surfaces at an operating Point.

The correlation metric can be calculated at every operating point. When considering the entire operational range, the average of the correlation has been defined as the ”Correlation Index”(*CI*), which is expressed as:

$$CI = \frac{\sum_{i=1}^{N} Correlation}{N} = \frac{\sum_{i=1}^{N} (1 - \frac{\theta_d^i}{180})}{N} \tag{3.2}$$

where N is the number of operational conditions.

A perfect *CI* (i.e. *CI* approaching 100) at a die location indicates that the response surfaces for both parameters are identical over the whole operational range. Based on the response surfaces considered in developing this analysis technique, it is unlikely that a perfect *CI* will occur.

The *CI* calculated at a given die location represents the average correlations of the temperature and encapsulated volume response surfaces over the operational range considered. The standard deviation of the angular differences of two surfaces at a given location can be calculated for use as a measure of the variability of correlations over the operational range. The standard deviation of the correlations at a die location over the operational range is computed as follows:

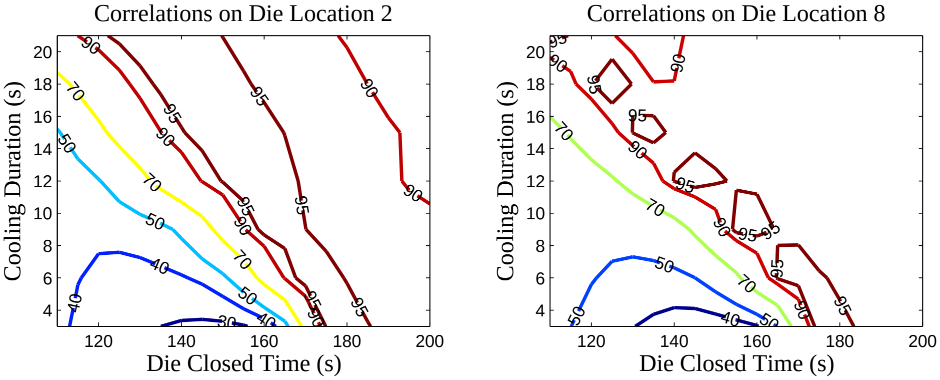
$$STD = \sqrt{\frac{\sum(\theta_d^i - \mu)^2}{(N-1)}}$$

where μ is the mean of θ_d^i and N is the number of operational conditions.

Contours of the correlation values, computed by Equation (3.1), between the volume of encapsulated liquid and Maximum Temperature at die locations corresponding to TC #2 and TC #8 are presented in Figure 3.4. The correlation values at TC #2 vary from $\sim$ 30 to $\sim$ 95 as shown in Figure 3.4a, while the correlation values at TC #8 vary from $\sim$ 40 to $\sim$ 95 as shown in Figure 3.4b. The *CI* and *STD* computed at TC #2 are 77.5 and 23.3 respectively, while *CI* and *STD* at TC #8 are 81.5 and 19.2 respectively.

3.3 Application of the Correlation Method

As mentioned in section 3.2, the *CI* and *STD* are calculated at a given die location from the temperature and encapsulated volume response surfaces over the operational range considered. In this subsection, *CI* analysis will first be used to evaluate the 8 thermocouples locations to determine the best for correlating die temperature and the volume of encapsulated liquid. Extending this approach, the *CI* and *STD* values will then be calculated for all the die locations to obtain the *CI* and *STD* distribution over the whole die. The distribution of *CI* can then be used to help determine the die location that exhibits the best correlation between die tempera-



(a) **(b)**

Figure 3.4: Contours of Correlations Between the Volume of Encapsulated Liquid and Maximum Temperature at Die Locations Corresponding to TC #2 and TC #8.

ture and the volume of encapsulated liquid, while the distribution of *STD* will help determine the die location that has the least correlation variability.

Two die temperatures will be compared from representative points in the casting cycle that are easy to identify: Maximum Temperature (peak temperature reached during a cycle) and Ejection Temperature (measured when the die opens to release the casting). Two measures of the volume of encapsulated liquid will be assessed in this subsection: i) the maximum volume of encapsulated liquid occurring in the casting (upper and lower sections) and ii) the volume occurring in the upper section of the casting.

CI values at the 8 locations where thermocouples were installed in the trial die are summarized in Table 3.1. The data indicate that the *CI* for the Maximum Temperature is higher than that for the Ejection Temperature for all locations except at TC #8. The *CI* calculated with Maximum Temperature shows a trend of gradual increase from TC #1 to TC #8, whereas the *CI* for the Ejection Temperature does not show a consistent trend. Table 3.1 suggests that the best location to monitor temperature for correlation with the volume of encapsulated liquid, is TC #8 near the bottom of the die.

The contour plots of the *CI* calculated throughout the die for the two candidate die temperatures: Maximum Temperature and Ejection Temperature are shown in Figure 3.5. In Figure 3.5, the entire casting volume of the encapsulated liquid was considered. The Maximum Temperature *CI* is highest ($\sim$ 84.25) at the bottom of the die, near the metal inlet on the casting/die interface, close to the parting line. The *CI* decreases with increasing distance from this location and reaches its lowest value ($\sim$ 77.00) at the top of the die. The *CI* calculated for the Ejection Temperature exhibits a similar variation albeit shifted to lower values (*CI* ranging from 70 to 82). Additionally, there is a region of low *CI* present on the casting / die interface in the lower enlarged section of the die. For both candidate die temperatures, the highest values of *CI* are observed at the bottom of the die, near the casting inlet.

Table 3.1: *CI* Calculated at TC Locations.

TC #	Correlation Index	
	Maximum Temperature	Ejection Temperature
1	77.20	75.95
2	77.50	77.14
3	77.78	76.42
4	77.78	75.41
5	78.24	76.86
6	79.91	75.25
7	81.38	74.34
8	81.50	83.84

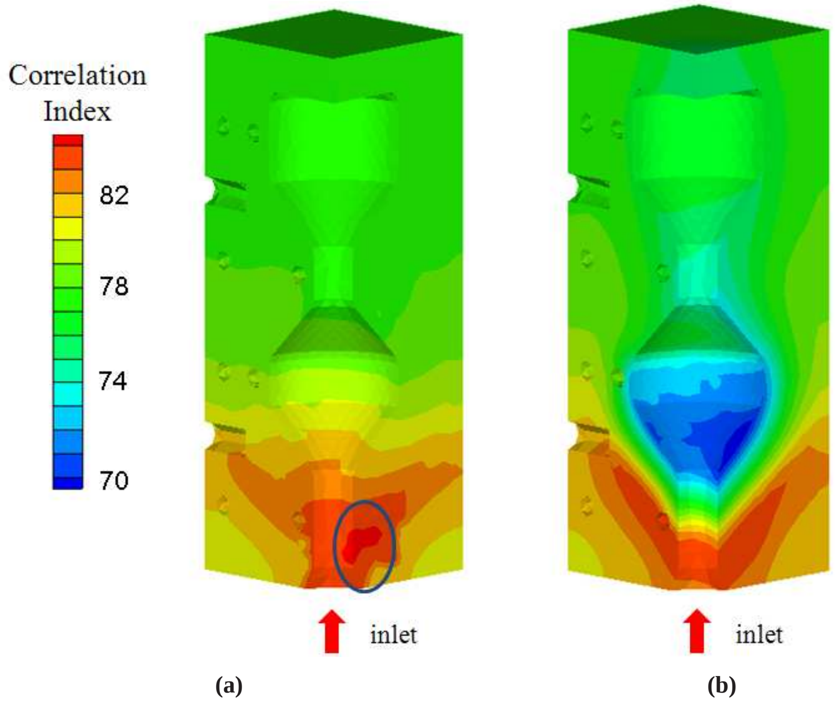


Figure 3.5: Contour Plots of the *CI* Based on Two Candidate Temperatures: (a) Maximum Temperature and (b) Ejection Temperature with the Total Volume of Encapsulated Liquid (refer to Figure 2.3).

Figure 3.6 shows the contour plots of the standard deviation of *CI* for the two candidate die temperatures. The bottom of the die has the smallest *STD* of *CI* for both temperatures, which means that the *CI* in this area has the smallest dispersion over the operational range considered. In other words, the *CI* at the bottom of the die is the least sensitive area to variability in the operational conditions. Thus, the bottom of the die near the casting/die interface exhibits both the highest *CI* and the smallest *STD* indicating this is the best location to monitor temperature for correlation to liquid encapsulation within the entire volume of the casting. Considering the casting geometry, this result is expected since this location is near the metal inlet which is the most critical location for liquid encapsulation in the entire casting volume. However, the methodology and metric employed enable a quantitative determination of this location.

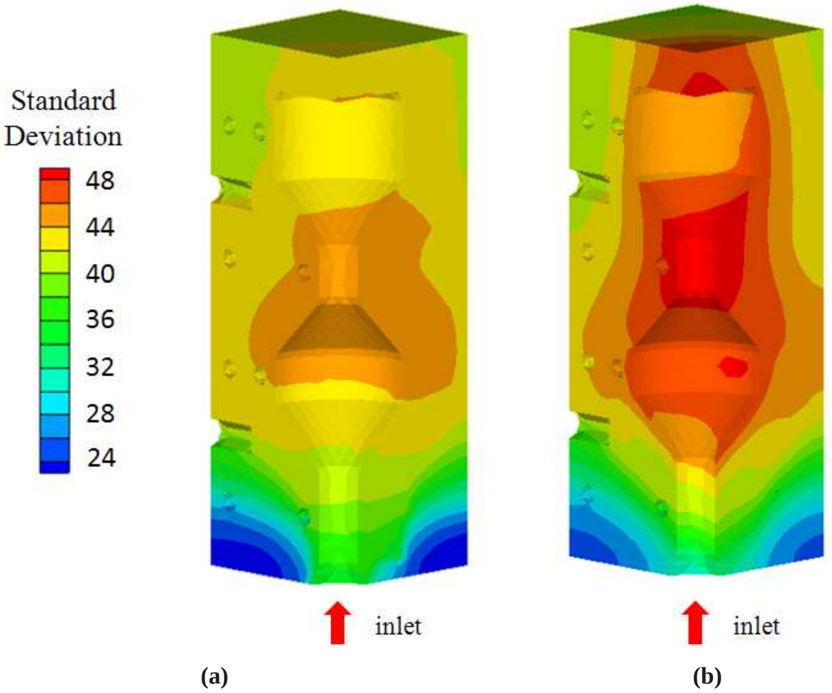


Figure 3.6: Contour Plots of the Standard Deviation for *CI* Based on Two Candidate Temperatures: (a) Maximum Temperature and (b) Ejection Temperature with the Total Volume of Encapsulated Liquid (refer to Figure 2.3).

If the focus on the region of encapsulated liquid in a casting is moved, the optimal location to monitor temperature will be changed accordingly. To support this point of view, the correlation analysis involving *CI* and *STD* could be performed on a subsection of the casting volume to determine what effect, if any, the region of interest has on the results. Figure 3.7 shows a contour plot of *CI* for the two die temperatures where the volume of encapsulated liquid was evaluated in the upper (enlarged cross-section) volume of the casting. In this case, the region of the highest *CI* values has moved upward, but remains below the narrow cross-section region in the middle of the die. The reason for this may lie in the fact that the temperature variation for the area below this narrow cross-section region has a strong effect on liquid encapsulation in the upper volume of the casting, hence the highest

CI (computed at steady state) occurs in this area. The *STD* of the correlations based on liquid encapsulation in the upper volume (Figure 3.8) indicate that the largest variation in the correlations over the operational range is expected in the narrow cross-section region. Thus the correlations in this region will show more sensitivity to changes in operational conditions making the correlation of die temperature to volume of encapsulated liquid inaccurate.

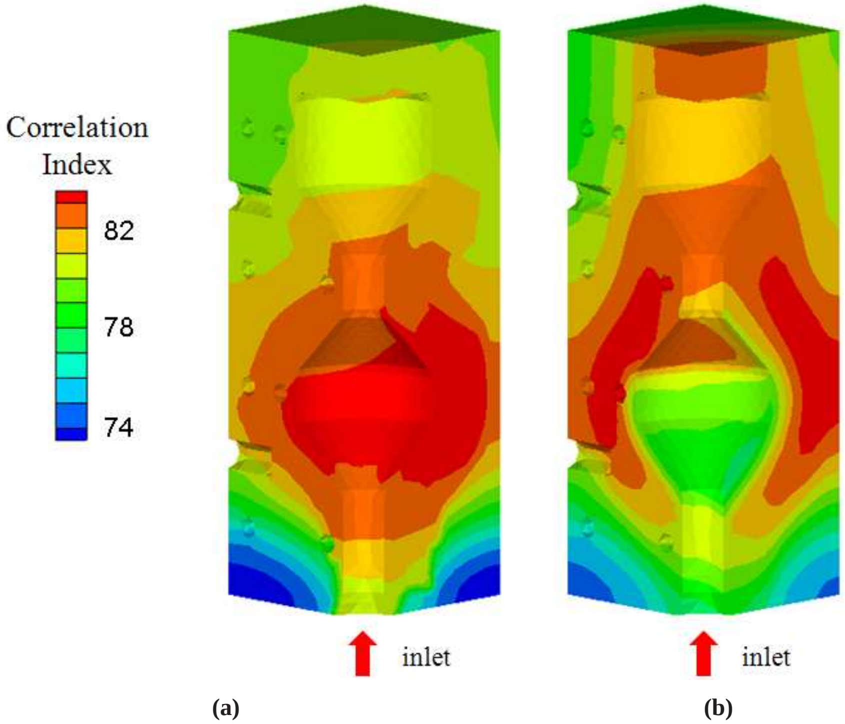


Figure 3.7: Contour Plots of *CI* Based on Two Candidates for Die Temperatures: (a) Maximum Temperature and (b) Ejection Temperature with the Upper Volume of Encapsulated Liquid (refer to Figure 2.3a).

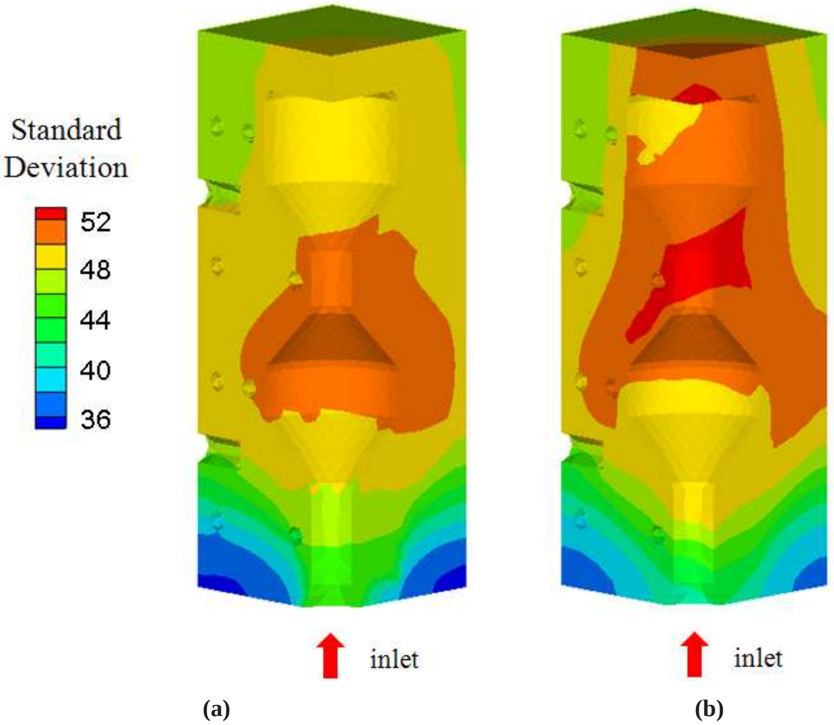


Figure 3.8: Contour Plots of Standard Deviation for *CI* Based on Two Candidates for Die Temperatures: (a) Maximum Temperature and b) Ejection Temperature with the Upper Volume of Encapsulated Liquid (refer to Figure 2.3a).

Table 3.2 summarizes the *CI* and standard deviation results for the two candidate die temperatures: Maximum Temperature and Ejection Temperature, based on the encapsulated liquid occurring in the entire casting volume. The Maximum Temperature shows higher overall *CI* values and smaller standard deviation of *CI*, and therefore is a better indicator of the volume of encapsulated liquid. This indicates that the Maximum Temperature is a better correlation to the volume of encapsulated liquid and exhibits the least amount of variability with changes in operational conditions. Based on the above analysis, the Maximum Temperature is proposed as an indicator for the volume of macro-porosity occurring in the casting.

In summary, a methodology has been developed to compute *CI* and *STD* to

Table 3.2: Summary of *CI* and *STD* for the Two Characteristic Die Temperatures Based on the Encapsulated Liquid Occurring in the Entire Casting.

Die Temperature	Correlation Index		Standard Deviation of Correlation Index	
	maximum	minimum	maximum	minimum
Maximum Temperature	84.25	77.00	44	24
Ejection Temperature	83.99	69.25	48	26

aid in determining the optimal die location for monitoring temperature as a proxy for encapsulated liquid volume. The optimal location was found to be the bottom of the die, near the metal inlet on the casting/die interface, the area which was marked with circle in Figure 3.5a. The next subsection presents an approach to quantitatively express the volume of encapsulated liquid based on die temperature.

3.4 Linear Regression (LR) Method

Given the Maximum Temperatures at selected die locations, the amount of encapsulated liquid will be quantified by applying a LR expression. The data necessary to perform this analysis can be collected from a process in a transient state, at steady state or the combination of both. A least squares LR model is proposed to express the linear relationship between Maximum Temperatures and the volume of liquid encapsulation. A LR model relates dependent variables y with independent variables x with the following expression,

$$y = a_0 + \sum_{i=1}^{N} a_i x_i \tag{3.3}$$

where a_i are regression coefficients associated with x_i, a_0 is a constant and N is the number of independent variables. In the current work, x_i are the Maximum Temperatures at die locations i, and y is the volume of liquid encapsulation. The specific value of a_0 and a_i can be computed using least squares method by comparing predicted and virtual process volumes of liquid encapsulation.

For example, if the Maximum Temperature at 11 locations, as marked in Figure 3.9, are selected to relate to the volume of liquid encapsulation under steady state, 11 sets of Maximum Temperature data at cyclic steady state for the 189 operational conditions and the corresponding volumes of encapsulated liquid are collected for analysis. The coefficients $a_0 - a_i$ in Equation (3.3) can be computed using the least squares method by comparing predicted and virtual process volumes of liquid encapsulation.

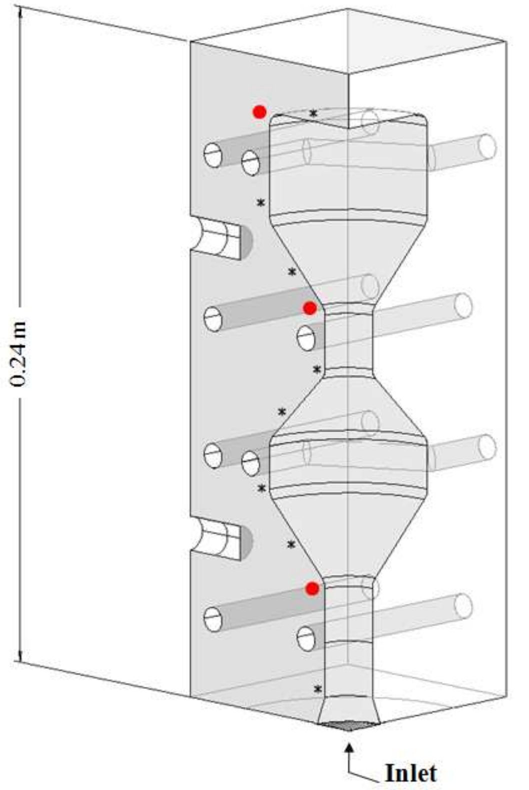


Figure 3.9: Schematic of a $\frac{1}{4}$ Section of Die with 11 Selected Locations Marked.

Figure 3.10 compares the response surfaces of the volume of liquid encapsulation calculated by the least squares expression and the virtual process over the operational range. The difference between the volume predicted by Equation (3.3) and the volume computed by virtual process appears to be small. The Fit, based on Equation (3.4), is estimated to be 92. A higher number indicates a better Fit between the virtual process and predicted response surfaces. A value of 100% signifies a perfect Fit.

$$Fit = 100 \times (1 - \frac{||y - \hat{y}||_2}{||y - \bar{y}||_2}) \qquad (3.4)$$

Where y is the virtual process data, $\hat{y}$ is the predicted data and $\bar{y}$ is the mean of y. $|| \, ||_2$ is the Euclidean Norm. This comparison indicates that the linear expression (Equation (3.3)) based on inputs of Maximum Temperatures from 11 die locations correctly quantifies the volume of liquid encapsulation.

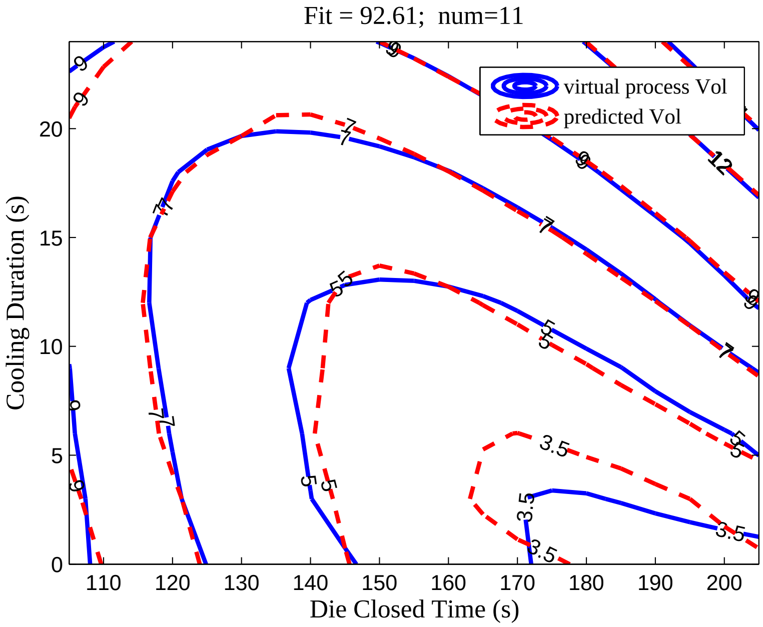


Figure 3.10: Contours of Predicted and Virtual Process Volumes of Encapsulated Liquid over the Operational Range.

3.5 Selection of Die Monitoring Locations

One difference between the two types of analysis is that the correlation methods are based on steady state conditions, while the LR method is more flexible and can apply to transient or steady state conditions to identify the volume of encapsulated liquid. Another difference is that the LR method offers a clear mathematical expression of the volume of encapsulated liquid using Maximum Temperatures. Thus, for control purposes, the LR method is better suited for use in control model to generate a model expression with the volume of encapsulated liquid as the model output which is driven by manipulated inputs. Establishing a control model of this type facilitates the implementation of model-based advanced control methodology with the goal of minimizing the volume of encapsulated liquid. However, the selection of die locations for control purposes is a continuing challenge in LPDC.

The method of correlation analysis explores the steady state relationship between in-cycle die temperatures and the amount of liquid encapsulation at a die location. The best location to assess the variation of the volume of encapsulated liquid by monitoring the Maximum Temperatures is at a die location with the highest *CI* and the smallest *STD*. However, it is not sufficient to measure temperature at a die location for the purpose of controlling the system to minimize liquid encapsulation due to a lack of an accurate mathematical expression for the volume of encapsulated liquid.

The following analysis is conducted on the virtual process to finalize the selection of die locations for the purpose of obtaining a ”best” fit mathematical expression for the volume of encapsulated liquid. A total of 11 die locations in Figure 3.9, which are evenly distributed along the die-casting interface, were used as candidates for the selection of die locations. The 11 locations are shown by 8 thermocouple locations marked with black stars and 3 additional locations marked with red dots.

Two selection rules for die temperature monitoring locations will be evaluated. This first is to choose locations with the highest *CI* and the second is to select locations that are evenly distributed along the casting-die interface. To assess which location selection philosophy is best, a series of simulations have been performed and the results are summarized in Table 3.3. The columns in Table 3.3 show the Fit

Table 3.3: LR Expression Fit Rate Comparison of the Two Candidate Rules for Die Locations Selection.

Selection rule	Number of Die Locations				
	1	2	3	5	11
Loc. with highest *CI*	41.79	42.74	42.77	87.12	92.61
Loc. evenly distributed	27.48	59.44	77.28	88.93	

of the LR expression for the volume of encapsulated liquid using different numbers of the die locations considered and the rows show the Fit of the LR expression for the volume of encapsulated liquid using the two candidate rules for die location selection. When only 1 of the 11 die locations (refer to locations marked in Figure 3.9) is selected, the location with highest *CI* has the higher Fit of 41.79. The evenly distributed location (middle of the die in this case) has the lower Fit, which implies that the location with the highest *CI* is best when there is only 1 location. When 2 or more locations are considered, die locations that are evenly distributed along the die-casting interface exhibit the best Fit.

Table 3.3 also shows that the fit of the LR expression increases as the number monitoring locations increases. This can be explained by the fact that more die locations are involved and these provide more information in predicting the volume of liquid encapsulation and hence the accuracy of the LR expression improves.

Regarding the selection of die locations, we come to the following conclusions:

- Increasing the number of locations were temperatures are monitored will improve the accuracy of the volume of encapsulated liquid calculated with the LR expression.

- If only one temperature monitoring location is possible, the location with the highest *CI* should be selected.

- For more than one monitoring location, choose die locations that are evenly distributed.

Both *CI* analysis and the LR expression methods were able to link die temperature to the volume of liquid encapsulation over an specified operational range. In the remaining chapters which consider the control of this LPDC process, evenly distributed locations along the casting/die interface will be employed to monitor the Maximum Temperature.

Chapter 4

System Identification[1]

4.1 Overview

Controlling a process using a model-based controller requires knowledge of the relationship between the process inputs and the process outputs. For a multi-input and multi-output process, a discrete state-space model structure is a good candidate for predicting the input-output behaviour of the process. The process of finding an approximate model that fits a set of measured input-output data is typically referred to as System Identification [92]. When the relationship between the multiple inputs and multiple outputs are linear, a method, known as ’Numerical Algorithms for Subspace State-Space System Identification’ (N4SID) [93], can be used to generate a state-space model derived from the measured input-output data of a process. However in the current research, a linear state-space model is not suitable since the relationship between system inputs and outputs is nonlinear over the operational range.

In the previous chapter, a 3D model was developed and validated with the

[1] Portions of this chapter have been published or prepared for publication in:
• X. Shi, D.M. Maijer and G. Dumont, “Assessing Controllability of Low Pressure Die Casting using Process Simulation”, Proceedings of the 7th Pacific Rim International Conference on Modeling of Casting and Solidification Processes 2007, Dalian, China, p635-642, August 2007.
• X. Shi, D.M. Maijer and G. Dumont, “Nonlinear Identification for Control of Low Pressure Die Casting” accepted, American Control Conference 2012, Montreal, Canada, June 27-29, 2012.
• X. Shi, Eranda Harinath, D.M. Maijer and G. Dumont, “Nonlinear Model Predictive Control in a Low Pressure Die Casting Process”, Journal of Process Control, submitted 2012.

goal of determining the boundary conditions necessary to describe the process. The tuning process used to determine the boundary conditions proceeded until the predicted data of the model were in agreement with measured data from a plant trial. For system identification and corresponding controller design, a 3D model was not deemed suitable due to its long simulation time, hence it has been replaced by a 2D-axisymmetric model, as shown in Figure 4.1, which incorporates a similar casting/die interface geometry and requires the same boundary conditions as those from the 3D validated model. This replacement was motivated by the belief that the core control strategy and its performance for the 2D model would not be greatly different from those of the 3D model since the 2D-axisymmetric model partially reproduces the geometry of the 3D model and has the same boundary conditions as the 3D model.

This chapter describes the following 8 steps used to perform nonlinear system identification.

1. Determine the process variables.
2. Determine the control variables and operational range.
3. Determine the feedforward variables.
4. Specify the nonlinear state-space model structure.
5. Perform nonlinear system identification for control variables'(u_v) model.
6. Perform linear system identification for feedforward variables'(u_f) model.
7. Generate a complete augmented State-Space model.
8. Validate the complete nonlinear state-space model.
9. Use LR expression to calculate the area of liquid encapsulation.

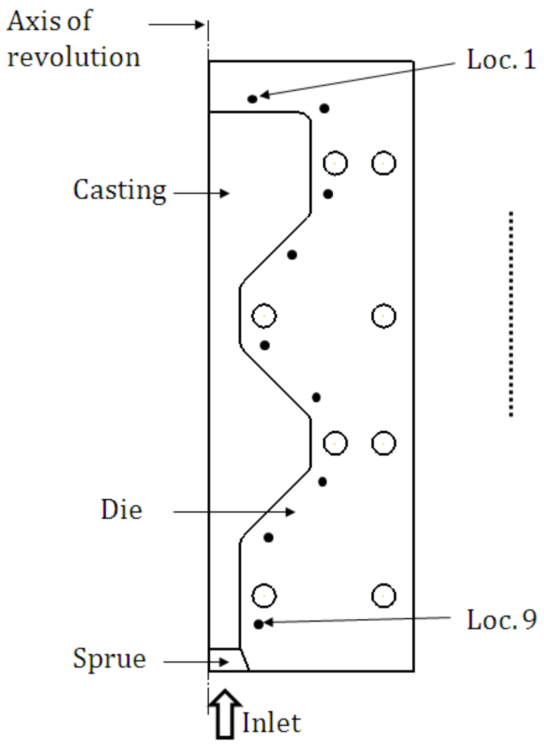


Figure 4.1: 2D-axisymmetric Model With 9 Selected Locations Marked With Dot.

4.2 Process Variables

The objective for controlling the current LPDC process is the minimization of macro-porosity in the casting and thus improve the process performance. The ability to identify and quantify the presence of macro-porosity in a casting is therefore important for any advanced control methodology seeking to control its formation.

To improve the process performance, the controller must drive the process variables of the system toward predetermined trajectories. For casting processes in general, it is not easy to set the process variables and their predetermined trajectories. Fortunately, the current work has led to the development of a LR expression linking die temperatures with the volume of liquid encapsulation which allows the desired trajectories for die temperatures to be indirectly determined while minimizing the volume of liquid encapsulation.

Two characteristic temperatures have been assessed at each die location for each casting cycle: Maximum Temperature and Ejection Temperature (refer to Figure 1.2b). By maintaining the Maximum Temperature at a die location at its optimal value, the emphasis is on the time in the casting cycle when solidification is occurring. Maintaining Ejection Temperatures at their optimal values emphasizes the portion of the casting cycle where solidification is nearly complete. The ”Correlation Index”, presented in Chapter 3.2, was used to analyze the correlation between die temperature and encapsulated liquid volume and Maximum Temperature was shown to be the optimal process variable output based on its high correlation with the volume of liquid encapsulation. Therefore identification of the state-space model will be performed using Maximum Temperature process variables.

As described in the previous chapter, die locations evenly distributed along the die-casting interface were selected to measure Maximum Temperatures for control in LPDC. These die locations are selected to surround the critical areas of the casting where choking occurs, motivated by the belief that if the optimal die temperature curves in these critical areas can be reproduced each cycle, then the amount of the liquid encapsulation can be reduced or eliminated. Nine locations in the die, evenly distributed along the die-casting interface, shown in Figure 4.1 (marked with dots), were selected. Note that these locations were selected as close as possible to the casting / die interface to enhance the accuracy of the correlation between die temperatures and the encapsulated liquid area.

4.3 Control Variables and Operational Range

The control variables must be selected to enable the controller to maintain the process variables at their optimum values. The optimum values in this case should result in minimal macro-porosity and therefore the control variables that affect the macro-porosity should be selected. In the case of a 2D-axisymmetric model, since the area of liquid encapsulation is the indirect indicator for the extent of macro-porosity and the formation of the area of liquid encapsulation is closely linked with the die temperatures, those variables which affect the die temperatures will initially be considered as the potential candidates for control variables.

The die casting process is essentially comprised of a heat source (the casting) surrounded by a heat sink (the die). The die facilitates the transport of heat from the casting to the surrounding environment. The majority of the heat from the die is transferred directly to the surrounding environment through the cooling channels when water is used as the coolant. Changes to this active cooling can have large effects on die temperature. Heat is also lost passively from the die to the surrounding environment via convection and radiation from the exterior die surfaces throughout the casting cycle and from the interior die surfaces when the die is open after the casting is ejected. By increasing or decreasing the total cycle time, the amount of heat transferred from the dies to the surrounding environment will also increase or decrease respectively. Thus, the total cycle time and duration of cooling in the cooling channels are the potential control variables.

By increasing the current cycle's duration, die temperatures during the next cycle will decrease. Likewise, by decreasing the current cycle's cooling duration, the next cycle temperatures will increase. Figure 2.2 shows the two main cycle timing parameters: Die Closed Time and Die Open Time. During the die closed time, the die halves close, the molten aluminum enters the die cavity, and the casting solidifies. When the die opens, the casting is ejected and the operator prepares for the next cycle. When preparations are complete, the operator presses a button to signal that the next cycle can proceed. The process then waits until a timer expires to start the next cycle. To modify the total cycle time, one or both of the cycle timing parameters can be modified. However, because the operator requires time to prepare the dies and has final control of the Die Open Time, trying to modify the Die Open Time with a controller is not desirable. Therefore the Die Closed Time parameter should be used as the control variable to vary the overall cycle time. Although the Die Closed Time could be increased indefinitely as required by the controller, it can not be decreased indefinitely. If the Die Closed Time is reduced too much, the casting will not completely solidify and production would stop. This introduces a lower constraint on the control variable.

The amount of heat removed by the forced water in the cooling channels can be modified in four ways. The first option is to vary the temperature of the water circulating in the channel. This option is not feasible in the process because the temperature of the incoming water is not controlled. The second option is to vary the flow rate of the water entering the cooling channel. Unfortunately this option is also not feasible because the flow rate of the circulating water is uncontrolled. The third option is to change the time in each cycle when water begins to circulate in the cooling channel. The fourth option is to vary the duration of time that water is flowing through the cooling channel. In the current process, it is possible to adjust the start time and cooling duration in each of the four cooling channels.

Nine parameters (start time, cooling duration for the 4 cooling channels, and die closed time) were evaluated to assess the effects on liquid encapsulation. A two-level factorial analysis was performed with the virtual process to determine the parameters that influence liquid encapsulation [88]. An orthogonal array was used to systematically vary and test the different levels for each of the control factors. This analysis indicated that Die Closed Time and Cooling Duration for cooling channel 1 had a non-trivial effect on the magnitude of liquid encapsulation meaning that that the area could be increased or decreased depending on the levels of these factors. It was found that the minimum are of liquid encapsulation occurred when the start time for cooling in Channel 1 was zero. The results also showed that turning on the cooling Channels 2 – 4 led to increased area of liquid encapsulation. Thus, changing the start time of cooling Channel 1 or activating cooling in Channels 2 – 4 is counter to the goal of minimizing liquid encapsulation, so they have been eliminated as possible control variables. The control variables identified through this analysis and their designations are:

- Die Closed Time (u_{v_1})
- Cooling Duration in Cooling Channel 1 (Cooling Duration or u_{v_2})

While assessing the control variables, the approximate operational region was examined. The response of the process was evaluated in increments of 10 s between Die Closed Time from 120 to 160 s and Cooling Duration varying from 0 to 40 s in increments of 10 s, making up 25 combinations of operating points. The virtual process was run for each of these conditions until the cyclic steady state was reached. The contour plot of the area of liquid encapsulation over the operational range is presented in Figure 4.2. The optimal operational condition at cyclic steady state has been identified as Die Closed Time equals to 150 s and Cooling Duration equals to 10 s. The other non-control related process variables include Die Open Time and incoming Metal Temperature where are equal to 50 s and 700 °C, respectively.

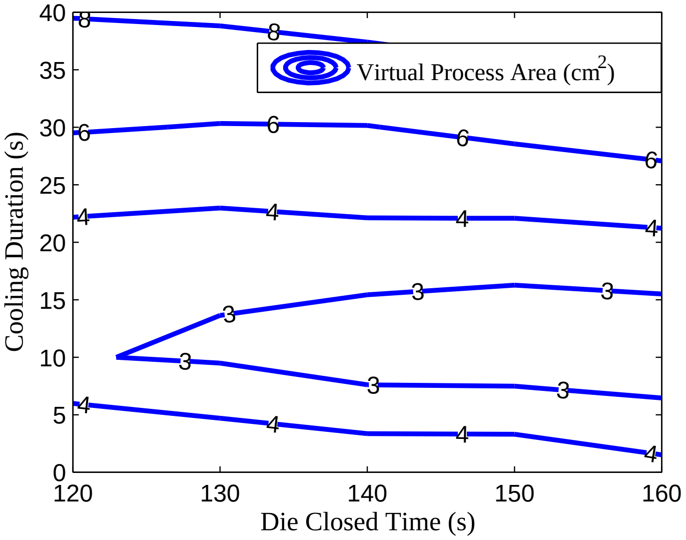


Figure 4.2: Contour Plot of the Area of Liquid Encapsulation Across the Operational Range.

4.4 Feedforward Variables

Feedforward variables are any process inputs that can be measured but not manipulated by the controller. Process upsets caused by disturbances are typically dealt with by using feedback to correct the negative effects of the upset. By measuring the disturbances and using them as feedforward variables in the MPC controller, the effects of the process upset can be predicted and dealt with before the disturbance propagates through the system. The following feedforward variables have been identified for LPDC casting:

- Incoming Metal Temperature (u_{f_1})

- Die Open Time (u_{f_2})

where u_{f_1} is the temperature of the molten aluminum when it enters the die at the start of the cycle and u_{f_2} is the length of time that the dies remain open after ejecting a casting. As discussed in Chapter 1, the incoming Metal Temperature can fluctuate in the industrial process due to variation in the holding furnace temperature and metal transfers. The time the die is open is constant from cycle to cycle during routine operation of the industrial process. However, this can vary when an operator has to perform extra maintenance on the die or experiences problem caused by the previous casting cycle. These upsets cause the die to remain open for a longer than usual time. The virtual process has been formulated with input tags corresponding to these feedforward variables to simulate the effects of deviations from the nominal values.

4.5 Nonlinear State-Space Model Structure

Figure 4.3 shows contour plots of the maximum die temperature at steady state for the operational range at four randomly selected die locations along the die-casting interface. Note that the system is nonlinear over the operational range as the contour lines are not all straight, and adjacent straight lines linking with the same temperature span do not have the same distance. Therefore, the state-space model should incorporate a nonlinear system structure.

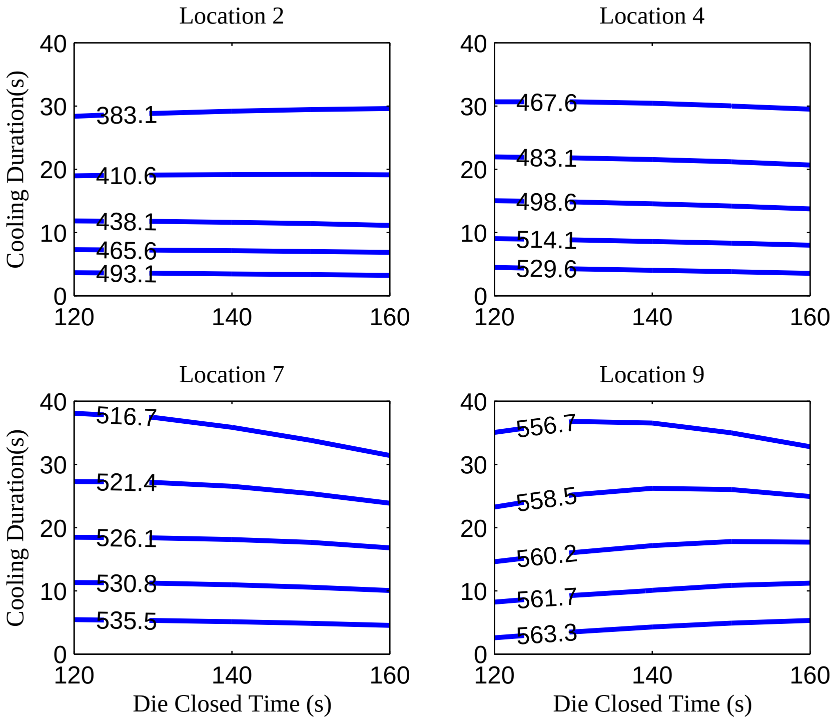


Figure 4.3: Contour Plots of the System Outputs at Cyclic Steady State over the Operational Range at 4 Die Locations.

Since the system has two different types of inputs: control variables and feedforward variables, a set of simple simulations were run to explore the superposition and homogeneity. The virtual process was run for a baseline condition (Die Closed Time, Cooling Duration, incoming Metal Temperature, and Die Open Time equal to 150 s, 10 s, 700 °C, and 50 s, respectively) until cyclic steady state was reached. The function $g(a_1,a_2,a_3,a_4)$ with the relative operating point set of $(\Delta u_{v_1} = a_1, \Delta u_{v_2} = a_2, \Delta u_{f_1} = a_3, \Delta u_{f_2} = a_4)$ upon the baseline represents outputs' steady state deviation from the baseline outputs. Note that $g(0,0,0,0) = 0$.

The virtual process was used to perform a series of simulations based on the combinations of 4 step sequences for 4 variables. The results yield the following expressions:

$$\begin{aligned}
&g(0,0,a_3,0)+g(0,0,0,a_4)\approx g(0,0,a_3,a_4)\\
&g(a_1,0,0,0)+g(0,a_2,0,0)\neq g(a_1,a_2,0,0)\\
&g(a_1,a_2,a_3,a_4)\approx g(a_1,a_2,0,0)+g(0,0,a_3,a_4)
\end{aligned}$$

Note that the control variable-driven system (u_v model) is nonlinear and does not meet superposition criteria whereas the feedforward variables-driven system (u_f model) is the opposite. Superposition exists between these two systems. Therefore, the entire system output y can be regarded as the sum of a control variable model output y_v and a feedforward variable model output y_f. The control variable system is nonlinear with linear dynamic behaviour and a nonlinear steady state gain, while the feedforward variable system can be treated as a linear system. The sampling time for all these discrete-time models is set as a cycle.

The properties and structure of two separated systems will now be examined before developing a complete augmented model.

4.6 Nonlinear System Identification for the u_v Model

u_v model structure is presented in Equation (4.1), where A_v is a diagonal matrix and $B_v = I - A_v$. This subspace model structure is designed to normalize x_v to u_v and then to decompose system eigenvalues in the purpose of facilitating the separation of linear dynamics and nonlinear static gain in the model.

$$\begin{aligned}
x_v(k+1) &= A_v x_v(k)+(I-A_v)u_v(k)\\
y_{v_i}(k) &= f_i(x_{u_{v1i}}(k),x_{u_{v2i}}(k))
\end{aligned}\tag{4.1}$$

where i is the die location, $i \in [1-9]$,k is the current cycle number; $u_v(k)$, $x_v(k)$, and $y_v(k)$ are the control variables, the state vectors, and the system outputs at time step k, respectively; $x_{u_{v1i}}(k)$, $x_{u_{v2i}}(k)$ are linear expressions of $x_v(k)$ with unit steady state gain for u_{v_1} and u_{v_2} respectively, representing the dynamic response from $u_{v_1} \rightarrow y_{v_{1i}}$ and $u_{v_2} \rightarrow y_{v_{2i}}$, respectively. The function f expresses the system's nonlinear static gain.

Nonlinear Static Gain Surface Identification

The function $f_i(a_1,a_2)$ with two independent variables a_1 and a_2 in Equation (4.1) denotes the steady state values for y_{v_i} when $a_1 = u_{v_1}$ and $a_2 = u_{v_2}$. The fitting function $\mathbf{Z}(a_1,a_2)$ for $f_i(a_1,a_2)$ can be generally expressed as,

$$\mathbf{Z}(a_1,a_2) = \sum_{i=0}^{p-1}\sum_{j=0}^{q-1} \theta_{ij} a_1^i a_2^j \tag{4.2}$$

where the coefficient θ makes up the matrix Θ with $p \times q$ dimension. Given p and q, θ can be computed using the least-squares method by comparing predicted and virtual process data. The method is detailed as follows:

First, the element $(\theta_{00} \ldots\ldots \theta_{p-1q-1})$ of the matrix Θ is the local minimum point of the function $\mathbf{S}$

$$\mathbf{S}(\theta_{00} \ldots\ldots \theta_{p-1q-1}) = \sum_{g=1}^{n} w(x_i,y_j)[\sum_{i=0}^{p-1}\sum_{j=0}^{q-1} \theta_{ij} x^i y^j - f(x_i,y_j)]^2$$

where, $w(x,y)$ is the weighting function which is set to 1 by default. Hence, $\theta_{00} \ldots\ldots \theta_{p-1q-1}$ must satisfy,

$$\frac{\partial \mathbf{S}}{\partial \theta_{ij}} = 0 \qquad (ij = 00,\ldots,p-1q-1)$$

Taking the partial derivative of $\mathbf{S}$ and moving this term to the right side of the equation gives,

$$\begin{bmatrix} (\varphi_{00},\varphi_{00}) & \cdots & (\varphi_{00},\varphi_{p-1q-1}) \\ \vdots & \ddots & \vdots \\ (\varphi_{p-1q-1},\varphi_{00}) & \cdots & (\varphi_{p-1q-1},\varphi_{p-1q-1}) \end{bmatrix} \times \begin{bmatrix} \theta_{00} \\ \vdots \\ \theta_{p-1q-1} \end{bmatrix} = \begin{bmatrix} (\varphi_{00},f) \\ \vdots \\ (\varphi_{p-1q-1},f) \end{bmatrix} \tag{4.3}$$

where,

$\varphi_{00} = x^0 y^0, \varphi_{p-1q-1} = x^{p-1} y^{q-1}$.

Equation (4.3) can be simplified to,

$$A \times \Theta_n = B \tag{4.4}$$

where,

$$\begin{aligned}
A &= \begin{bmatrix} \sum x_g^0 y_g^0 \cdot x_g^0 y_g^0 & \dots & \sum x_g^0 y_g^0 \cdot x_g^{p-1} y_g^{q-1} \\ \vdots & \ddots & \vdots \\ \sum x_g^{p-1} y_g^{q-1} \cdot x_g^0 y_g^0 & \dots & \sum x_g^{p-1} y_g^{q-1} \cdot x_g^{p-1} y_g^{q-1} \end{bmatrix} \\
\Theta_n &= \begin{bmatrix} \theta_{00} \\ \vdots \\ \theta_{p-1q-1} \end{bmatrix} \\
B &= \begin{bmatrix} \sum x_g^0 y_g^0 \cdot z_g \\ \vdots \\ \sum x_g^{p-1} y_g^{q-1} \cdot z_g \end{bmatrix}
\end{aligned}$$

The least squares solution of Equation (4.4) provides the elements θ_{ij} of Θ_n Matrix. The final p and q are selected based on the minimization of error according to an appropriate criteria. In the current work, 2 criteria have been identified,

$$\begin{aligned}
error1 &= \sum_{i=1}^{N} |z_i - f(x_i, y_i)| \\
error2 &= \sum_{i=1}^{N} (z_i - f(x_i, y_i))^2
\end{aligned}$$

Based on this procedure, p and q equal to 3 and 4, respectively, minimizes *error*1 and *error*2 for all 9 die locations. The corresponding error values for the different die locations are shown in Table 4.1.

Table 4.1: Errors of the Fitting Function **Z** (Equation (4.2)) for Different Die Locations (unit:°C).

Loc.#	1	2	3	4	5	6	7	8	9
Error1	0.212	0.320	0.366	0.284	0.269	0.293	0.296	0.205	0.161
Error2	0.005	0.012	0.013	0.007	0.005	0.007	0.008	0.004	0.002

The fitting function coefficient, $\Theta_1 - \Theta_9$ (refer to Equation (4.2)), were then computed for all 9 die locations to define the static gain of the local outputs (refer to Equation (4.1)). Figure 4.4 compares the predicted data from the fitting function **Z** to the virtual process output data for a randomly selected die location over the operational range. The 2D contours comparison shows that the two contours surfaces are nearly overlaid, indicating that the function f in the model (Equation (4.1)) is captured correctly by the fitting function **Z**.

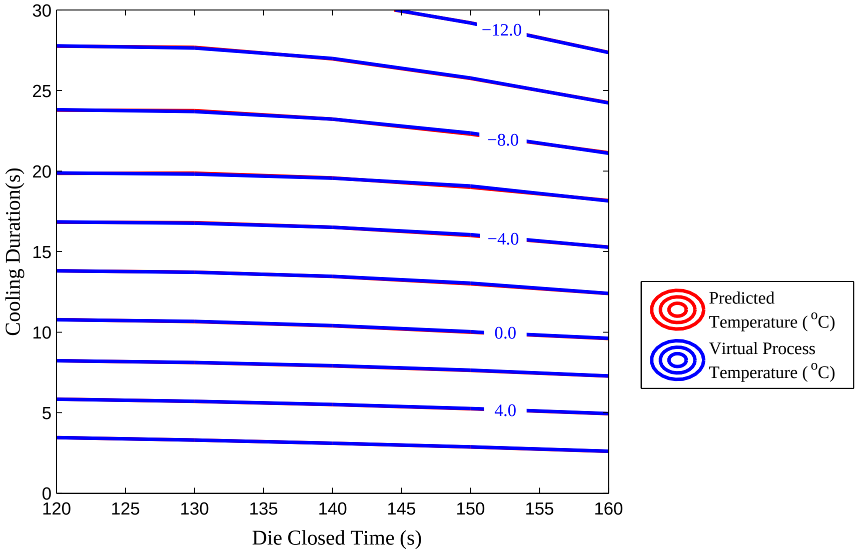
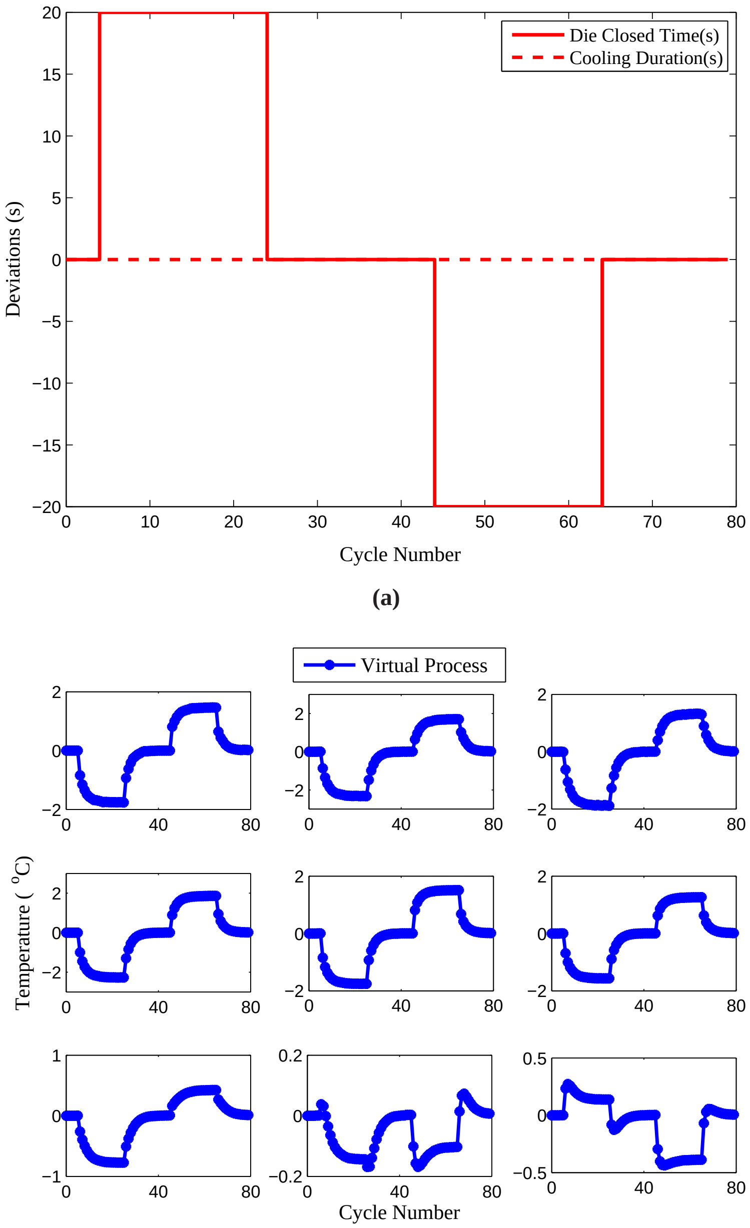


Figure 4.4: 2D Contours of Maximum Temperature for both Static Gain Predicted by Equation (4.2) and Virtual Process Steady State Data over the Operational Range at a Random Die Location.

Linear Dynamic Behaviour Identification

In order to determine the dynamic behaviour, input step sequences were independently executed for the two control variables on the virtual process. The dynamic response of the u_v system was evaluated at the 9 die locations. Figure 4.5 presents the system response for the u_{v_1} step sequence. These results show that the system contains no complex-conjugate poles since the curves are not oscillating. The same is true for the system shown in Figure 4.6. Therefore, a single-input multi-output (SIMO) model structure with real poles will adequately describe the dynamic behaviour from u_{v_1} to y_{v_i} and from u_{v_2} to y_{v_i}. Considering Figure 4.5 and Figure 4.6, it is apparent that the responses with significant gains have time constants of approximately 5 cycles.

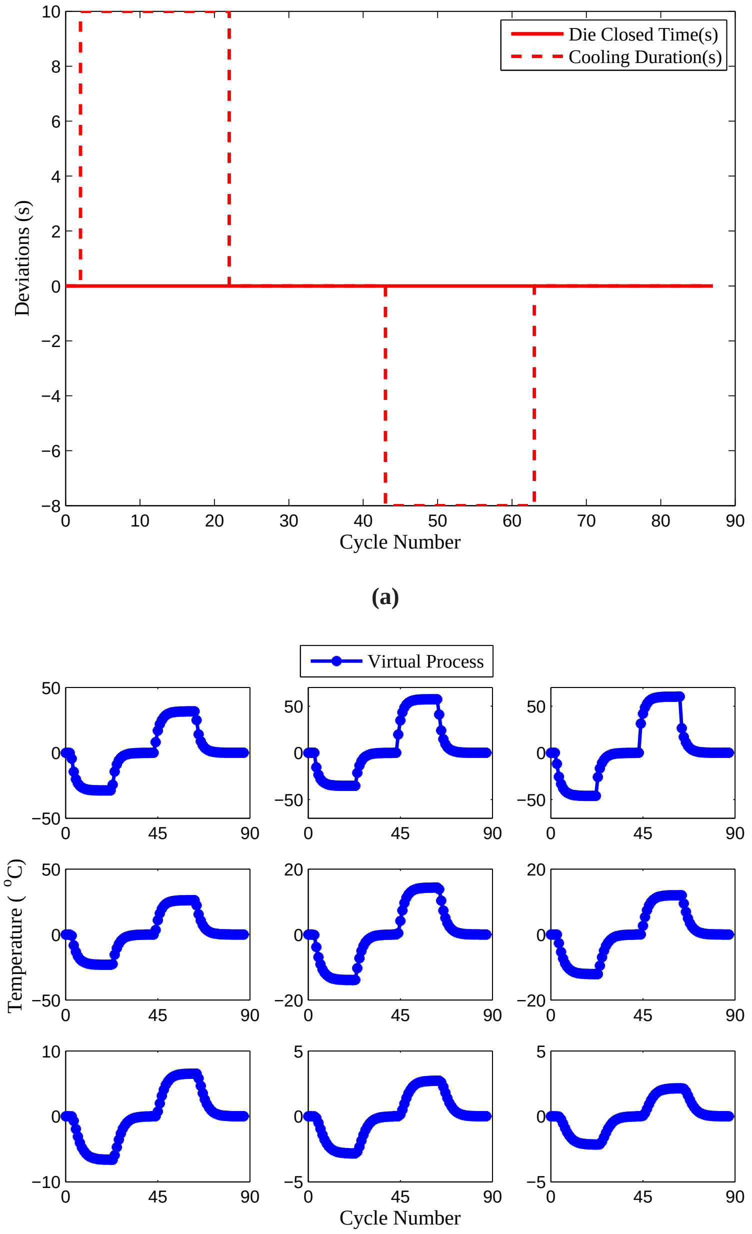
The approach to generate the first part of Equation (4.1) to describe the linear dynamic behaviour in the model and to obtain $x_{u_{v1i}}(k)$ and $x_{u_{v2i}}(k)$ for use in the



(a)

(b)

Figure 4.5: (a) The Step Sequence of Control Variable (u_{v_1}) for Identification and (b) the System Response to the Step Sequence.



(a)

(b)

Figure 4.6: (a) The Step Sequence of Control Variable (u_{v_2}) for Identification and (b) the System Response to the Step Sequence.

second part of the nonlinear static gain function is as follows:

- Extract y_{v_i} from the virtual process data for input variations to u_{v_1}. Normalize the steady state gain of y_{v_i} for each input condition to 1 and remove one cycle time delay (u_{v_1} has a one cycle time delay on all y_{v_i} due to the implementation).

- Perform a linear system identification on the processed data to generate a 2^{nd} order state-space model with the matrices $(A,B,C,0)$. A 2^{nd} order model is used because it is the minimum acceptable order that accurately identified the virtual process.

- On the basis of not affecting model dynamics and outputs, convert the matrices $(A,B,C,0)$ into $(A_v,B_v,C_v,0)$ to get $A_v = eig(A)$ and $B_v = I - A_v$. C_v can be computed as follows:

  By similar matrix T transformation [21], $(A,B,C,0)$ is converted into diagonal matrices $(A_d,B_d,C_d,0)$. Where A_d is the diagonal matrix, $A_d = T^{-1}AT$, $B_d = T^{-1}B$=$[B_{1d};B_{2d}]$, and $C_d = CT = [C_{1d} \quad C_{2d}]$. The output Y from the system with the matrices $(A,B,C,0)$ can then be expressed as:

$$\begin{aligned} Y &= [C_{1d} \times \frac{B_{1d}}{Z - A_{1d}} + C_{2d} \times \frac{B_{2d}}{Z - A_{2d}}] \times U \\ &= [\frac{C_{1d}\ B_{1d}}{1 - A_{1d}} \times \frac{1 - A_{1d}}{Z - A_{1d}} + \frac{C_{2d}\ B_{2d}}{1 - A_{2d}} \times \frac{1 - A_{2d}}{Z - A_{2d}}] \times U \\ &= [\alpha \times \frac{1 - A_{1d}}{Z - A_{1d}} + \beta \times \frac{1 - A_{2d}}{Z - A_{2d}}] \times U \end{aligned}$$

  where,
  $A_{1d}, A_{2d}, B_{1d}, B_{2d}$ are scalars; C_{1d}, C_{2d} are column vectors.
  $\alpha = \frac{C_{1d}\ B_{1d}}{1-A_{1d}}, \beta = \frac{C_{2d}\ B_{2d}}{1-A_{2d}}$; $\beta = 1 - \alpha$ for $\alpha + \beta = 1$ under steady state. If $A_v = A_d$, $B_v = I - A_d$, then $C_v = [\alpha \quad 1 - \alpha]$. Note that the system $(A_v,B_v,C_v,0)$ has been successfully decoupled from the system state vectors.

- The new expression for the u_{v_1} model is:

$$\begin{aligned}
\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} &= \begin{bmatrix} a_1 & 0 \\ 0 & a_2 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 1-a_1 \\ 1-a_2 \end{bmatrix} u_{v_1}(k-1) \\
y_{v_{1i}}(k) &= \alpha_i \, x_1(k) + \beta_i \, x_2(k)
\end{aligned}$$

where the steady state gain $(u_{v_1} \rightarrow y_{v_{1i}}) = 1$.

- The augmented x expression with the combination of one cycle time delay is:

$$\begin{aligned}
\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{bmatrix} &= \begin{bmatrix} 0 & 0 & 0 \\ 1-a_1 & a_1 & 0 \\ 1-a_2 & 0 & a_2 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u_{v_1}(k) \\
y_{v_{1i}}(k) &= \alpha_i \, x_2(k) + \beta_i \, x_3(k) \qquad (4.5)
\end{aligned}$$

- Applying the same approach for the u_{v_2} model, an additional state is necessary to account for the time delay (either 0 or 1) which is not constant for $y_{v_{2i}}$. As a result, a 3^{rd} order state-space SIMO system for the u_{v_2} model has been defined as:

$$\begin{aligned}
\begin{bmatrix} x_4(k+1) \\ x_5(k+1) \\ x_6(k+1) \end{bmatrix} &= \begin{bmatrix} a_3 & & \\ & a_4 & \\ & & a_5 \end{bmatrix} \begin{bmatrix} x_4(k) \\ x_5(k) \\ x_6(k) \end{bmatrix} + \begin{bmatrix} 1-a_3 \\ 1-a_4 \\ 1-a_5 \end{bmatrix} u_{v_2}(k) \\
y_{v_{2i}}(k) &= \alpha_{anoi} \, x_4(k) + \beta_{anoi} \, x_5(k) + \gamma_{anoi} \, x_6(k) \qquad (4.6)
\end{aligned}$$

where,
$\alpha_{ano} = \frac{C_{1d} \, B_{1d}}{1-A_{1d}}$, $\beta_{ano} = \frac{C_{2d} \, B_{2d}}{1-A_{2d}}$, $\gamma_{ano} = \frac{C_{3d} \, B_{3d}}{1-A_{3d}}$, $\alpha_{ano} + \beta_{ano} + \gamma_{ano} = 1$ and steady state gain $(u_{v_2} \rightarrow y_{v_{2i}}) = 1$.

Complete Nonlinear State-Space u_v Model

By combining Equation (4.5) and Equation (4.6), the complete u_v model is expressed as:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \\ x_4(k+1) \\ x_5(k+1) \\ x_6(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & & & \\ 1-a_1 & a_1 & 0 & & & \\ 1-a_2 & 0 & a_2 & & & \\ & & & a_3 & & \\ & & & & a_4 & \\ & & & & & a_5 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \\ x_5(k) \\ x_6(k) \end{bmatrix}$$

$$+ \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1-a_3 \\ 0 & 1-a_4 \\ 0 & 1-a_5 \end{bmatrix} \begin{bmatrix} u_{v_1}(k) \\ u_{v_2}(k) \end{bmatrix}$$

$$\begin{aligned} y_{v_i}(k) &= f_i(y_{v_{1i}}(k), y_{v_{2i}}(k)) \\ &= f_i(x_{u_{v1i}}(k), x_{u_{v2i}}(k)) \end{aligned} \tag{4.7}$$

Note that a_1 and a_2 are the common eigenvalues for the u_{v_1} model and $x_{u_{v1i}}(k)$ is the dynamic response from $u_{v_1} \rightarrow y_{v_{1i}}$, with its steady state gain equal to the value of u_{v_1}; similar rationale and explanation apply for the u_{v_2} model.

Validation of the u_v Model

The virtual process data has been compared to predictions made with the u_v model for u_{v_1} and u_{v_2} step sequences in Figure 4.7 and Figure 4.8, respectively. Overall, the responses predicted by the u_v model match the dynamic behaviour of the virtual process correctly. The u_v model appears to inaccurately describe the response to Control Variable 1 at two die locations. However, a close inspection of the graphs reveals that the gain at these locations is small and the error is relatively small.

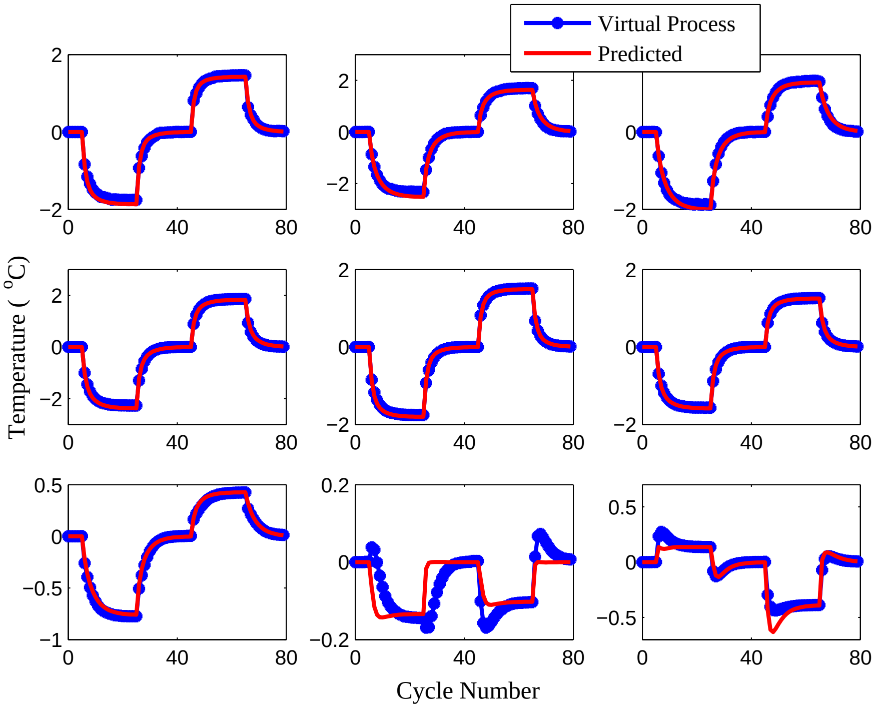


Figure 4.7: Predicted and Virtual Process Temperature Responses under the Step Sequence of Control Variable (u_{v_1}) for Identification.

4.7 Linear System Identification for the u_f System

For the linear subspace system identification, the N4SID algorithm was applied to process the data generated with the virtual process. All of the data had to be pre-processed to remove the mean value of the data so that the data was centred on zero. No pre-processing was needed on the control and feedforward variables because by definition these variables are deviations from zero. The process variables were pre-processed by subtracting the steady state temperatures from the data. The steady state temperatures were defined as the temperatures from the first cycle of identification before any changes to the inputs were made.

To generate the input-output data for system identification (SI), u_{f_1} and u_{f_2} step sequences (refer to Figure 4.9a) were sequentially applied to the virtual process. The data was then pre-processed to remove the time delay caused by u_{f_2}. Using

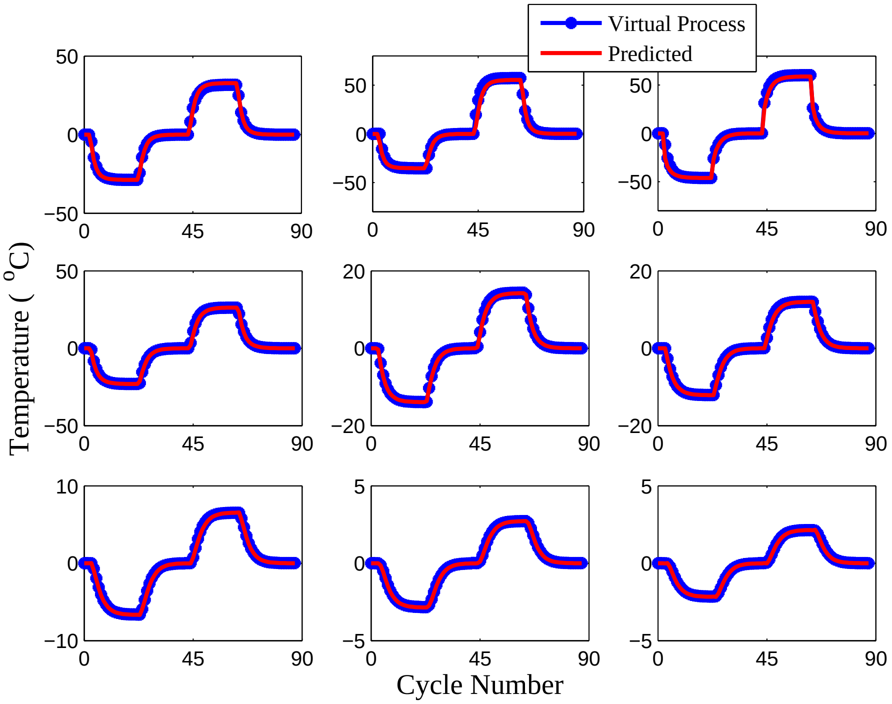


Figure 4.8: Predicted and Virtual Process Temperature Responses under the Step Sequence of Control Variable (u_{v_2}) for Identification.

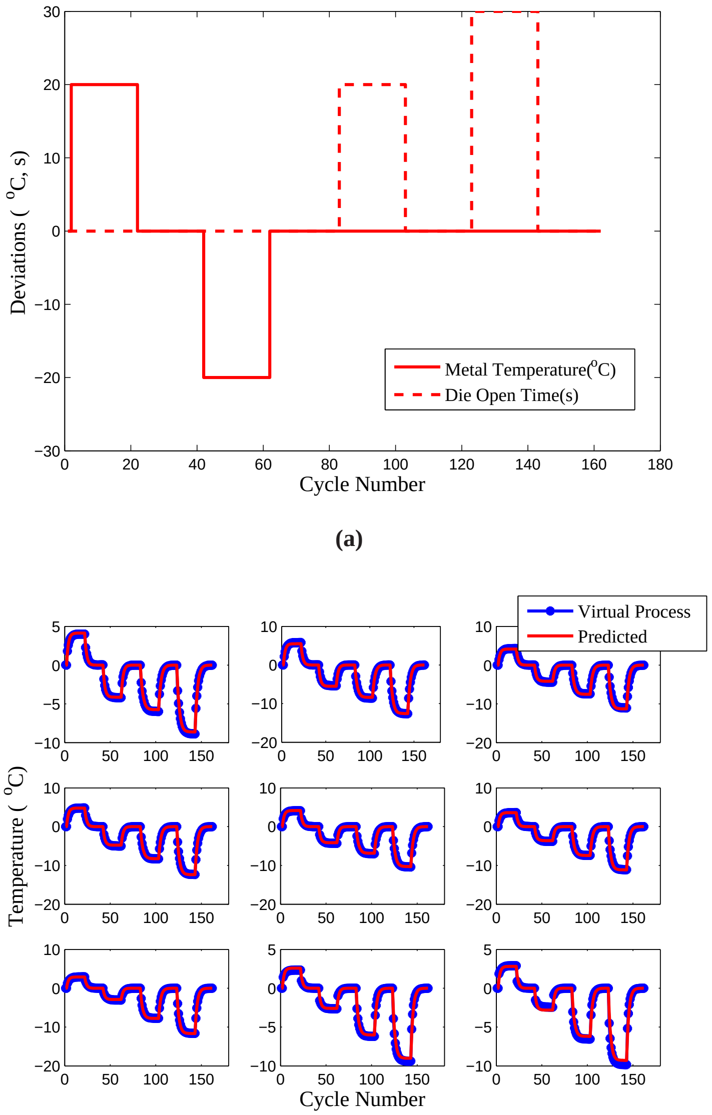
the N4SID algorithm [93], a 2nd order linear subspace u_f model was defined as:

$$\begin{aligned}\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} &= \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \begin{bmatrix} u_{f_1}(k) \\ u_{f_2}(k-1) \end{bmatrix} \\ y_f(k) &= C_f \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}\end{aligned}$$

An alternative expression with augmented x accounting for one time delay is:

$$\begin{aligned}\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{bmatrix} &= \begin{bmatrix} a_{11} & a_{12} & b_{12} \\ a_{21} & a_{22} & b_{22} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \begin{bmatrix} b_{11} & 0 \\ b_{21} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_{f_1}(k) \\ u_{f_2}(k) \end{bmatrix} \\ y_f(k) &= [C_f \; 0] \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} \end{aligned} \tag{4.8}$$

In response to the u_{f_1} and u_{f_2} step sequences applied to the virtual process, the virtual process data (blue symbols) and predicted (solid red curve) outputs at 9 die locations were obtained and compared in Figure 4.9b). The agreement between the virtual process and predicted data confirms the excellent accuracy of the linear u_f model in describing the u_f-driven virtual process.



(a)

(b)

Figure 4.9: (a) The Step Sequence of Feedforward Variables (u_f) for Identification and (b) the Corresponding System Predicted and Virtual Process Responses.

4.8 Complete Augmented State-Space Model

As described previously, the complete system output (y) can be obtained as the sum of u_f system output (y_f) and u_v system output (y_v). By combining Equation (4.7) and Equation (4.8), the complete model is expressed as:

$$
\begin{aligned}
\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \\ x_4(k+1) \\ x_5(k+1) \\ x_6(k+1) \\ x_7(k+1) \\ x_8(k+1) \\ x_9(k+1) \end{bmatrix}
&= \begin{bmatrix}
0 & 0 & 0 & & & & & & \\
1-a_1 & a_1 & 0 & & & & & & \\
1-a_2 & 0 & a_2 & & & & & & \\
 & & & a_3 & & & & & \\
 & & & & a_4 & & & & \\
 & & & & & a_5 & & & \\
 & & & & & & a_{11} & a_{12} & b_{12} \\
 & & & & & & a_{21} & a_{22} & b_{22} \\
 & & & & & & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \\ x_5(k) \\ x_6(k) \\ x_7(k) \\ x_8(k) \\ x_9(k) \end{bmatrix} \\
&+ \begin{bmatrix}
1 & 0 & & \\
0 & 0 & & \\
0 & 0 & & \\
0 & 1-a_3 & & \\
0 & 1-a_4 & & \\
0 & 1-a_5 & & \\
 & & b_{11} & 0 \\
 & & b_{21} & 0 \\
 & & 0 & 1
\end{bmatrix}
\begin{bmatrix} u_{v_1}(k) \\ u_{v_2}(k) \\ u_{f_1}(k) \\ u_{f_2}(k) \end{bmatrix} \\
y_i(k) &= y_{v_i}(k) + y_{f_i}(k) \\
&= f_i(x_{u_{v1i}}(k), x_{u_{v2i}}(k)) + [C_{fi}\ 0] \begin{bmatrix} x_7(k) \\ x_8(k) \\ x_9(k) \end{bmatrix} \qquad (4.9)
\end{aligned}
$$

Where,

i is the die location, $i \in [1-9]$,

$x_{u_{v1i}}(k) = \alpha_i\, x_2(k) + \beta_i\, x_3(k)$,

$x_{u_{v2i}}(k) = \alpha_{anoi}\, x_4(k) + \beta_{anoi}\, x_5(k) + \gamma_{anoi}\, x_6(k).$

The function f represents the u_v system's nonlinear static gain,$x_{u_{v1i}}(k)$ is the dynamic response from $u_{v_1} \rightarrow y_{v_i}$ with its steady state gain equal to the value of u_{v_1} and $x_{u_{v2i}}(k)$ is the dynamic response from $u_{v_2} \rightarrow y_{v_i}$ with its steady state gain equal to the value of u_{v_2}.

4.9 Nonlinear State-Space Model Validation

To verify if the identified state-space model accurately predicts the input-output behaviour of the virtual process, an input sequence of 4 variables with 2 u_v and 2 u_f values, shown in Figure 4.10, was applied to both the identified state-space model and the virtual process. For process identification, each input was changed independently while holding the other inputs at their nominal values. In this validation step, the inputs were changed concurrently with magnitudes different than those used for identification. The outputs were then compared to assess the accuracy of the state-space model Equation (4.9) in approximating the virtual process.

The system responses to the input sequence used for validation are shown in Figure 4.11. Each subplot corresponds to the output at different die locations where the solid red curve represents predicted temperatures and the virtual process temperatures are represented by the blue symbols. The model fit percentage, indicating how well the process variables of the identified model *Fit* the virtual process data are presented in the title for each subplot. A value of 100% would signify that the model is a perfect fit to the data. The fit percentages for the 9 die locations are within the range of 85% to 97% indicating that the dynamics of the process are captured correctly by the identified nonlinear state-space model. A small discrepancy exists when comparing the gains shown in Figure 4.9b, which indicates that the u_f-driven virtual process is not exactly linear and that the steady state gain of the u_v model over the operational range is inaccurately expressed by the fitting function $\mathbf{Z}$ with parameters p and q set as 3 and 4, respectively.

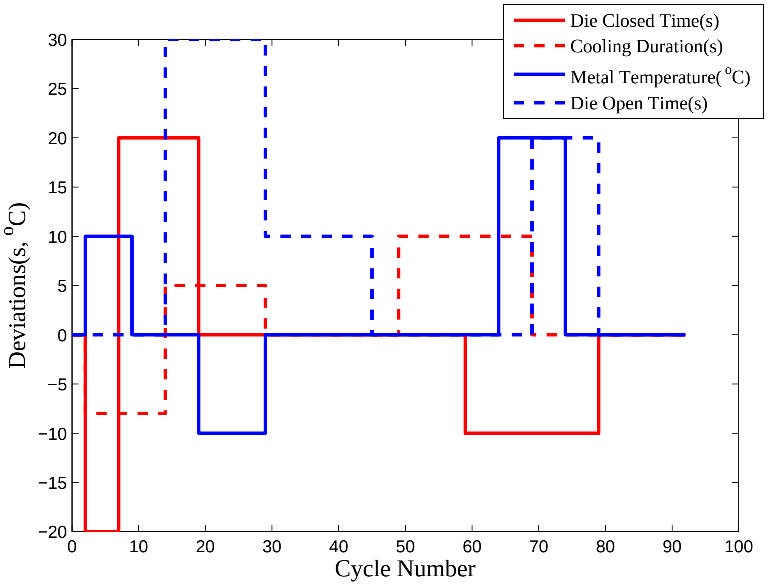


Figure 4.10: Concurrent Step Sequences of Control Variables and Feedforward Variables for Validation.

4.10 LR Expression of Liquid Encapsulation

Identification

The data set (x,y) from 9 die locations including the Maximum Temperatures (x) and liquid encapsulation area (y) is the input-output information necessary to use the least-squares algorithm to identify a LR model between temperatures and area of encapsulation. There are a variety of system input sequences that can be applied to determine the different types of process dynamics. A step change in an input will show the entire range of frequency response but will emphasize the low frequency dynamics. Testing with the virtual process has shown that the process dynamics are dominated by low frequency components and therefore a step system input sequence was employed for the LR model identification.

Starting from the baseline where $(u_{v_1}, u_{v_2}, u_{f_1}, u_{f_2}$ are equal to 150 s, 10 s, 700 °C, and 50 s, respectively), the input sequences $(u_{v_1} = \pm 20s, u_{v_2} = \pm 10s, u_{f_1} =$

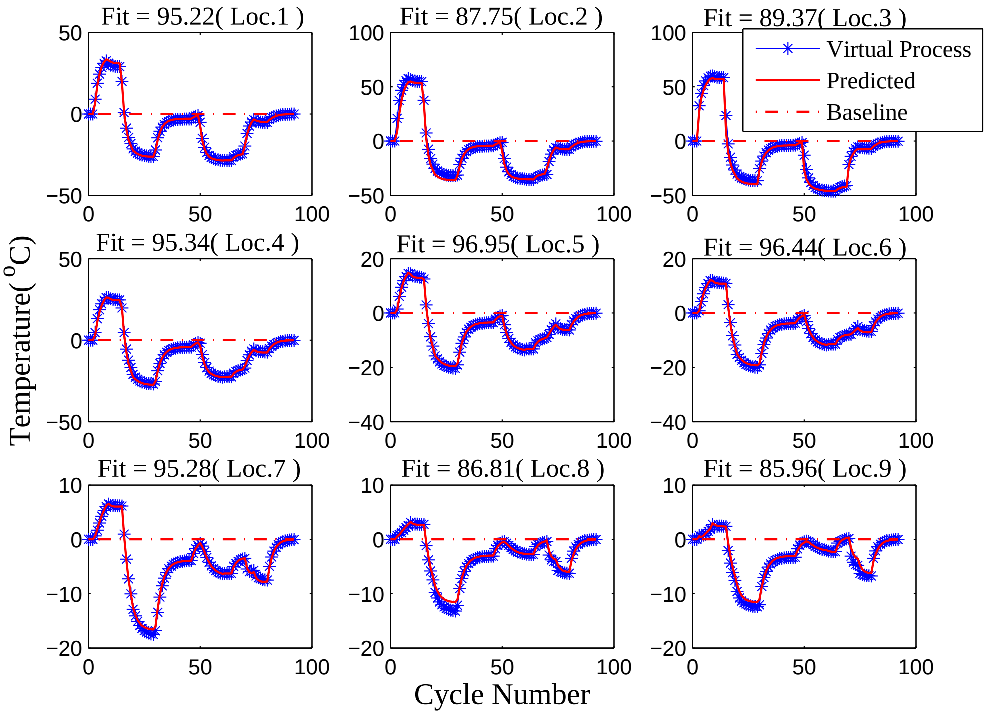


Figure 4.11: Predicted and Virtual Process Temperature Responses at 9 Die Locations under the Concurrent Step Sequences of Control Variables and Feedforward Variables for Validation.

$\pm 2^{\circ}C, u_{f_2} = +20, -30s$) were selected for $u_{v_1}, u_{v_2}, u_{f_1}, u_{f_2}$ as sequential step change inputs to the virtual process. To start, the input for u_{v_1} (Die Closed Time) was increased by 20 seconds from the baseline for 20 cycles, u_{v_1} was then returned to its baseline value for 20 cycles to allow the virtual process to return to steady state. Following this, the input for u_{v_1} was decreased by 20 seconds from the baseline for 20 cycles, and then returned to its baseline value for another 20 cycles. Similar input sequences were employed for the other three variables. Each input was changed independently while holding the other inputs at their baseline values.

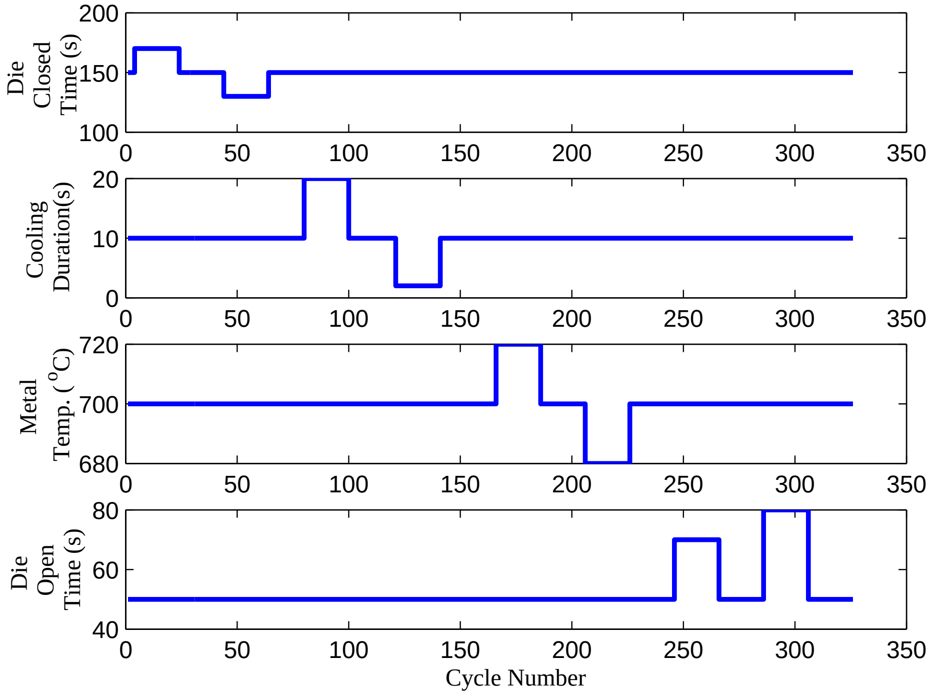


Figure 4.12: A Schematic Set of the Input Sequences for LR Model Identification.

Based on the virtual process output in response to the input sequence shown in Figure 4.12, a data set consisting of the Maximum Temperature at 9 die locations and the area of encapsulated liquid was extracted. The following LR model was then obtained:

$$\hat{y} = 451.73 + \sum_{i=1}^{N} a_i x_i \tag{4.10}$$

Where,

$$a = [0.040 \quad 0.091 \quad 0.021 \quad -0.801 \quad 2.299 \quad -1.859 \quad 1.184 \quad -1.931 \quad 0.162]'.$$

Figure 4.13 compares the LR model prediction of area of encapsulated liquid with the virtual process data. The difference between the two data sets is very small. The fit was assessed using the following equation:

$$Fit = 100 \times \left(1 - \frac{||y - \hat{y}||_2}{||y - \bar{y}||_2}\right) \tag{4.11}$$

Where y is the virtual process data, $\hat{y}$ is the predicted data and $\bar{y}$ is the mean of y. This comparison, with fits is as high as 93.28, indicates that the LR model captures the area of liquid encapsulation very well.

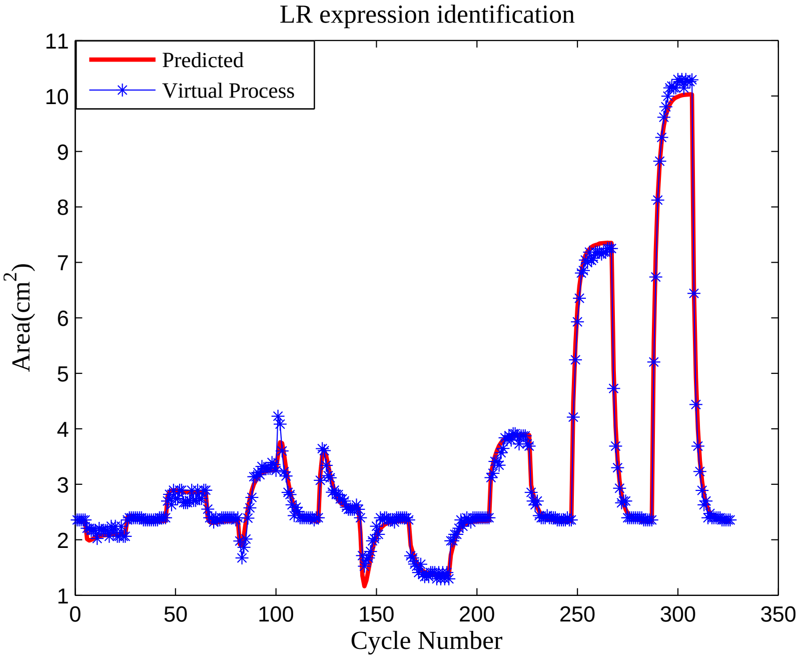


Figure 4.13: Predicted and Virtual Process Areas of Liquid Encapsulation under the Identified Input Sequences.

Validation

To verify the accuracy of the LR model for the area of liquid encapsulation, it has been tested on another data set that is different than that used to identify the LR model. The input sequence used for this validation, shown in Figure 4.14, includes cycles where multiple parameters are varied at once.

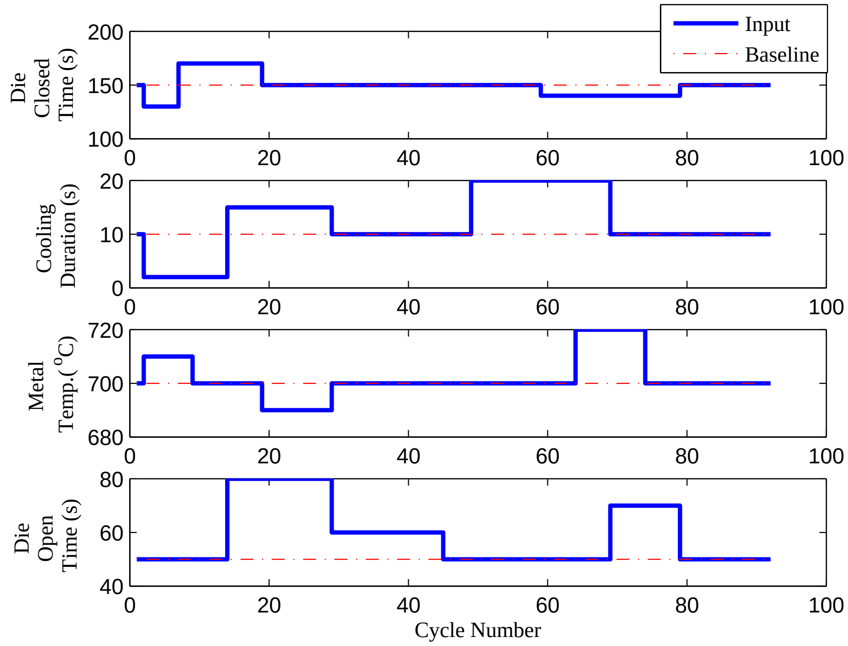


Figure 4.14: A Schematic Set of Input Sequences for LR Model Validation.

Figure 4.15 compares the area of encapsulated liquid predicted by the LR model with the virtual process data. The *Fit* of LR model in this application was computed as 86.75, indicating that the LR model accurately captures the area of liquid encapsulation.

In summary, the LR model provides good prediction for the area of liquid encapsulation hence providing a quantitative expression for macro-porosity due to liquid encapsulation.

In this chapter, a nonlinear state-space model has been developed using data from the virtual process. The model incorporates the linear dynamic behaviour of the feedforward inputs and the nonlinear static gain due to the control variable inputs. Using the virtual process for comparison, the nonlinear state-space model was shown to accurately reproduce the process dynamics over a wide variety of

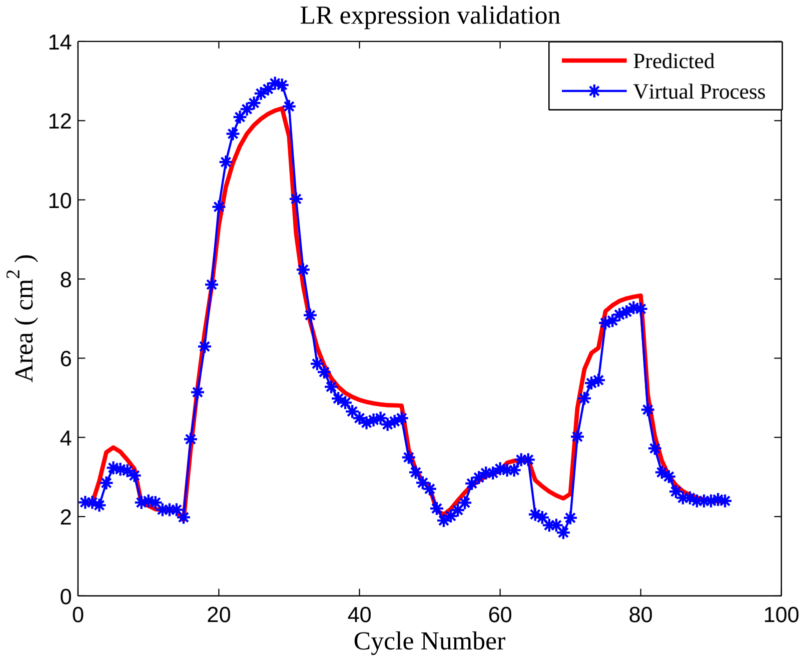


Figure 4.15: Areas of Encapsulated Liquid Predicted by the LR Model and Generated by the Virtual Process for the Validation Input Sequences.

inputs. Also, a least squares expression to predict the area of encapsulated liquid based on the Maximum Temperature at 9 locations was evaluated and shown to be accurate.

Chapter 5

Nonlinear Model-based Predictive Control[1]

This chapter presents the development of a nonlinear model-based predictive controller (NMPC) for use in controlling an operational LPDC process and its implementation in MATLAB. Section 5.1 reviews the nonlinear state-space model developed in Chapter 4 for use in predictive control. The formulation of using the nonlinear state-space model to predict the future behaviour of the process will then be presented in Section 5.2. Section 5.3 introduces the cost function approach that will be employed to optimize the controller operation. To correct for noise in process inputs, an observer-based correction has been developed. Section 5.5 outlines how constraints can be implemented in this NMPC framework. Finally, after defining the necessary components of the controller, Section 5.6 describes its implementation in MATLAB and subsequent tuning process for control parameters employed in MPC.

5.1 Nonlinear State-Space Model

In the Chapter 4, a nonlinear discrete-time state-space model was selected and developed to represent the virtual process of the demonstration LPDC process.

[1] Portions of this chapter have been prepared for publication in:
• X. Shi, Eranda Harinath, D.M. Maijer and G. Dumont, “Nonlinear Model Predictive Control in a Low Pressure Die Casting Process”, Journal of Process Control, submitted 2012.

In general, state-space modeling is a well known mathematical technique that is inherently multivariable and can be formulated to represent discrete time (or batch) processes. For this work, the state-space model with linear dynamic behaviour and nonlinear static gain is presented in Equation (5.1).

$$\begin{aligned} x(k+1) &= Ax(k)+Bu(k) \\ y(k) &= y_v(k)+y_f(k) \\ &= f(x(k))+Cx(k) \end{aligned} \tag{5.1}$$

where the index k counts discrete time steps and A and B are matrices that operate on the vectors $x(k)$ and $u(k)$ to predict the next step $x(k+1)$. The A matrix represents the unforced or natural response of the system while the B matrix represents the forced response. C is a matrix that linearly operates on the $x(k)$ vector to represent the system feedforward response $y_f(k)$ and the function f produces the nonlinear response $y_v(k)$ based on $x(k)$. The $x(k)$ and $x(k+1)$ vectors represent the state of the dynamics of the system being modeled at the current and next time steps, respectively.

The sizes of the matrices A, B and C are determined by the sizes of the state-space model vectors. A is a square matrix with dimensions equal to the length of the state vector $x(k)$. The B matrix's dimensions are related to the vector lengths of $x(k)$ and $u(k)$ for the row and columns, respectively. While the number of rows in C is equal to the length of the vector $y_f(k)$ and the number of columns is the equal to the length of vector $x(k)$. The function f generates a vector of the same length as $y_v(k)$ that represents the nonlinear expressions of the state vectors over model outputs at different die locations.

In this work, the basic control sequence of actions at time step k is as follows:

1. Measure $y(k)$ and $u(k-1)$.

2. Compute the required process inputs $u(k)$.

3. Apply $u(k)$ to the plant.

This sequence shows that there is always a delay between measuring $y(k)$ and applying $u(k)$. For this reason there is no direct 'feed-through' from $u(k)$ to $y(k)$ in Equation (5.1). This expression is necessary to simplify the computation of the prediction and optimization equations in the following sections. If the process has direct feed-through then a number of methods exist to modify the original state-space model to remove the term from $u(k)$ to $y(k)$ [72].

5.2 Prediction

In order to solve the predictive control problem, a means of computing the predicted values of the controlled variables $\hat{y}(k+i|k)$ is sought, based the best estimate of the current state, $\hat{x}(k|k)$, and assuming future inputs $\hat{u}(k+i|k)$. The nonlinear state-space model formulation given in Equation (5.1) is only useful for predicting the process outputs one time step into the future. The MPC framework requires a model that predicts many time steps into the future to formulate control operations for the current time. This prediction could be done iteratively as in Equation (5.1) where the current state and inputs are used to calculate the next state and then repeated for as many steps as needed. The next process variables can then be determined as the sum of the next state vector $x(k)$ multiplied by the C matrix and the nonlinear function f of the next state vector at each iteration.

Eqn. 5.1 introduces a notation to distinguish between known and predicted values. For example, $\hat{x}(k+1|k)$ is the predicted state at the next time step. The "ˆ" signifies that this value is a prediction. The $k+1|k$ notation indicates that the prediction is for the time step one cycle into the further "$k+1$" and the current time step is "k".

$$\begin{aligned}
\hat{x}(k+1|k) &= Ax(k)+B\,\hat{u}(k|k) \\
\hat{x}(k+2|k) &= Ax(k+1|k)+B\,\hat{u}(k+1|k) \\
\hat{x}(k+3|k) &= Ax(k+2|k)+B\,\hat{u}(k+2|k) \\
&\vdots
\end{aligned} \tag{5.2}$$

This method of iteratively predicting the state vector can be applied if all the future inputs were known in advance. For MPC, the future control variables will be calculated and are therefore currently unknown. In addition, although the current values for the feedforward variables are known, the future values must be predicted. Therefore, the input vector $u(k)$ must be separated into known and unknown values and methods to define the unknown values must be developed.

The first step to develop the necessary predictive model capability is to separate the input vector, $u(k)$, into two separate components, $\hat{u}_v(k)$and $\hat{u}_f(k)$, to represent the predicted control variables and the current feedforward variables, respectively. For the prediction equations that do not use the current values of the feedforward variables, $\hat{u}_f(k+1|k)$ is introduced to represent the predicted feedforward variables. With the input vector separated, the B matrix must be separated into B_v and B_f to operate on their respective input vectors. Equation (5.2) can then be written as:

$$\begin{aligned}
\hat{x}(k+1|k) &= Ax(k)+B_v\hat{u}_v(k|k)+B_f u_f(k) \\
\hat{x}(k+2|k) &= A\hat{x}(k+1|k)+B_v\hat{u}_v(k+1|k)+B_f\hat{u}_f(k+1|k) \\
&\vdots \\
\hat{x}(k+H_p|k) &= A\hat{x}(k+H_p-1|k)+B_v\hat{u}_v(k+H_p-1|k)+B_f\hat{u}_f(k+H_p-1|k)
\end{aligned} \tag{5.3}$$

In the first line of Equation (5.2), $\hat{u}_v(k|k)$ is used rather than $u(k)$, because $u(k)$ is unknown at the beginning of this calculation. Equation (5.3) includes terms for the unknown future values of the feedforward variables, $\hat{u}_f(k+n|k)$. If the nature of the disturbances were known, a prediction for the future values of the feedforward variables could be made. However for simplicity, the future values of the feedforward variables will be assumed to be held constant at the current measured values. This implies that $\hat{u}_f(k+n|k) = u_f(k)$ for all future times steps n.

Additionally, we need to formulate each of the state prediction equations in terms of the current state, $x(k)$ which is known. This reformulation is accomplished by substituting the state prediction equation for $\hat{x}(k+1|k)$ into the $\hat{x}(k+2|k)$ equation. This substitution is then repeated for $\hat{x}(k+3|k)$, $\hat{x}(k+4|k)$, etc. The state-space model with the modifications made so far is shown in Equation (5.4).

$$\begin{aligned}
\hat{x}(k+1|k) &= Ax(k)+B_v\hat{u}_v(k|k)+B_f u_f(k) \\
\hat{x}(k+2|k) &= A\hat{x}(k+1|k)+B_v\hat{u}_v(k+1|k)+B_f u_f(k) \\
&= A^2x(k)+AB_v\hat{u}_v(k|k)+B_v\hat{u}_v(k+1|k)+(A+I)B_f u_f(k) \\
&\vdots \\
\hat{x}(k+H_p|k) &= A\hat{x}(k+H_p-1|k)+B_v\hat{u}_v(k+H_p-1|k)+B_f u_f(k) \\
&= A^{H_p}x(k)+A^{H_p-1}B_v\hat{u}_v(k|k)+\cdots+B_v\hat{u}_v(k+H_p-1|k) \\
&\quad +(A^{H_p-1}+\cdots+A+I)B_f u_f(k)
\end{aligned} \tag{5.4}$$

Now recall the assumption that the input will only change at times k, $k+1,\cdots,\ k+H_u-1$, where H_u is the control horizon, and will remain constant after that. Thus, $\hat{u}_v(k+i|k)=\hat{u}_v(k+H_u-1)$ for $H_u \le i \le H_p-1$, where H_p is the prediction horizon. This leads to Equation (5.5):

$$\begin{aligned}
\hat{x}(k+1|k) &= Ax(k)+B_v\hat{u}_v(k|k)+B_f u_f(k) \\
\hat{x}(k+2|k) &= A^2x(k)+AB_v\hat{u}_v(k|k)+B_v\hat{u}_v(k+1|k)+(A+I)B_f u_f(k) \\
&\vdots \\
\hat{x}(k+H_u|k) &= A^{H_u}x(k)+A^{H_u-1}B_v\hat{u}_v(k|k)+\cdots+B_v\hat{u}_v(k+H_u-1|k) \\
&\quad +(A^{H_u-1}+\cdots+A+I)B_f u_f(k)
\end{aligned}$$

Note: after this point: $\hat{u}_v(k+i|k)=\hat{u}_v(k+H_u-1)$ for $H_u \le i \le H_p-1$

$$
\begin{aligned}
\hat{x}(k+H_u+1|k) &= A^{H_u+1}x(k)+A^{H_u}B_v\hat{u}_v(k|k)+\cdots+AB_v\hat{u}_v(k+H_u-1|k) \\
&\quad +B_v\hat{u}_v(k+H_u-1|k)+(A^{H_u}+\cdots+A+I)B_f u_f(k) \\
&\vdots \\
\hat{x}(k+H_p|k) &= A^{H_p}x(k)+A^{H_p-1}B_v\hat{u}_v(k|k)+\cdots+A^{H_p-H_u}B_v\hat{u}_v(k+H_u-1|k) \\
&\quad +A^{H_p-H_u-1}B_v\hat{u}_v(k+H_u-1|k)+\cdots+B_v\hat{u}_v(k+H_u-1|k) \\
&\quad +(A^{H_p}+\cdots+A+I)B_f u_f(k)
\end{aligned}
\tag{5.5}
$$

The state prediction described by Equation (5.5) can be rewritten in a matrix-vector form as:

$$
\begin{bmatrix} \hat{x}(k+1|k) \\ \hat{x}(k+2|k) \\ \vdots \\ \hat{x}(k+H_u|k) \\ \hat{x}(k+H_u+1|k) \\ \vdots \\ \hat{x}(k+H_p|k) \end{bmatrix}
=
\begin{bmatrix} A \\ A^2 \\ \vdots \\ A^{H_u} \\ A^{H_u+1} \\ \vdots \\ A^{H_p} \end{bmatrix} x(k) +
\begin{bmatrix}
B_v & 0 & \cdots & 0 & 0 \\
AB_v & B_v & \cdots & 0 & 0 \\
 & & \vdots & & \\
A^{H_u-1}B_v & A^{H_u-2}B_v & \cdots & AB_v & B_v \\
A^{H_u}B_v & A^{H_u-1}B_v & \cdots & A^2B_v & AB_v+B_v \\
 & & \vdots & & \\
A^{H_p-1}B_v & A^{H_p-2}B_v & \cdots & A^{H_p-H_u-1}B_v & \sum_{i=0}^{H_p-H_u} A^iB_v
\end{bmatrix}
$$

$$
\times \begin{bmatrix} \hat{u}_v(k|k) \\ \hat{u}_v(k+1|k) \\ \vdots \\ \hat{u}_v(k+H_u-2|k) \\ \hat{u}_v(k+H_u-1|k) \end{bmatrix}
+ \begin{bmatrix} B_f \\ AB_f+B_f \\ \vdots \\ \sum_{i=0}^{H_u-1} A^iB_f \\ \sum_{i=0}^{H_u} A^iB_f \\ \vdots \\ sum_{i=0}^{H_p-1} A^iB_f \end{bmatrix} u_f(k)
\tag{5.6}
$$

By introducing two vectors $\hat{X}(k)$ and $\hat{U}(k)$ which contain the predicted state and control variable vectors, respectively, Equation (5.6) can be simplified to:

$$
\begin{aligned}
\hat{X}(k) &= \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^{H_u} \\ A^{H_u+1} \\ \vdots \\ A^{H_p} \end{bmatrix} x(k) + \begin{bmatrix} B_v & 0 & \cdots & 0 & 0 \\ AB_v & B_v & \cdots & 0 & 0 \\ & & \vdots & & \\ A^{H_u-1}B_v & A^{H_u-2}B_v & \cdots & AB_v & B_v \\ A^{H_u}B_v & A^{H_u-1}B_v & \cdots & A^2B_v & AB_v + B_v \\ & & \vdots & & \\ A^{H_p-1}B_v & A^{H_p-2}B_v & \cdots & A^{H_p-H_u-1}B_v & \sum_{i=0}^{H_p-H_u} A^iB_v \end{bmatrix} \\
&\quad \times \hat{U}_V(k) + \begin{bmatrix} B_f \\ AB_f + B_f \\ \vdots \\ \sum_{i=0}^{H_u-1} A^iB_f \\ \sum_{i=0}^{H_u} A^iB_f \\ \vdots \\ sum_{i=0}^{H_p-1} A^iB_f \end{bmatrix} u_f(k) \\
&= F_{U_V \to X}(\hat{U}_V(k))
\end{aligned}
\tag{5.7}
$$

where,

$$
\hat{X}(k) = \begin{bmatrix} \hat{x}(k+1|k) \\ \hat{x}(k+2|k) \\ \vdots \\ \hat{x}(k+H_u|k) \\ \hat{x}(k+H_u+1|k) \\ \vdots \\ \hat{x}(k+H_p|k) \end{bmatrix} \quad and \quad \hat{U}_V(k) = \begin{bmatrix} \hat{u}_v(k|k) \\ \hat{u}_v(k+1|k) \\ \vdots \\ \hat{u}_v(k+H_u-2|k) \\ \hat{u}_v(k+H_u-1|k) \end{bmatrix}
$$

Ultimately, the state-space model should be used to predict the area of liquid encapsulation in the casting at some time in the future. Note that the area of liquid encapsulation occurring in the future can be linearly expressed in terms of the process variables in a future cycle, $\hat{Y}(k)$. The prediction of $\hat{Y}(k)$ is based on the expression for $y(k)$ as follows,

$$y(k) = y_v(k) + y_f(k) = f(x(k)) + Cx(k)$$

Therefore, $\hat{Y}(k)$ can be expressed as:

$$\begin{aligned}
\hat{Y}(k) &= \begin{bmatrix} \hat{y}(k+1|k) \\ \hat{y}(k+2|k) \\ \vdots \\ \hat{y}(k+H_p-1|k) \\ \hat{y}(k+H_p|k) \end{bmatrix} \\
&= \begin{bmatrix} f(\hat{x}(k+1|k)) \\ f(\hat{x}(k+2|k)) \\ \vdots \\ f(\hat{x}(k+H_p-1|k)) \\ f(\hat{x}(k+H_p|k)) \end{bmatrix} + \begin{bmatrix} C(\hat{x}(k+1|k)) \\ C(\hat{x}(k+2|k)) \\ \vdots \\ C(\hat{x}(k+H_p-1|k)) \\ C(\hat{x}(k+H_p|k)) \end{bmatrix} \\
&= F_{X\to Y}(\hat{X}(k)) \qquad (5.8)
\end{aligned}$$

According to the previous chapter, the area of liquid encapsulation can be linearly expressed by:

$$\begin{aligned}
area(k) &= \gamma_0 + \sum_{i=1}^{N} \gamma_i \; y_i(k) \\
&= \gamma_0 + \Gamma \; Y(k) \qquad (5.9)
\end{aligned}$$

where,

$$\Gamma = [\gamma_1 \quad \cdots \quad \gamma_9], \quad Y(k) = [y_1(k) \quad \cdots \quad y_9(k)]'$$

With Equation (5.8) and Equation (5.9) the $A\hat{R}EA(k)$ can be computed as follows,

$$
\begin{aligned}
A\hat{RE}A(k) &= \begin{bmatrix} ar\hat{e}a(k+1|k) \\ ar\hat{e}a(k+2|k) \\ \vdots \\ ar\hat{e}a(k+H_p-1|k) \\ ar\hat{e}a(k+H_p|k) \end{bmatrix} = \gamma_0 + \Gamma \begin{bmatrix} \hat{Y}(k+1|k) \\ \hat{Y}(k+2|k) \\ \vdots \\ \hat{Y}(k+H_p-1|k) \\ \hat{Y}(k+H_p|k) \end{bmatrix} \\
&= F_{Y\to AREA}\hat{Y}(k) = F_{Y\to AREA}(F_{X\to Y}(F_{U_V\to X}(\hat{U}_V(k)))) \\
&= F_{U_V\to AREA}\hat{U}_V(k) \qquad (5.10)
\end{aligned}
$$

Note that the functions involved in Equation (5.10) will change when the nonlinear state-space model, prediction horizon, or control horizon changes.

5.3 Cost Function

The basic principle of predictive control is to compute future control variables such that the measured process variables follow future reference trajectories. The cost function V penalizes deviations of the predicted process variables $\hat{y}(k+i|k)$ from a reference trajectory $r(k+i|k)$ vector. However, this methodology does not work for the current case because the term to be penalized is the area of liquid encapsulation which cannot be measured. However, the area of liquid encapsulation can be expressed in terms of other measurable process variables, $i.e.$ die temperature.

The primary goal of the model-based predictive controller is to minimize the area of liquid encapsulation. A secondary goal of the model-based predictive controller is to increase the production rate of the batch process by decreasing Die Closed Time to effect an overall reduction in cycle time while still accomplishing the primary goal. This translates into minimizing the areas of liquid encapsulation over the prediction horizon while keeping Die Closed Time to a minimum.

A function that accomplishes the primary and secondary goals by assigning ’cost’ to the predicted area of liquid encapsulation and the Die Closed Time is described in Equation (5.11). The optimal control variables, are then calculated by minimizing the cost function as follows.

$$V(k) \quad = \quad \sum_{i=1}^{H_p} \|ar\hat{e}a(k+i|k)\|_Q^2 \quad + \quad \sum_{i=1}^{H_p} \|\hat{u}_{v_1}(k+i|k)\|_R^2 \tag{5.11}$$

where $\hat{u}_{v_1}$ is the predicted Die Closed Time.

The cost function introduces the weighted Euclidean norm [21], x_Q^2, defined as $\|x\|_Q^2 = x^T Q\, x$, where x is a vector and Q is a diagonal matrix. The matrices Q and R weigh the predicted area of liquid encapsulation and predicted Die Closed Time, respectively. The diagonal elements $Q(i)$ and $R(i)$ on these weighting matrices are defined to be dependent on the prediction index i in sequence. This time dependency allows different weights to be applied over different regions of the prediction horizons. Typically these elements remain constant over the entire prediction and control horizons and the i subscript can be ignored.

The first term of the cost function, $ar\hat{e}a(k+i|k)$, calculates the weighted norm of the area of liquid encapsulation. These weighted norms are summed over the prediction horizon from 1 to H_p to define the cost of the primary goal. The weighted norms of the Die Closed Time, $\hat{u}_{v_1}(k+i|k)$, are summed over the prediction horizon from 0 to H_p to define the cost of the secondary goal, i.e. production rate.

Using the vectors defined previously in Equation (5.7) and Equation (5.10), the cost function equation can be expressed as a function of $\hat{U}_V(k)$ as follows,

$$\begin{aligned} V(k) \quad &= \quad \sum_{i=1}^{H_p} \|AR\hat{E}A(k)\|_{Q(i)}^2 \quad + \quad \sum_{i=1}^{H_p} \|\hat{u}_{v_1}(k+i|k)\|_{R(i)}^2 \\ &= \quad \|F_{U_V \to AREA}(\hat{U}_V(k))\|_{Q(i)}^2 + \|\hat{U}_{V_1}(k)\|_{R(i)}^2 \end{aligned} \tag{5.12}$$

The cost function in Equation (5.12) can be reframed as a constrained optimization problem as follows:

$$\min_{\hat{\mathbf{u}}} \quad f(x)$$
$$s.t: \begin{cases} c(x) \leqslant 0 \\ c_{eq}(x) = 0 \\ Ax \leqslant b \\ A_{eq}x \leqslant b_{eq} \\ lb \leqslant x \leqslant ub \end{cases} \tag{5.13}$$

where x, b, b_{eq}, lb and ub are vectors, A and A_{eq} are matrices, $c(x)$ and $c_{eq}(x)$ are functions that return vectors, and $f(x)$ is a function that returns a scalar. $f(x), c(x)$ and $c_{eq}(x)$ can be nonlinear functions.

In the current work, $f(x)$ is defined as $V(u)$ in Equation (5.12). Note that the first term of $V(k)$ is the nonlinear expression for $F_{U_V \to AREA}$, therefore it can not be treated with the regular linear Quadratic Programming (QP) approach[72]. For this kind of nonlinear constrained optimization problem, the Sequential Quadratic Programming method (SQP) has arguably become the most successful method since its popularization in the late 1970s [94]. As with most optimization methods, SQP is not a single algorithm, but rather a conceptual approach from which numerous specific algorithms have evolved. SQP is applied by converting a constrained nonlinear optimization problem into Quadratic Programming (QP) subproblems that can be solved by the active-set method at each iteration [94]. The active-set method was exploited to compute a quasi-Newton approximation to the Hessian matrix from the Lagrangian function [95]. The active-set method has several features. First, the iterations remain stable once an initial feasible solution has been found. Second, since it starts from the active set of constraints, successive iterations continue towards a feasible solution under the same active set. The feasible solution can be evaluated by Karush-Kuhn-Tucker (or KKT)[72] technique to verify that new solution is the global optimum. Finally, the rate of finding a solution with the active set method is dominated by the speed with which this factorization can be performed. It is noted that the main iteration of the active set method searches among points on the boundary of the feasible region.

The algorithm used to solve such a nonlinear constrained optimization problem shown in Equation (5.13) is available in the Optimization Toolbox in MATLAB. The function **fmincon** in MATLAB can solve for a constrained minimum of a scalar function of several variables starting at an initial estimate. In short, **fmincon** uses the SQP method to solve QP subproblems in each iteration. An estimate of the Hessian matrix of the Lagrangian function is updated at each iteration. A line search is performed using a merit function and the QP subproblem is solved using an active set strategy [[95, 96]]. Note that based on where **fmincon** starts, it may terminate at the global minimum or at one of the local minima. In this work, to ensure that the global minimum is found, **fmincon** is first run with multiple preset starting points. The starting point with the lowest function value is then selected.

5.4 Constant Output Disturbance Observer

Due to model inaccuracies, noise on the output signals and / or disturbances in the output, discrepancies will exist between the state-space model predicted outputs calculated with Equation (5.8) and the virtual process outputs. The controller must account for this discrepancy in order to ensure accurate prediction of the process outputs. The technique to compensate for these inaccuracies is to use an observer that estimates and corrects the state of the process by comparing the predicted outputs with the virtual process outputs. A simple observer design would be to assume that any error between the predicted and the virtual process outputs is the result of a disturbance on the outputs. By assuming that this output disturbance is constant over the prediction horizon, the future predicted outputs are calculated by adding the output error to the values predicted by Equation (5.8). This type of estimation is known as ’deadbeat’ estimation because the disturbance estimate remains the same after only one sample.

The constant output disturbance observer is well suited to deal with situations where the output error is mainly due to model inaccuracies. However, it is incapable of dealing with inaccuracies caused by ’noisy’ output signals. If noise is an issue when the controller is implemented in an industrial environment, the temperatures could be filtered in the PLC to remove the noise as is typical of most industrial controller implementations or a more elaborate state observer such as a nonlinear Kalman filter may be required [[97]]. In the case of the virtual process studied in this research, measurement noise is not simulated and all output errors are due to discrepancies between the internal process model of the controller and the virtual process. Therefore, the constant output disturbance observer is acceptable.

To integrate the observer into the prediction equations, a simple modification to the predicted output definition of Equation (5.8) is required. As defined previously, the predicted output of the system is the sum of the predicted control variables' responses y_v and predicted feedforward variables' responses. The observer corrects the whole predicted output by adding the output error to all future predicted values. The modified equation is given as Equation (5.14) where $Y_{error}(k)$ is defined as the difference between the predicted and the virtual process outputs at time step k.

$$
\begin{aligned}
\hat{Y}(k) &= \begin{bmatrix} \hat{y}(k+1|k) \\ \hat{y}(k+2|k) \\ \vdots \\ \hat{y}(k+H_p-1|k) \\ \hat{y}(k+H_p|k) \end{bmatrix} + Y_{err}(k) \\
&= \begin{bmatrix} f(\hat{x}(k+1|k)) \\ f(\hat{x}(k+2|k)) \\ \vdots \\ f(\hat{x}(k+H_p-1|k)) \\ f(\hat{x}(k+H_p|k)) \end{bmatrix} + \begin{bmatrix} C(\hat{x}(k+1|k)) \\ C(\hat{x}(k+2|k)) \\ \vdots \\ C(\hat{x}(k+H_p-1|k)) \\ C(\hat{x}(k+H_p|k)) \end{bmatrix} + Y_{err}(k) \\
&= F_{XY}(\hat{X}(k)) + Y_{err}(k) \qquad (5.14)
\end{aligned}
$$

5.5 Constraints

In this research, a Nonlinear Model-based Predictive Control is being applied to minimize the cost function V in Equation (5.12) subject to constraints on u_v, which are defined in Equation (5.13). Based on the formulation presented, the constraints for the control variables apply only to Die Closed Time. A 'hard' lower constraint is necessary to ensure that the controller does not reduce the Die Closed Time to the extent that the die opens when the casting is still molten. Based on a sensitivity analysis with the Virtual process, this constraint should be equal to -40 s. In theory, ’hard’ upper constraints are not necessary because there is not an upper bound for Die Closed Time or Cooling Duration. However, operational ranges without bounds can lead to unanticipated performance issues, so ’hard’ upper constraints of 40 s and 30 s were imposed on Die Closed Time and Cooling, respectively.

Another constraint has been set to limit the rate of control variable variation over adjacent cycles. In this research, a nonlinear state-space model with linear dynamics has been developed to capture the nonlinear dynamics of the virtual process. To ensure model accuracy, the rate of variation of u has been constrained as follows,

$$\triangle u \leqslant b_{\triangle u} \tag{5.15}$$

To convert this expression to the global form of $Au \leqslant b_u$ in Equation (5.13), the procedure is as follows,

Eqn. 5.13 implies that,

$$b_{\triangle u} \leqslant u(k) - u(k+1) \leqslant b_{\triangle u}$$

$\Longrightarrow$

$$\begin{cases} u(k) - u(k+1) \leqslant b_{\triangle u} \\ -u(k) + u(k+1) \leqslant b_{\triangle u} \end{cases}$$

$\Longrightarrow$

$$\begin{bmatrix} +1 & -1 \\ -1 & +1 \end{bmatrix} \cdot \begin{bmatrix} u(k) \\ u(k+1) \end{bmatrix} \leqslant \begin{bmatrix} b_{\triangle u} \\ b_{\triangle u} \end{bmatrix}$$

which is the form of $A_u \leqslant \triangle b_u$.

In summary, the form of the constrained optimization problem for the LPDC process studied in this research is as follows,

$$\begin{aligned} \min_{\hat{\mathbf{u}}} \quad & V(u) \\ s.t: \quad & Au \leqslant b_u \end{aligned} \tag{5.16}$$

where the expression of V is given by Equation (5.12).

5.6 Controller Implementation and Tuning in MATLAB

5.6.1 Controller Implementation in MATLAB

A MATLAB program was developed to tune and assess the performance of the MPC controller prior to applying it to the virtual process. The program was formulated to assess the process response in open and closed loop modes, and to examine the effects of the tuning parameters on the closed loop performance. MATLAB offers convenient functions to implement the control algorithm including the nonlinear state-space model and constraint optimization. The MPC performance using this approach is expected to mimic its performance on the virtual process when the nonlinear state-space model accurately represents the virtual process. A flowchart of the MATLAB program is shown in Figure 5.1.

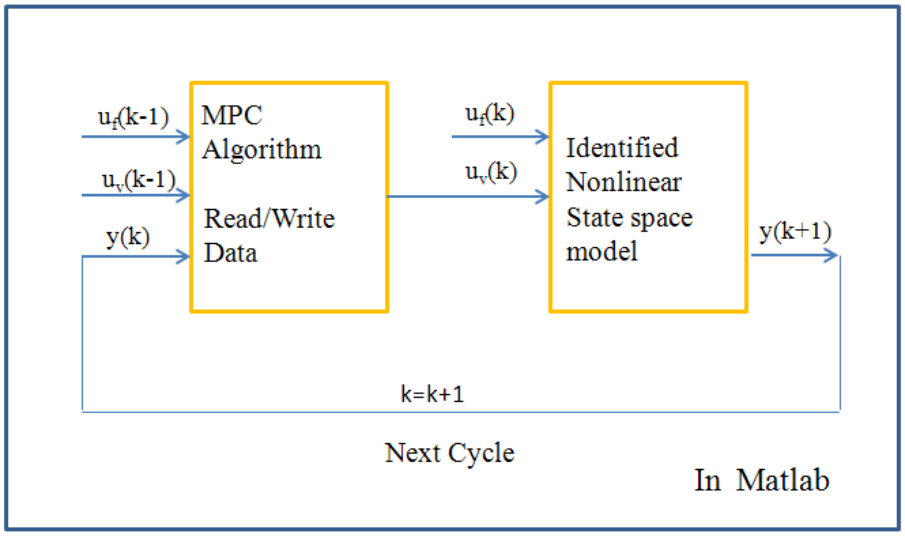


Figure 5.1: Data Flow Diagram for MATLAB Program with MPC Involved.

The controller operation shown in Figure 5.1 occurs in two stages. In the 1^{st} stage, the MPC Algorithm computes the control inputs for the state-space model in the current cycle. Upon collecting the data $u_v(k-1)$, $u_f(k-1)$ and $y(k)$ at the current time step k, the optimal $u_v(k)$ is computed by solving the constraint optimization problem shown in Equation 5.16.

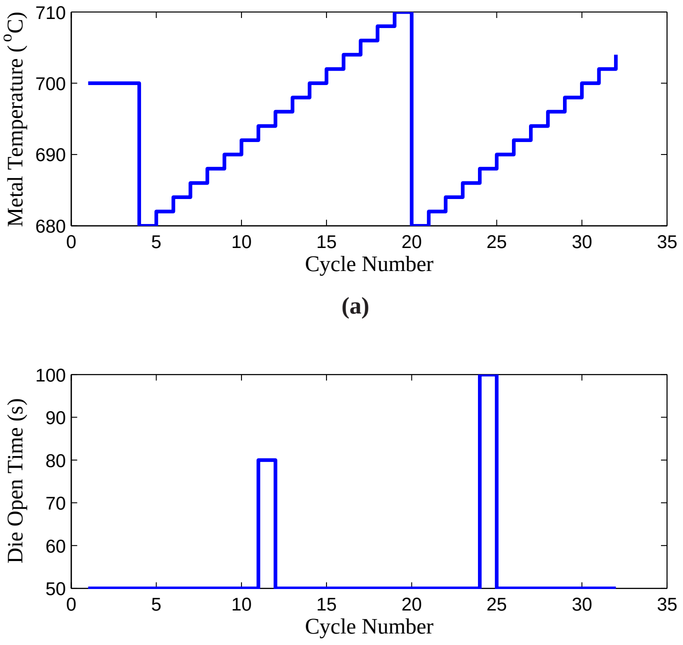
In the 2^{nd} stage of controller operation, the nonlinear state-space model is run with inputs $u_v(k)$ and $u_f(k)$ to calculate the new process output $y(k+1)$ for use in the next cycle's calculations. When a new cycle starts, MATLAB activates the MPC algorithm to compute the updated $u_v(k)$ again. By repeatedly operating in this mode, MATLAB simulates the application of MPC to the casting process represented with the nonlinear state-space model. The corresponding MATLAB code for the MPC controller is presented in Appendix.

5.6.2 Controller Tuning

The MPC controller described above must be 'tuned' to optimize the closed loop performance of the virtual process. The parameters associated with the cost function in Equation (5.12) are defined as follows:

- H_p - Prediction Horizon
- H_u - Control Horizon
- Q - Predicted Amount of Liquid Encapsulation Weighting Matrix
- R - Predicted u_{v_1} Weighting Matrix

To tune the MPC controller, the closed loop response using different values of the tuning parameters was calculated with the MATLAB program. Each simulation was started from the steady state operating point (0,0). Worst case disturbance scenario, shown in Figure 5.2, was imposed on the process, which assumes a drop in incoming Metal Temperature in Figure 5.2a coincides with an event that causes longer Die Open Time in Figure 5.2b. The resulting control inputs and process output (area of encapsulated liquid) were evaluated to determine optimal tuning parameters.



(a)

(b)

Figure 5.2: Worst Case Disturbance Scenario: (a) Metal Temperature and (b) Die Open Time.

In the following, the results of each tuning parameter assessment will be presented in figures composed of two graphs. The top graph displays the changes to control variables while in closed loop and the bottom graph shows the area of the liquid encapsulation for each casting assuming closed loop and open loop operation. The elements Q and R were set to 1 and 0, respectively, unless otherwise specified.

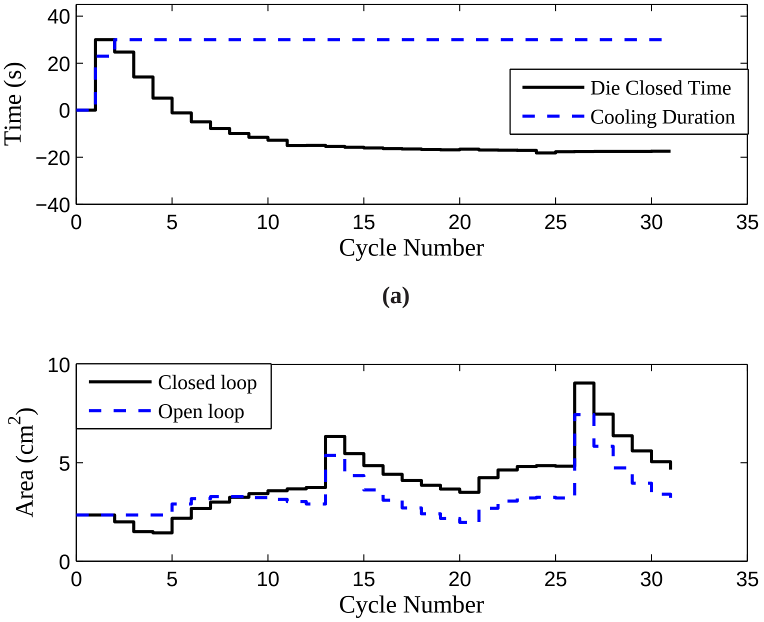
H_p Tuning

H_p defines the number of predictions the controller will make into the future. Usually the value of H_p is set to be approximately three times as long as the time constants found during dynamic behaviour testing, such as presented in Chapter 4, which allows the controller to predict into the future where the process variables have settled to within at least 95% of their final values. Based on the results presented in Chapter 4, the corresponding time constants for the 9 die locations being considered were from 3 - 6 cycles. Four H_p values (5, 10, 15, and 20) were used with the MATLAB program to assess the effect of the prediction horizon on controller performance. An H_u of 2 was used during this assessment.

The responses of the closed loop control variables with different H_p settings are shown in Figures 5.3a to 5.6a, and Figures 5.3b to 5.6b present the corresponding results for area of encapsulated liquid of the closed loop and open loop operation of the process. Table 5.1 summarizes the total area of liquid encapsulation for open loop and closed loop with different H_p settings. When H_p is equal to 5, the total area of encapsulated liquid in the closed loop mode increased by 26% compared to open loop (refer to Table 5.1). This poor closed loop performance, as shown in Figure 5.3b, is because the process response at the prediction horizon has not reached its final values based the current process inputs.

Table 5.1: Summary of the Total Area of Liquid Encapsulation for Open Loop and Closed Loop with Different H_p Settings.

	Closed loop (H_u, H_p)				Open loop
	(2, 5)	(2, 10)	(2, 15)	(2, 20)	
Total area of liquid encapsulation (cm^2)	130.5	99.8	92.5	92.5	103.8



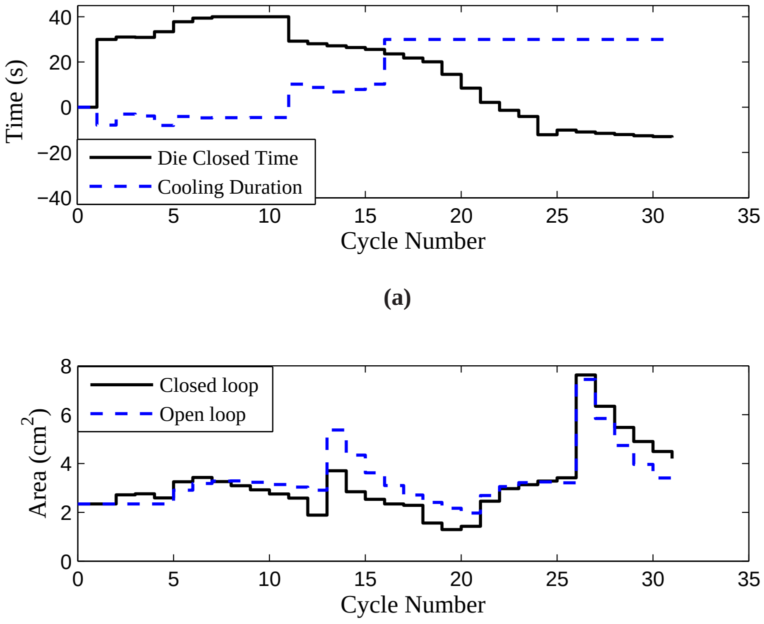
(a)

(b)

Figure 5.3: (a) The Responses of the Closed Loop Control Variables and (b) the Results for Area of Encapsulated Liquid of the Closed Loop and Open Loop When Running NMPC in MATLAB with Tuning Parameters Set of $H_u = 2$ and $H_p = 5$.

When H_p is equal to 10 (refer to Figure 5.4), the closed loop performance improves, especially in the longer term performance, leading to a 4% reduction in the area of encapsulated liquid compared with the open loop as shown in Table 5.1.

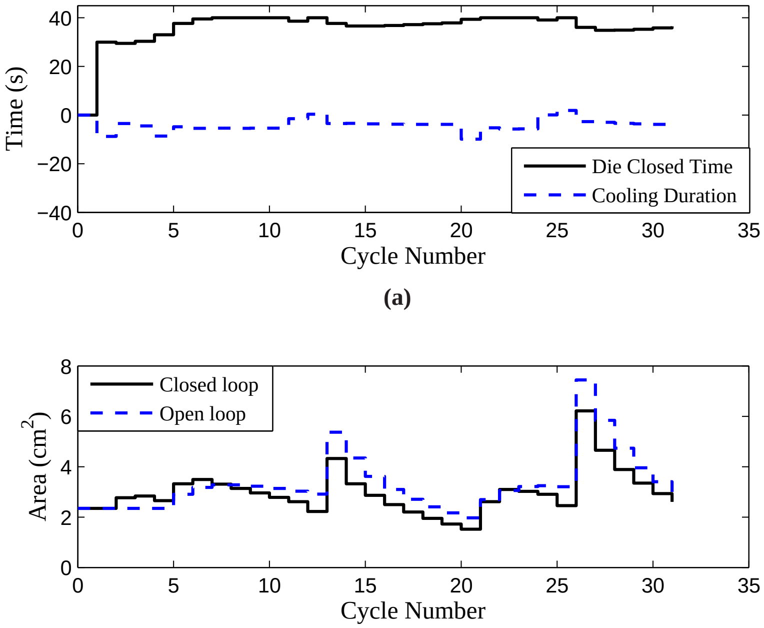
Additional improvements are observed when H_p is increased to 15 (refer to Figure 5.8). In this case, the total area of encapsulated liquid is reduced by 11% in closed loop mode relative to open loop as presented in Table 5.1. These three cases indicate that as H_p increases the closed loop performance improves. However, when H_p is increased to 20, as shown in Figure 5.6 and Table 5.1, the closed loop performance shows no improvement compared to an H_p of 15. Based on these results, H_p was set equal to 15 and this value was applied for the remaining analysis.



(a)

(b)

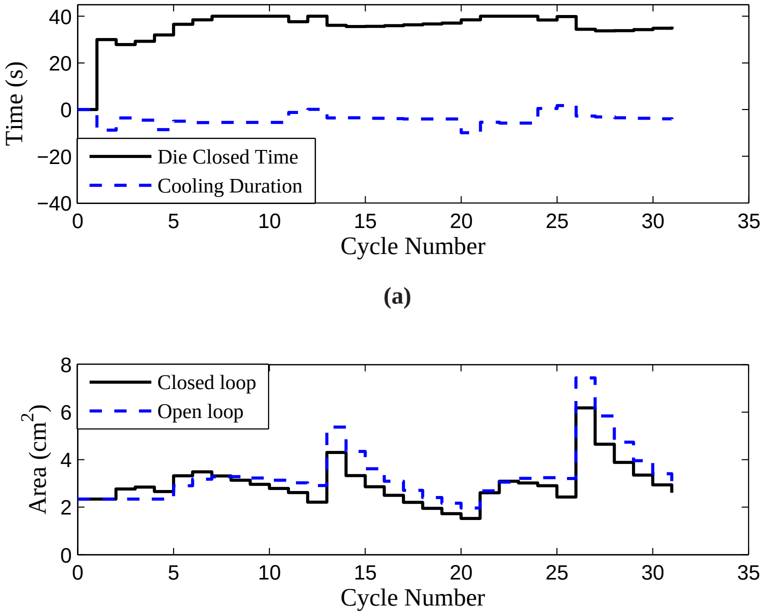
Figure 5.4: (a) The Responses of the Closed Loop Control Variables and (b) the Results for Area of Encapsulated Liquid of the Closed Loop and Open Loop When Running NMPC in MATLAB with Tuning Parameters Set of $H_u = 2$ and $H_p = 10$.



(a)

(b)

Figure 5.5: (a) The Responses of the Closed Loop Control Variables and (b) the Results for Area of Encapsulated Liquid of the Closed Loop and Open Loop When Running NMPC in MATLAB with Tuning Parameters Set of $H_u = 2$ and $H_p = 15$.



(a)

(b)

Figure 5.6: (a) The Responses of the Closed Loop Control Variables and (b) the Results for Area of Encapsulated Liquid of the Closed Loop and Open Loop When Running NMPC in MATLAB with Tuning Parameters Set of $H_u = 2$ and $H_p = 20$.

In conclusion, as shown in Table 5.1, higher values of H_p lead to better performance [72]. Also, a long prediction horizon ensures closed-loop stability for an open-loop stable system, and minimizes the risk of driving the plant into ’dead ends’ from which no feasible solution is possible [72]. Thus there are good reasons for making H_p as large as computation speed will allow and performance will improve.

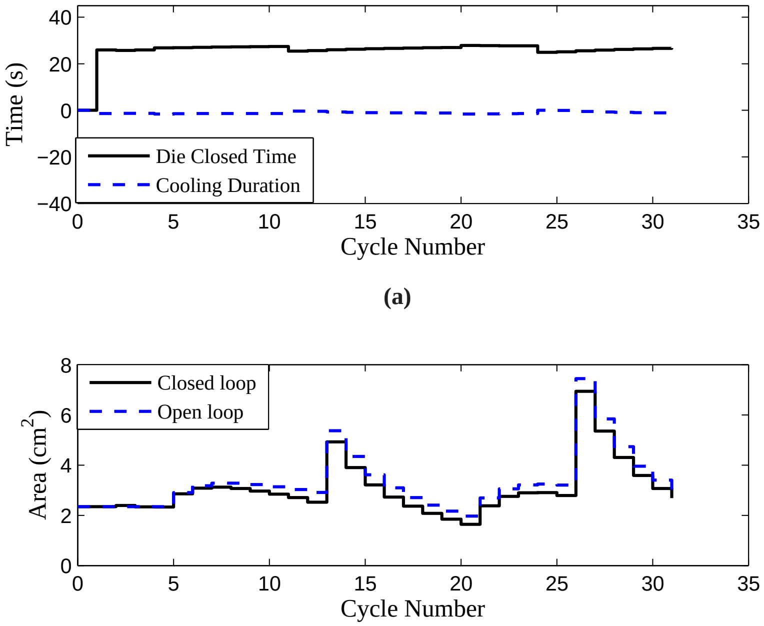
H_u Tuning

The control horizon, H_u, defines the number of changes to the control variables that the controller will calculate to drive the predicted process variables to their desired trajectories. Because the prediction horizon was chosen to be 15, the control horizon must be chosen to be in the range of $1 \leqslant H_u \leqslant 15$. As the control horizon is increased, the computed control moves tend to become more aggressive and more computation time is needed. In this research, three H_u values (1, 2, and 3) were used in the MATLAB program to the effect of H_u on the controller performance.

The responses of the closed loop control variables with different H_u settings are shown in Figures 5.7a to 5.9a, and Figures 5.7b to 5.9b present the corresponding results for area of encapsulated liquid of the closed loop and open loop operation of the process. Table 5.2 summarizes the total area of liquid encapsulation for open loop and closed loop with different H_u settings. When H_u is equal to 1 (refer to Figure 5.7 and Table 5.2), the total area of encapsulated liquid in closed loop mode has decreased by 8.5% compared to open loop. When H_u is equal to 2 (refer to Figure 5.8 and Table 5.2), the total area of encapsulated liquid in closed loop mode is decreased by 10.9%. As the control horizon is increased from 1 to 2, the freedom of the controller to perform control moves is increased and the controller is able to respond faster to the process disturbance scenario, resulting in better closed loop performance. However, when H_u is further increased to 3, shown in Figure 5.9 and Table 5.2, the closed loop performance does not improve accordingly compared to the case with H_u set to 2. This reveals that the increased freedom to make control moves resulting from increased H_u does not always bring the better closed loop performance. By simulating the closed loop performance with different values of H_u using MATLAB, a control horizon of 2 was found to perform the best and any value greater than 2 did not improve the performance. Therefore, an H_u of 2 will be used in the controller and has been applied for the remaining analysis.

Table 5.2: Summary of the Total Area of Liquid Encapsulation for Open Loop and Closed Loop with Different H_u Settings.

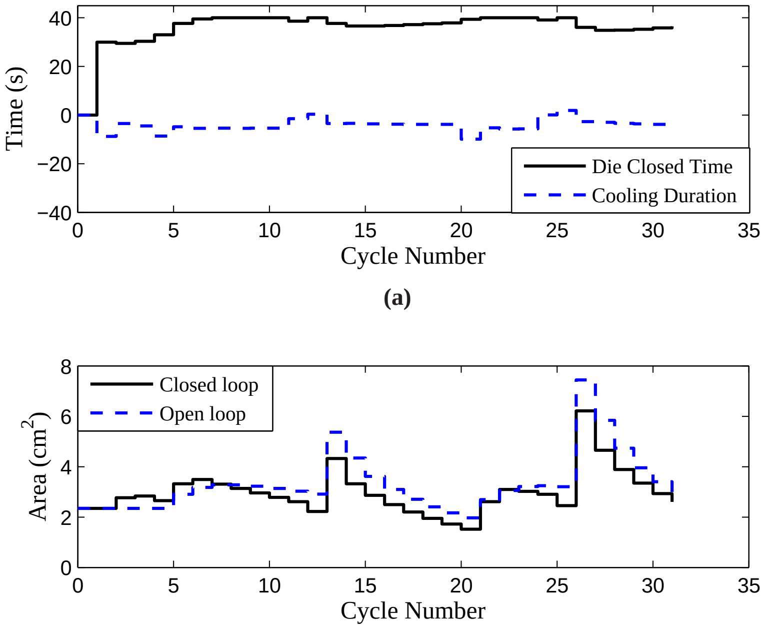
	Closed loop (H_u, H_p)			Open loop
	(1, 15)	(2, 15)	(3, 15)	
Total area of liquid encapsulation (cm^2)	95.0	92.5	92.9	103.8



(a)

(b)

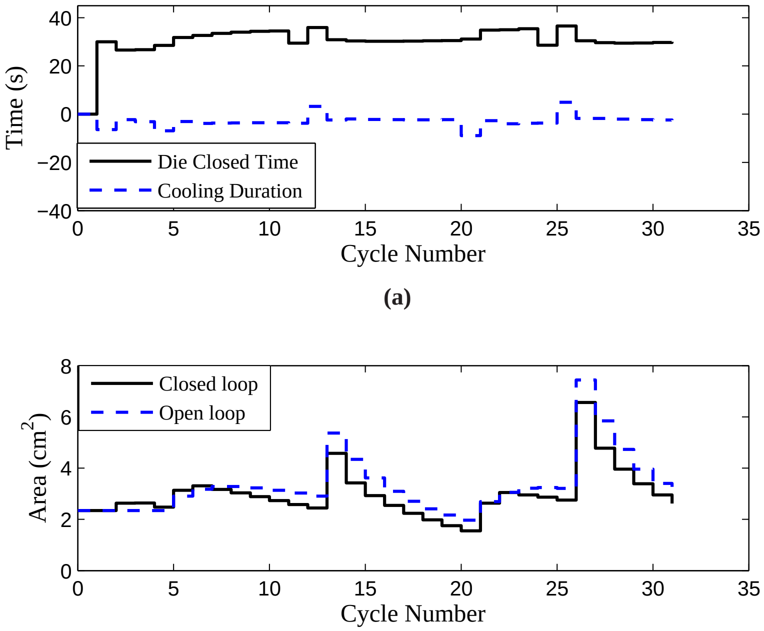
Figure 5.7: (a) The Responses of the Closed Loop Control Variables and (b) the Results for Area of Encapsulated Liquid of the Closed Loop and Open Loop When Running NMPC in MATLAB with Tuning Parameters Set of $H_u = 1$ and $H_p = 15$.



(a)

(b)

Figure 5.8: (a) The Responses of the Closed Loop Control Variables and (b) the Results for Area of Encapsulated Liquid of the Closed Loop and Open Loop When Running NMPC in MATLAB with Tuning Parameters Set of $H_u = 2$ and $H_p = 15$.



(a)

(b)

Figure 5.9: (a) The Responses of the Closed Loop Control Variables and (b) the Results for Area of Encapsulated Liquid of the Closed Loop and Open Loop When Running NMPC in MATLAB with Tuning Parameters Set of $H_u = 3$ and $H_p = 15$.

Q and R Tuning

The tuning values in the Q and R matrices are used to control the emphasis placed on the two terms in the cost function. The 1st term (refer to Equation (5.12)) considers casting quality and decreases with decreasing liquid encapsulation. The 2nd term incorporates the effects of production rate and deceases with reduced Die Closed Time.

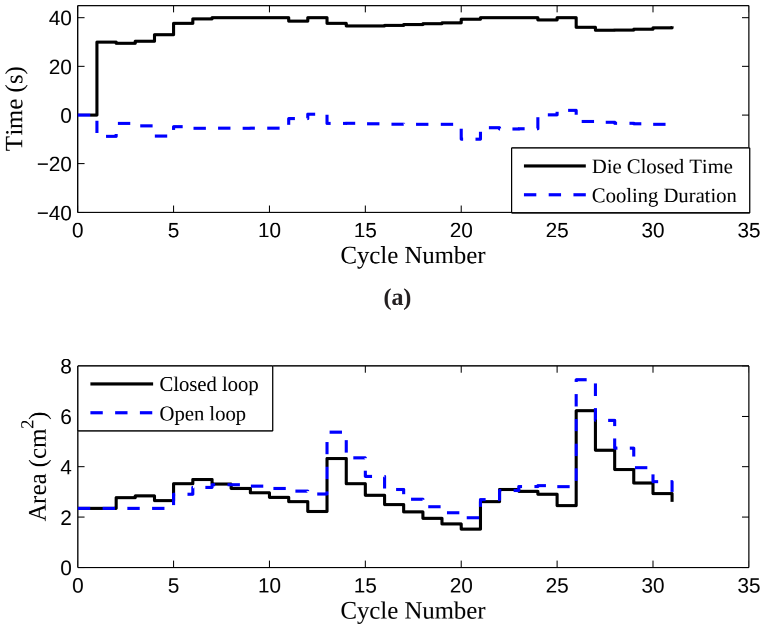
$Q(i)$ in Equation (5.12) is the penalty applied to the predicted area of liquid encapsulation in time step i. The time steps considered include from i equal to 1 to the prediction horizon H_p. $Q(i)$ is typically held constant over the prediction horizon based on the expectation that the importance of liquid encapsulation will not change along the prediction horizon. Since the value of $Q(i)$ is only important relative to the values in R, $Q(i)$ was initially set to be equal to 1.

$R(i)$ defines the penalty applied to the Die Closed Time in time step i. In the limiting case, where all $R(i)$ are assigned zero values, no penalty would be applied to the value of Die Closed Time and therefore the cost function would be minimized only based on the area of encapsulated liquid. In the other limiting case, where all $R(i)$ approach infinity, the penalty placed on Die Closed Time emphasizes production rate only. In practice, both product quality and production rate are important to industry. If the same importance is specified for both terms, the weights Q and R need to be adjusted to account for the magnitude difference in the quantities being evaluated. For example, if the predicted area of liquid encapsulation is 3 cm^2 and Die Closed Time is 150 s, the term linked with Q must be scaled by a compensation coefficient $(150 \times 150)/(3 \times 3) = 2500$ so that both terms have the same relative importance. To assess the influence of Q and R on controller performance, four cases with different cost function weightings have been simulated in MATLAB. The responses of the closed loop control variables with different Q and R settings are shown in Figures 5.10a to 5.13a, and Figures 5.10b to 5.13b present the corresponding results for area of encapsulated liquid of the closed loop and open loop operation of the process. Table 5.3 summarizes the average Die Closed Time and the total area of liquid encapsulation for open loop and closed loop with different Q and R settings.

Table 5.3: Summary of the Average Die Closed Time and the Total Area of Liquid Encapsulation for Open Loop and Closed Loop with Different Q and R Settings.

	Closed loop (Q, R)				Open loop
	(1, 0)	(2500, 1)	(3000, 1)	(0, 1)	
Average of Die Closed Time (s)	37.12	-4.2	1.1	-39.7	0
Total area of liquid encapsulation (cm^2)	92.5	104.2	101.4	184.2	103.8

In the 1st case, Q and R, equal to 1 and 0, respectively, have been selected to emphasize casting quality only at the expense of production rate. As shown in Figure 5.10 and Table 5.3, the closed loop system reduces liquid encapsulation by 11%, but the production rate is decreased as the system requires on average about 37 s longer per cycle compared to the open loop.

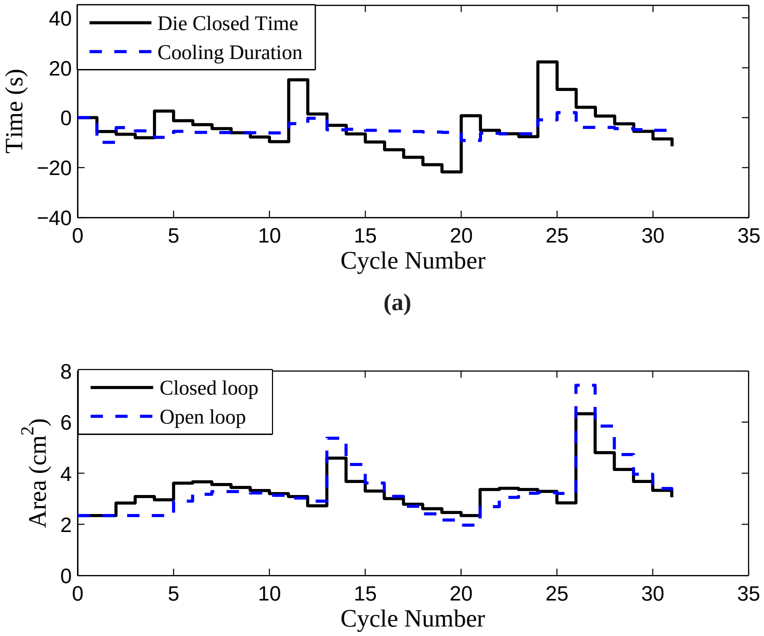


(a)

(b)

Figure 5.10: (a) The Responses of the Closed Loop Control Variables and (b) the Results for Area of Encapsulated Liquid of the Closed Loop and Open Loop When Running NMPC in MATLAB with Tuning Parameters Set of $Q = 1$ and $R = 0$.

In the 2^{nd} case, Q and R are set equal to 2500 and 1, respectively. Both terms of the cost function are nearly equally weighted with these values. The results for this case, shown in Figure 5.11 and Table 5.3, indicate that the average Die Closed Time has decreased by 4 s compared to open loop, representing a small improvement in production rate. The total area of liquid encapsulation is almost unchanged from open loop.

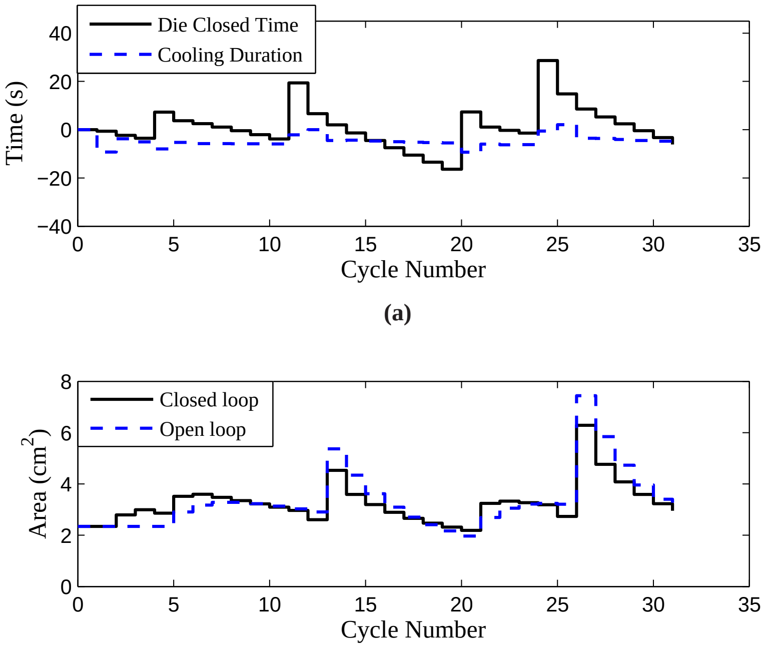


(a)

(b)

Figure 5.11: (a) The Responses of the Closed Loop Control Variables and (b) the Results for Area of Encapsulated Liquid of the Closed Loop and Open Loop When Running NMPC in MATLAB with Tuning Parameters Set of $Q = 2500$ and $R = 1$.

In the 3^{rd} case, Q and R are set equal to 3000 and 1, respectively, to change the weighting in favour of casting quality. In this instance, shown in Figure 5.12 and Table 5.3, the average Die Closed Time is 1 s compared to 0 s in the open loop which is a small reduction in production rate whereas the total area of liquid encapsulation has decreased by 2% compared to the open loop.

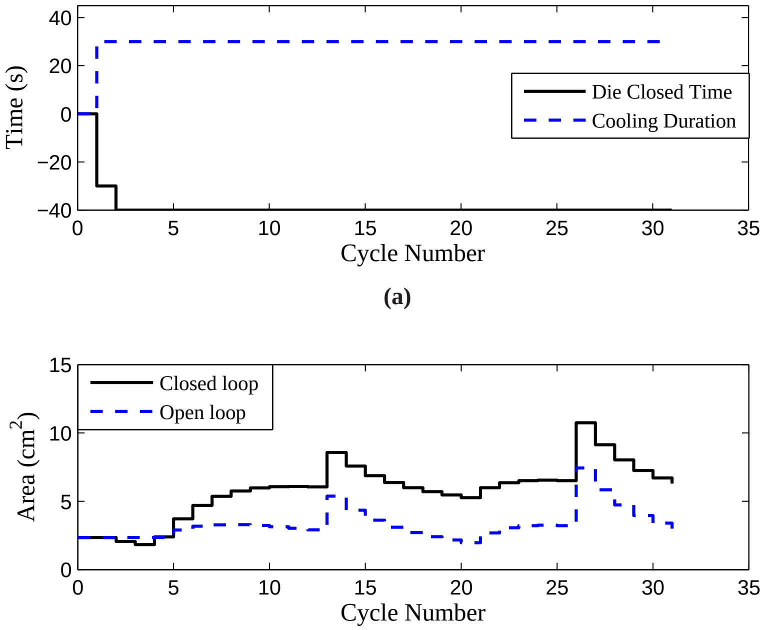


(a)

(b)

Figure 5.12: (a) The Responses of the Closed Loop Control Variables and (b) the Results for Area of Encapsulated Liquid of the Closed Loop and Open Loop When Running NMPC in MATLAB with Tuning Parameters Set of $Q = 3000$ and $R = 1$.

In the 4th case, Q and R, equal to 0 and 1, respectively, have been selected to improve production rate at the expense of casting quality. As shown in Figure 5.13 and Table 5.3, the closed loop system improves production rate through a 40 s reduction in cycle time, but the area of encapsulated liquid increases by 77% compared to the open loop.



(a)

(b)

Figure 5.13: (a) The Responses of the Closed Loop Control Variables and (b) the Results for Area of Encapsulated Liquid of the Closed Loop and Open Loop When Running NMPC in MATLAB with Tuning Parameters Set of $Q = 0$ and $R = 1$.

Q and R settings can be used to adjust the weightings of two terms in the cost function (Equation (5.12)). In practice, Q and R are adjusted based on the priorities of the manufacturer. Since casting quality is a central theme in this research, Q and R were set to be 1 and 0 for the remaining analysis.

Chapter 6

MPC Control of a Virtual Process[1]

6.1 Preparation

The ultimate goal of this research is to develop a control solution for LPDC processes that will minimize macro-porosity in castings. The MPC controller that has been developed will be applied in this chapter to control a virtual process experiencing different scenarios. The tuning parameters applied to the MPC controller, determined in the previous chapter, include the control and prediction horizons (H_u and H_p) equal to 2 and 15, respectively, and cost function weighting coefficients (Q and R) equal to 1 and 0, respectively. These settings have been selected to provide good cycle to cycle sensitivity and to minimize the area of encapsulated liquid, which is being used as an indirect indicator of macro-porosity. The virtual process used in this chapter is based on the previously described 2D-axisymmetric formulation of the casting model with boundary conditions adopted from the 3D model which was validated with plant trial data. The 2D-axisymmetric version of the FE model was selected to conduct this analysis because it exhibits process dynamics that are similar to the actual process while offering faster simulation speed.

[1] Portions of this chapter have been prepared for publication in:
• X. Shi, E. Harinath, D.M. Maijer and G. Dumont, “Nonlinear Model Predictive Control in a Low Pressure Die Casting Process”, Journal of Process Control, submitted 2012.

To complete the development of the MPC controller, it was necessary to develop a means of linking the MPC algorithm to the virtual process. The MPC algorithm was first implemented in MATLAB and then a communication protocol was developed to enable control of the virtual process. The corresponding data flow diagram is shown in Figure 6.1.

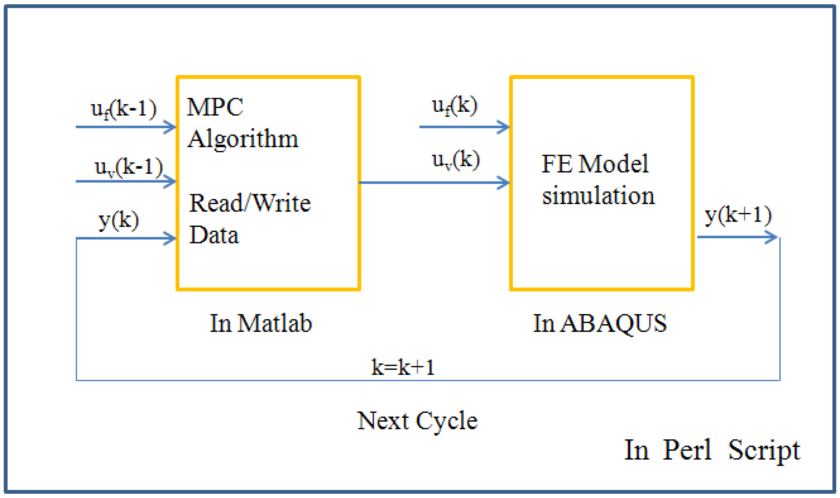


Figure 6.1: Data Flow Diagram for Virtual Process with MPC Involved.

The data flow for MPC control of the virtual process is similar to the MATLAB implementation described in the previous chapter. The only difference in the 1st stage of the controller operation is that the input/output exchange occurs via external text files used to communicate with the Perl script that operates the virtual process. Upon collecting the $u_v(k-1)$, $u_f(k-1)$ and $y(k)$ data at the current time step k, the optimal $u_v(k)$ is computed by solving the constraint optimization problem shown in Equation 5.16. In the 2nd stage of controller operation, the virtual process accepts $u_v(k)$ from the MPC algorithm and runs the FE model with updated process parameters. As a result, the new system output $y(k+1)$ is generated for use in the next cycle's calculations.

The Perl wrapper script, which enables the continuous operation of the virtual process, reads and writes data to and from various communications files at different stages in the process. When a new cycle starts, the Perl script waits for the MPC algorithm to compute the updated $u_v(k)$. Once the new data $u_v(k)$ is computed, it updates and then runs the new simulation to produce the new data $y(k+1)$. Repeatedly operating in this mode simulates the cyclic operation of the virtual process with MPC.

In this research, the nonlinear state-space model which was developed to reproduce linear dynamic behaviour does not completely reproduce the nonlinear dynamics of the virtual process throughout the operational range examined. This will affect the model-based predictive control performance on the virtual process. However, the model accuracy is greatly improved if the cycle-to-cycle rate-of-change of the control variables is limited to a reasonable range, as the nonlinear dynamics of the virtual process can be approximated with linear dynamics over smaller cycle-to-cycle operational ranges. With the aid of a small sensitivity analysis on the virtual process, the rate-of-change of the control variables was limited to $|\triangle u_{v_1}, \triangle u_{v_2}| \leq [10s, 5s]$ for the remaining analysis of this chapter.

The control solution's performance has been evaluated by considering: (1) the effect of process disturbances, (2) typical operational conditions, (3) its performance compared to Linear MPC and (4) the tolerance and response of the control solution to process uncertainties.

To evaluate the performance of the control solution in rejecting process disturbances, three disturbance scenarios have been applied to both the open loop and closed loop virtual process. The scenarios are the same as those used to test the controller tuning in Chapter 5. The first scenario is a ramping incoming Metal Temperature input disturbance, which simulates a drop and a slow ramp back in incoming Metal Temperature consistent with the addition of new metal to the holding furnace below a casting machine. The second scenario simulates a large single cycle Die Open Time input disturbance which occurs when a die is held open longer than normal while the operator performs process maintenance. The final scenario is a 'worst case' combination of the incoming Metal Temperature and Die Open Time disturbances.

The control solution has been evaluated on the virtual process during steady state and start-up operational conditions to assess its performance with the process in different operation states. The impact of controlling the process starting at two different steady state operating points will be analyzed first. The three disturbance scenarios have been applied to both the open loop and closed loop virtual process with these initial steady states. The ability of the controller to improve performance relative to the open loop case for start-up conditions will then be assessed using both the worst case disturbance scenario and no disturbances.

The differences between Linear and Non-linear MPC have been assessed using the virtual process. The worst case disturbance scenario with two different initial steady states will be used for this assessment. The results from both controllers will be used to demonstrate the advantage of applying Non-linear MPC over Linear MPC on the virtual process.

Finally, the robustness, or stability, of the control solution while experiencing typical process uncertainties that affect heat transfer in the die will be explored. The two parameters considered in this analysis were the heat transfer coefficients applied along the casting/die interface (h_1) and inside Cooling Channel 1(h_2). The heat transfer conditions at these locations are known to vary during casting campaigns in the industrial process [3]. In the industrial process, the coating that protects the die surface from erosion/corrosion by the liquid metal degrades at different locations. These areas are intermittently re-coated by the operator as part of the cycle-to-cycle checks that are performed during Die Open Time. Thus, the cyclic variation of the coating thickness is a source of uncertainty for the interfacial heat transfer. Likewise, the cooling intensity in the Cooling Channels varies cycle-to-cycle with changes in the water pressure and flow rate. A sensitivity analysis will be conducted with h_1 and h_2 to assess the impact of changes in these parameters on the process performance during closed and open loop.

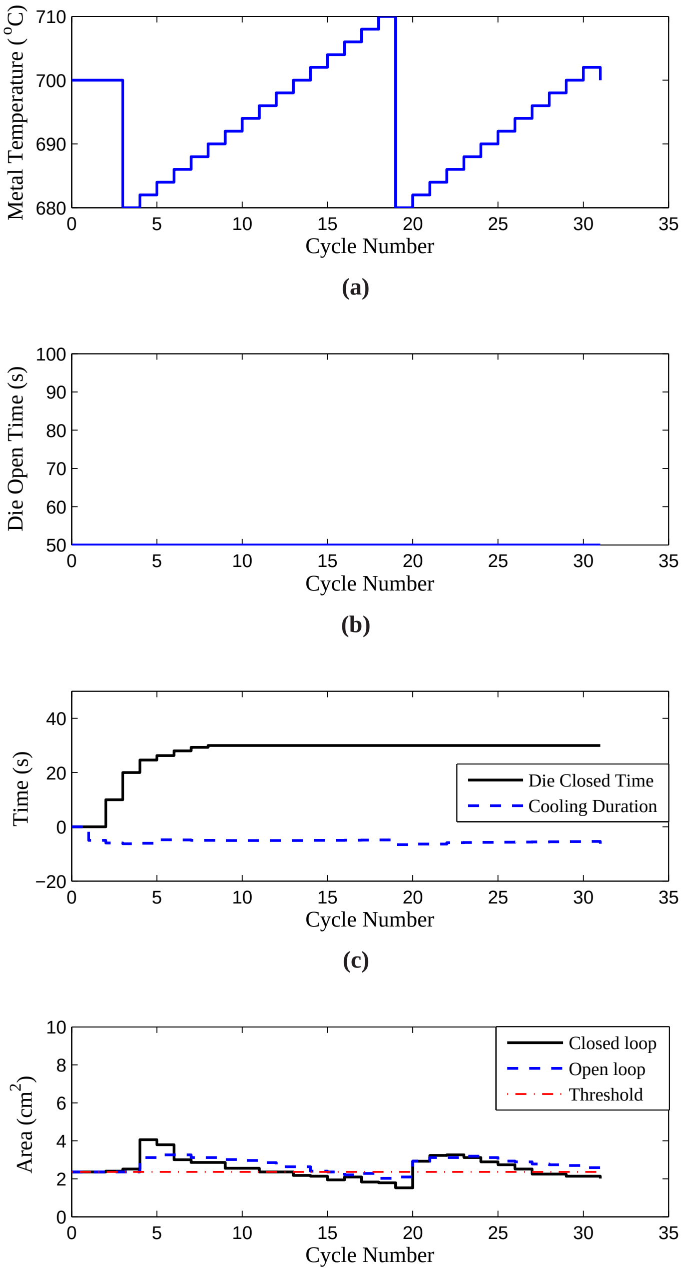
6.2 NMPC of Steady State Operation

To assess the performance of the NMPC in controlling the virtual process starting from steady state operation, two steady state operating points of Die Closed Time and Cooling Duration equal to (150 s, 10 s) and (130 s, 30 s) were selected. Simulations of open loop and closed loop operation were performed for both settings. The (150 s, 10 s) state was selected because it is a near optimal operational condition for the virtual process while (130 s, 30 s) state was selected because it is a condition that is away from the optimal operational region.

Each of the following subsections presents four graphs that were generated by applying the previously described disturbance scenarios. The difference between the area predictions for open and closed-loop operation of the virtual process are used to assess the performance of the control solution. The figures in each subsection are composed of graphs (a) to (d), where (a) and (b) are two disturbance variables in the scenario, the graph in (c) displays the changes made to the control variables relative to the baseline operational condition of (150s, 10s) for the closed loop and the graph in (d) compares the closed loop and open loop areas of encapsulated liquid which occurs during each scenario. The red dashdotted line in (d), which is defined as the area prediction for the initial steady state operation, represents the threshold value for the area of encapsulated liquid in the casting. This threshold represent the maximum acceptable area of encapsulated liquid for a casting, i.e. if the area of encapsulated liquid is below this threshold, the casting will be accepted, otherwise this casting is rejected. The ’good’ casting ratio equal to the number of acceptable castings over the total number simulated is then reported. The total area of encapsulated liquid for both open and closed-loop operation is also presented to aid in controller performance assessment.

6.2.1 Steady State Operation (150s, 10s)

A series of simulations were executed to assess the performance of the NMPC for different disturbance scenarios during steady state operation based on a Die Closed Time of 150s and Cooling Duration of 10s. Figures 6.2 to 6.4 show the results for the different disturbance scenarios. Table 6.1 summarizes the results that compare the open loop and closed loop simulations starting from steady state operation at (150 s, 10 s) for the different disturbance scenarios.



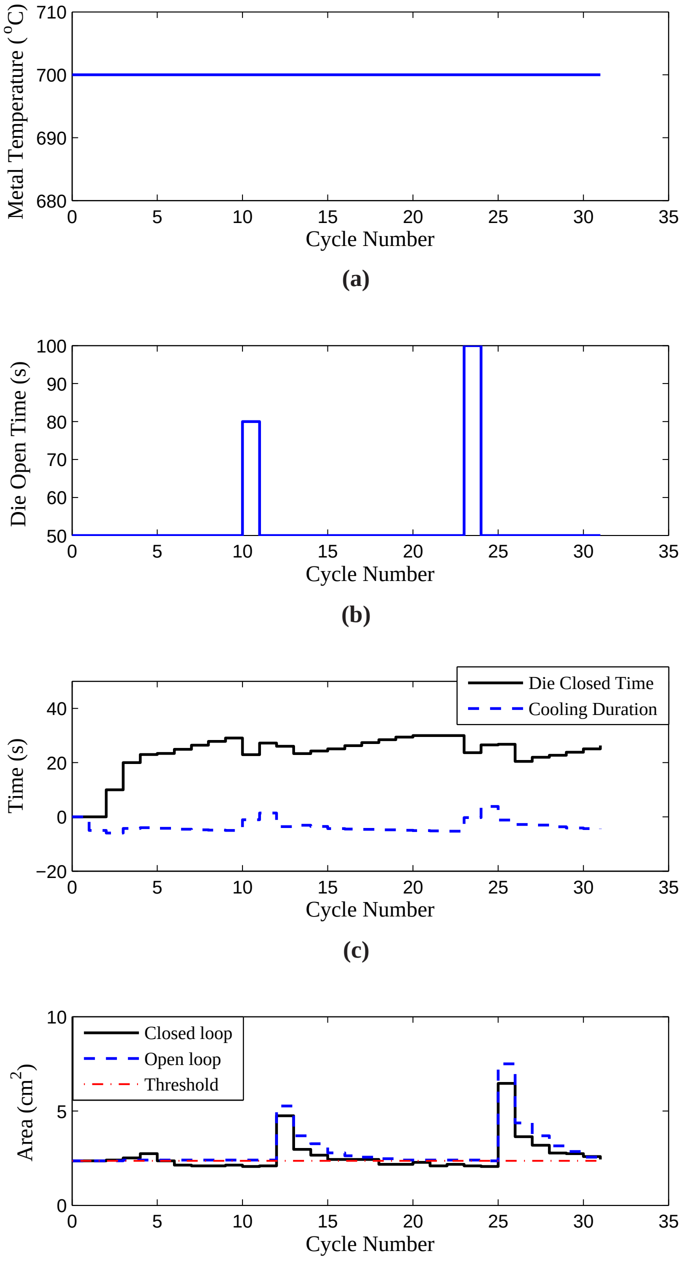
(a)

(b)

(c)

(d)

Figure 6.2: (a)-(b)Virtual Process Metal Temperature Disturbance Scenario and (c) the responses of the Closed Loop Control Variables and (d) the results for Area of Encapsulated Liquid of the Closed Loop and Open Loop with Steady State Operation (150s, 10s).



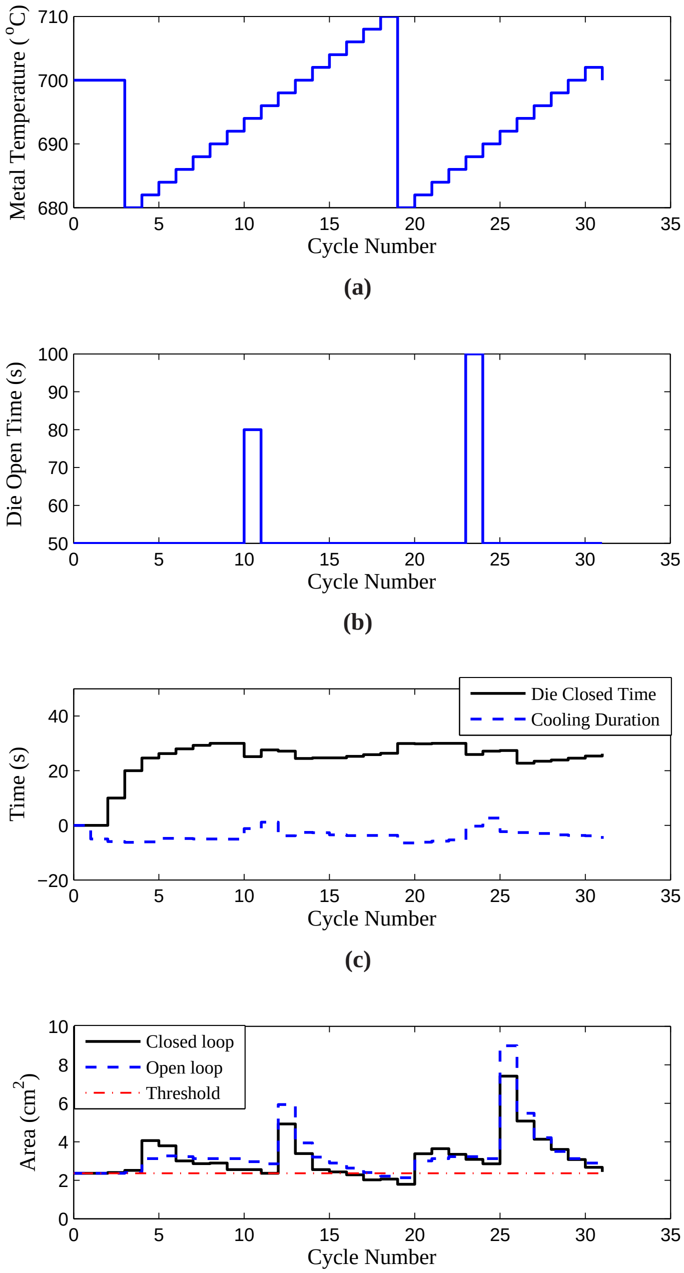
(a)

(b)

(c)

(d)

Figure 6.3: a)-(b)Virtual Process Die Open Time Disturbance Scenario and (c) the responses of the Closed Loop Control Variables and (d) the results for Area of Encapsulated Liquid of the Closed Loop and Open Loop with Steady State Operation (150s, 10s).



(a)

(b)

(c)

(d)

Figure 6.4: (a)-(b)Virtual Process Worst Case Disturbance Scenario and (c) the responses of the Closed Loop Control Variables and (d) the results for Area of Encapsulated Liquid of the Closed Loop and Open Loop with Steady State Operation (150s, 10s).

Table 6.1: Comparison of Sum of Area and Good Casting Ratio between Open Loop and Closed Loop Simulations with initial Steady State Operation (150s, 10s) under Different Disturbance Scenarios.

Disturbance	Sum of Area (cm^2)			Good Casting Ratio (%)	
Scenario	Open loop	Closed loop	Reduction rate (%)	Open loop	Closed loop
Metal Temperature	84.9	78.6	7.5	25.8	48.4
Die Open Time	90.0	81.6	9.4	9.7	48.4
Worst Case	103.7	97.5	7.0	12.9	19.4

Figure 6.2d shows the closed loop and open loop response of the virtual process to the ramping incoming Metal Temperature disturbance. The control variables during closed loop operation, shown in Figure 6.2c, keep varying to mitigate the negative effect from the disturbance. When the drops in incoming Metal Temperature occur in the 3rd and 19th cycles, the area of encapsulated liquid increases in both the open and closed loop modes. For open loop operation, it takes about 11 cycles to return to an area that is below the threshold, while only 8 cycles are required in closed loop mode. To achieve this response, the Die Closed Time steadily increases initially until it saturates at +40 s over the baseline (150 s) and the Cooling Duration fluctuates between 0 s and -5 s over the baseline (10 s) as the incoming Metal Temperature changes relative to the nominal value of 700 °C. The performance improvement of the closed loop over the open loop is clearly seen in Table 6.1, where the sum of area of encapsulated liquid decreased from 84.9 cm^2 in the open loop to 78.6 cm^2 in the closed loop, resulting in a 7.5 % reduction; and the good casting ratio increased from 25.8 % in the open loop to 48.4 % in the closed loop. The reduced number of cycles necessary to return the area of encapsulated liquid to within the accepted standard highlights the improved capabilities of the NMPC to reject the incoming Metal Temperature disturbance.

In the case of a Die Open Time disturbance, the results presented in Figure 6.3d highlight the faster response of the closed loop process to drop the area within the accepted standard. In this case, the Die Closed Time in Figure 6.3c does not saturate as in the last case, but rather varies both before and after the Die Open Time disturbances and the Cooling Duration exhibits a larger range of variation of

-5 s to 5 s. The sum of area of encapsulated liquid decreases by 9.4 % in the closed loop process with a value of 81.6 cm^2 compared to the open loop case of 90.0 cm^2. The good casting ratio increases from 9.7 % in open loop to 48.4 % in closed loop, as shown in Table 6.1.

For the worst case scenario, where both the incoming Metal Temperature and the Die Open Time disturbances are concurrent, the closed loop dynamic response of the virtual process is still better than the open loop response, as shown in Figure 6.4. The Die Closed Time and Cooling Duration exhibit more variations over the sudden drop of Metal Temperature disturbance and sudden impulse of the Die Open Time disturbance. The total area of encapsulated liquid is reduced 7.0 % from 103.7 cm^2 in the open loop to 97.5 cm^2 in the closed loop, and good casting ratio goes up from 12.9 % in open loop to 19.4 % in the closed loop (refer to Table 6.1). Note that worst case disturbance scenario is not the worst for good casting ratio. The reason is when the process is in the open loop under die open time disturbance scenario, the area at many cycles, shown in Figure 6.3c, is narrowly above the threshold, hence decreasing good casting ratio to the minimum.

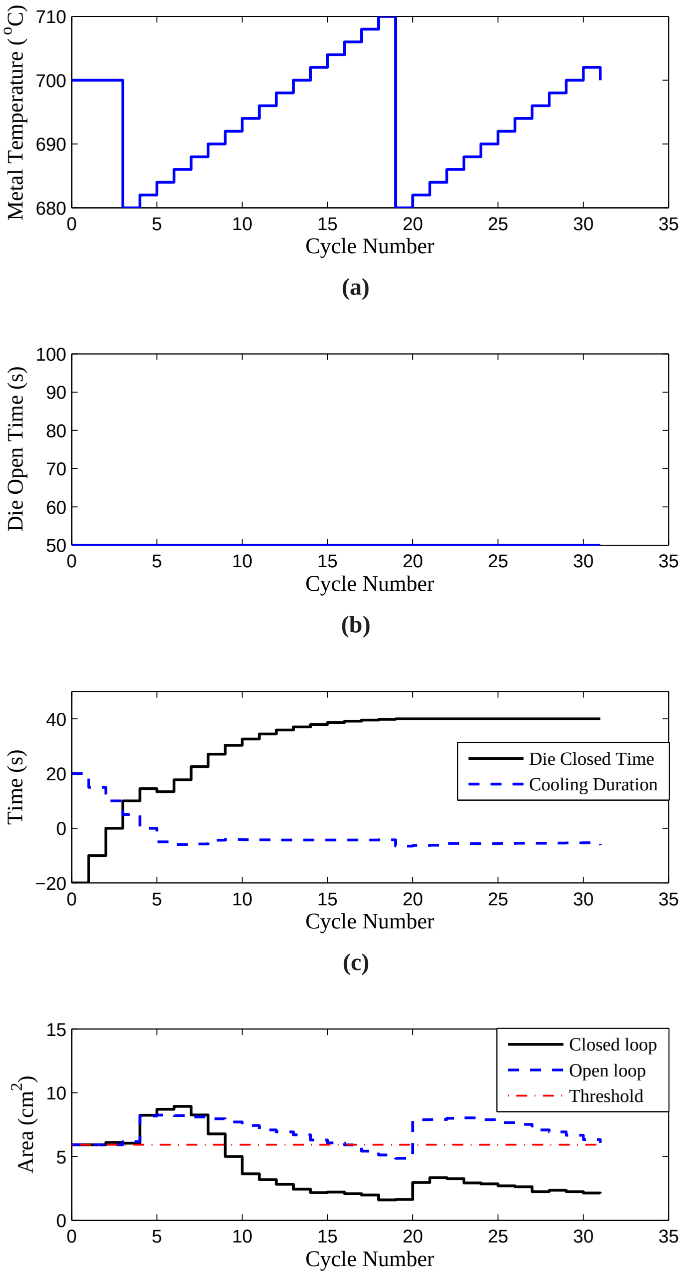
Overall, the performance of the virtual process during closed loop operation is better than in the open loop for all three disturbance scenarios when started from steady state operation (150s, 10s). In terms of the two evaluation metrics, the closed loop process exhibits a lower total area and a higher good casting ratio than the open loop process (refer to Table 6.1). It should be noted that even though the analysis was started from steady state, based on a Die Closed Time of 150s and the Cooling Duration of 10s, which is close to the static optimal state, total area of encapsulated liquid was reduced by 6% – 10% in the three disturbance scenarios. If the starting point is set far away from the optimal steady state, a higher reduction rate is expected, as will be shown in the following subsection.

6.2.2 Steady State Operation (130s, 30s)

The performance of the NMPC in rejecting the three disturbance scenarios when starting from a steady state point that is far from the optimal point was assessed in a similar manner to the previous section. In this case, the starting steady state condition was based on a Die Closed Time equal to 130s and a Cooling Duration equal to 30s. Figures 6.5 to 6.7 present the results for the different disturbance scenarios and Table 6.2 summarizes the sum of area of encapsulated liquid and good casting ratio between open loop and closed loop simulations. The relative values of the starting point are (-20 s, 20 s) compared to the baseline condition of (150 s, 10 s). Thus, the initial values of the control variables in Figures 6.5c to 6.7c start with (-20 s, 20 s).

Table 6.2: Comparison of Sum of Area and Good Casting Ratio between Open Loop and Closed Loop Simulations with initial Steady State Operation (130s, 30s) under Different Disturbance Scenarios.

Disturbance	Sum of Area(cm^2)			Good Casting Ratio(%)	
Scenario	Open loop	Closed loop	Reduction rate (%)	Open loop	Closed loop
Metal Temperature	216.2	119.6	45.0	19.4	77.4
Die Open Time	210.4	113.3	46.1	6.5	77.4
Worst Case	239.6	137.8	42.5	16.1	74.2



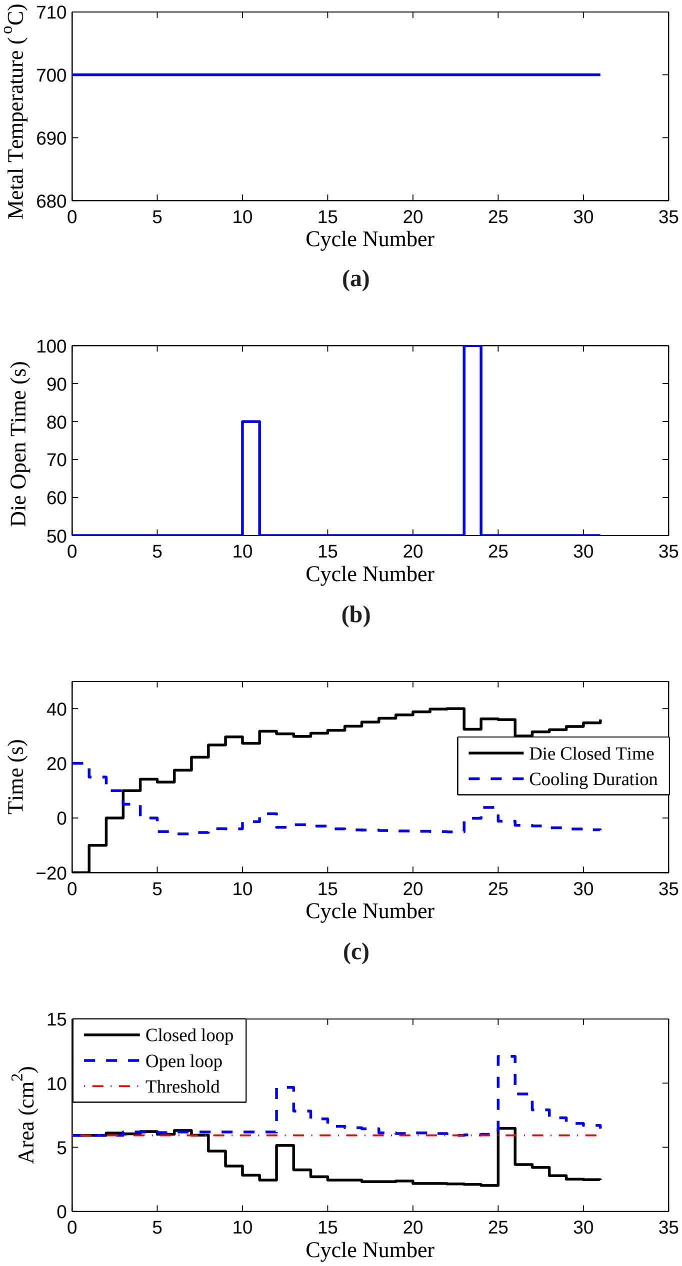
(a)

(b)

(c)

(d)

Figure 6.5: (a)-(b)Virtual Process Metal Temperature Disturbance Scenario and (c) the responses of the Closed Loop Control Variables and (d) the results for Area of Encapsulated Liquid of the Closed Loop and Open Loop with Steady State Operation (130s, 30s).



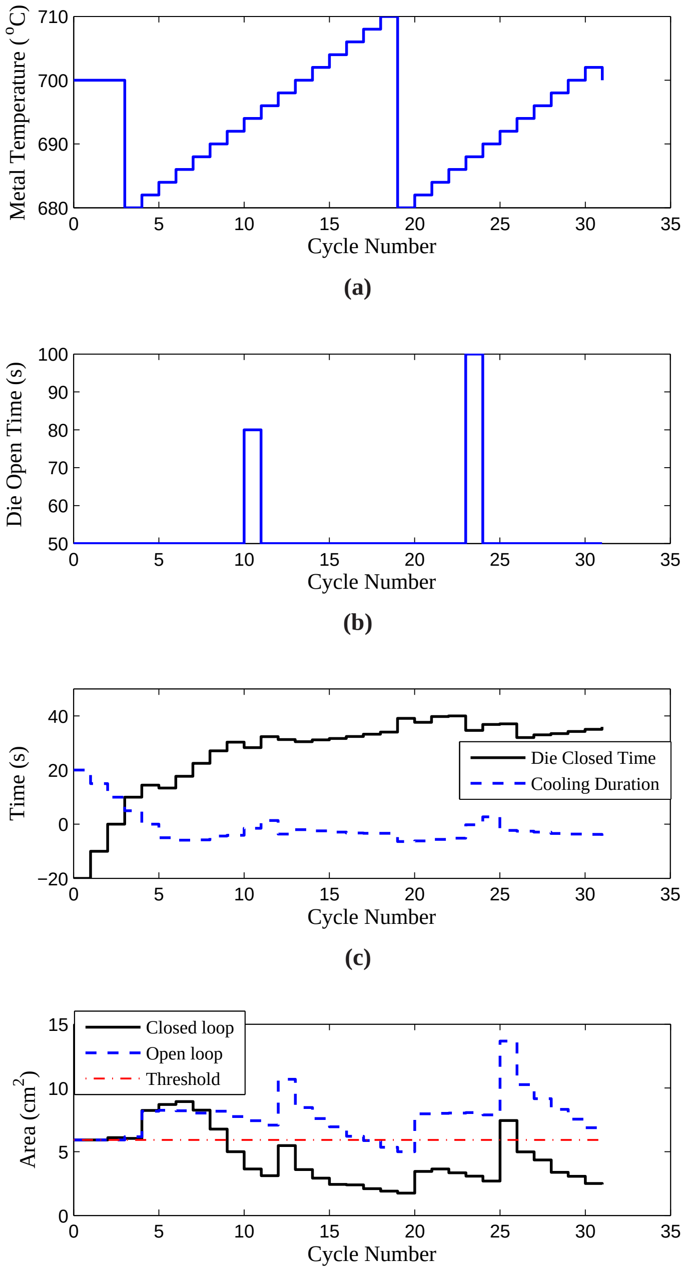
(a)

(b)

(c)

(d)

Figure 6.6: (a)-(b)Virtual Process Die Open Time Disturbance Scenario and (c) the responses of the Closed Loop Control Variables and (d) the results for Area of Encapsulated Liquid of the Closed Loop and Open Loop with Steady State Operation (130s, 30s).



(a)

(b)

(c)

(d)

Figure 6.7: (a)-(b)Virtual Process Worst Case Disturbance Scenario and (c) the responses of the Closed Loop Control Variables and (d) the results for Area of Encapsulated Liquid of the Closed Loop and Open Loop with Steady State Operation (130s, 30s).

The results presented in Figure 6.5 show the closed loop and open loop response of the virtual process during the ramping incoming Metal Temperature disturbance. The control variables change rapidly as the move toward the optimal operating point in the process at the beginning of the scenario, when the Cooling Duration reduces gradually to approximately -5s where it begins to change as a function of the disturbance. The closed loop process responds much faster to incoming Metal Temperature disturbance than the open loop process, as shown in Figure 6.5d where the closed loop only takes 6 cycles (3^{rd} to 9^{th}) to respond to the the 1^{st} drop in incoming Metal Temperature by driving the area of encapsulated liquid below the threshold and the open loop consumed 14 cycles (3^{rd} to 17^{th}). The performance improvement of closed loop operation over open loop is summarized in Table 6.2. The NMPC controller decreases the total area from 216.2 cm^2 in the open loop case to 119.6 cm^2 in the closed loop case i.e. a 45.0% reduction; and the good casting ratio increases from 19.4 % in the open loop case to 77.4 % in the closed loop case. The good casting ratio is significantly higher for the closed loop operation because the reference area of encapsulated liquid, which defines the threshold, is larger for the non-optimal steady state starting condition.

The results of for the Die Open Time disturbance scenario case are presented in Figure 6.6. In this case, the changes in the control variables cause the area of encapsulated liquid to decrease below the threshold during closed loop operation after the 9^{th} cycle and remain their except for one cycle after the 2^{nd} impulse in Die Open Time. This superior performance compared to the open loop operation is further demonstrated in Table 6.2, which shows that the total area decreases by 46.1% in the closed loop process and the good casting ratio increased from 6.5 % in the open loop to 77.4 % in the closed loop.

In the worst case disturbance scenario, the closed loop dynamic response of the area of encapsulated liquid continues to be better than the open loop response (refer to Figure 6.7d). The area of encapsulated liquid is reduced by approximately 42.5% in closed loop compared to open loop, and the good casting ratio rises from 16.1 % in the open loop to 74.2 % in the closed loop (refer to Table 6.2).

6.2.3 Discussion and Conclusion

As shown in Tables 6.1 and 6.2, the MPC controller is capable to rejecting typical disturbances starting from a variety of steady state conditions. The total area of encapsulated liquid was reduced between $7-40\%$ compared to the open loop operation. Also, the good casting ratio increased in the closed loop compared to the open loop. The level of reduction in the area of encapsulated liquid and the increase in the good casting ratio are affected by the choice of the steady state operating point from which the closed loop and open loop process starts. Further away from the optimal operational condition for the chosen starting steady state leads to the more improvement in the closed loop performance. This is mainly caused by the poorer open loop response that results from the suboptimal steady state condition relative to the performance of the closed loop process which gets better as it moves toward the optimal operation condition. Considering that the area of encapsulated liquid is reduced and good casting ratio is increased under all disturbance scenarios for both starting steady state operation conditions, indicates that the closed loop performance is substantially better than open loop performance under any disturbance scenario when the process starts from steady state.

It is important to note that regardless of the steady state condition that the analysis was started from, the control variables are driven to the optimal operational condition for the virtual process. To highlight this capability, two additional cases were considered with the NMPC closed loop virtual process starting with steady state operating points of (140 s, 0 s) and (160 s, 30 s). Combined with the initial cases, the four worst case scenario results from four data with different initial steady state were analyzed to assess the evolution of the control variables, as shown in Figure 6.8. The traces of the control variables for the four cases are nearly overlapped during the last stages of the scenario (from about 20^{th} cycle to the end). This response illustrates two of the inherent benefits of applying NMPC. One is the automated search for the optimal operational condition by controller during both dynamic or static process operation, the other is that the effects from different starting steady state on the traces searching for the optimal operational condition decay over with increasing cycles.

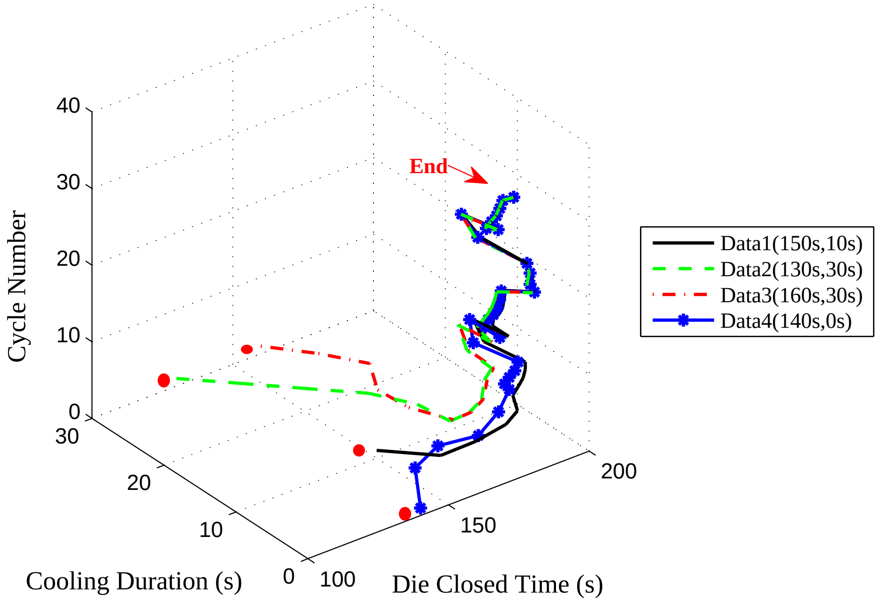


Figure 6.8: NMPC Control Variables' Traces with Different Initial Steady States on the Virtual Process under Worst Case Disturbance Scenario.

6.3 NMPC During Process Start-Up

The casting process start-up is a significant transient event which occurs during each campaign. Following a preheating operation where the die may be heated in an oven or with burners, casting is started often with non-standard operational parameters including longer die closed times and little, or no, cooling for preset number of cycles. The ability of a casting process to quickly approach steady state conditions or to begin producing acceptable castings is important. The use of NMPC to vary the process conditions during the start-up period may reduce its duration and result in more good castings. To assess the performance of the NMPC controller during start-up, two preset operating points of (160 s, 0 s) and (130 s, 30 s) were selected to apply initially during the start-up. In these cases, the virtual process was started with a uniform initial die temperature of 400°C in the first cycle. In order to generate the required process data to start the NMPC controller, the first two cycles were run with the preset operational conditions prior to enabling the controller. In this analysis, either no disturbance scenario or the worst case disturbance scenario were applied to the closed loop and open loop virtual process. Since the virtual process has not reached steady state when the controller is activated, the initial value of the state vector x_0 in the Equation (4.9) is difficult to predict. For the cases presented here, the initial values of x_0 have been set $x_0 = 0$ with the expectation that the controller will determine the optimal condition regardless of the starting point.

The figures in each of the following subsections have the same structures as those presented in the previous section 6.2. Each figure contains subfigures (a) to (d). Figure (a) and (b) displays the two disturbance variables applied to the virtual process respectively. Figure (c) displays the changes to the control variables in the closed loop with the relative values compared to (150 s, 10 s), and Figure (d) compares the area of liquid encapsulation during closed loop and open loop operation.

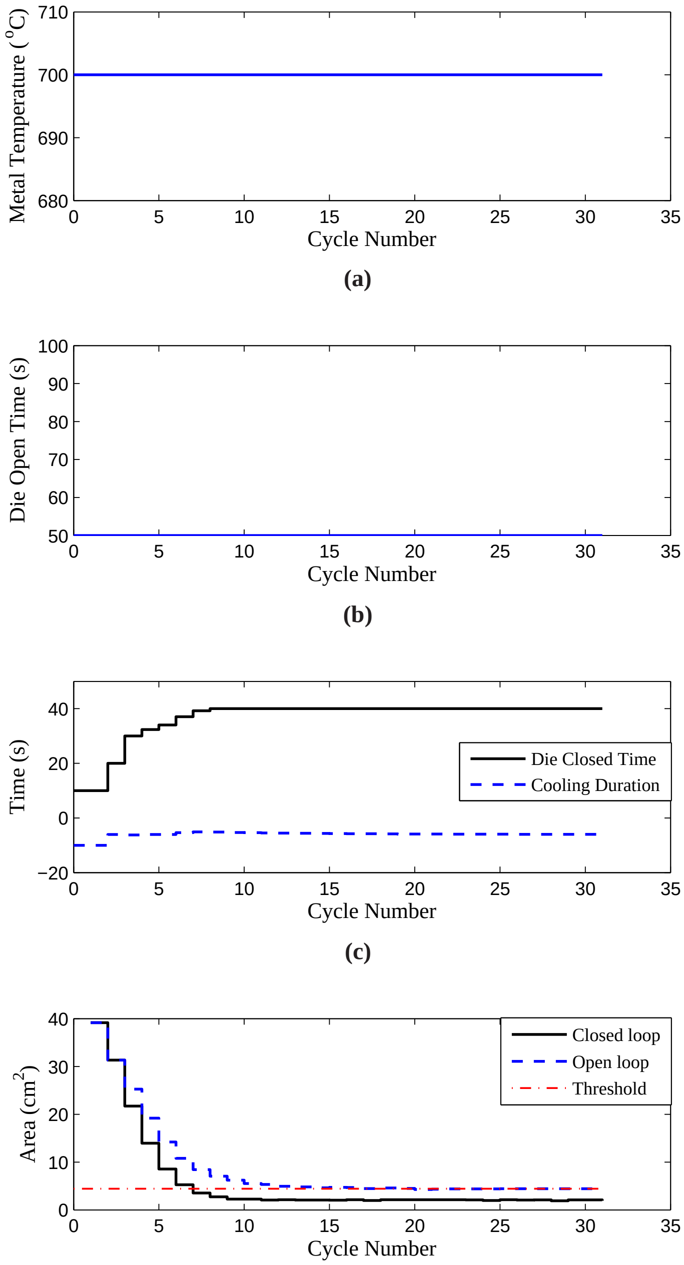
6.3.1 Start-up With Initial Operating Point of (160 s, 0 s)

The performance of the controller during start-up with either no disturbance or worst case disturbance scenario was assessed with the virtual process for the closed loop and open loop operation. The preset operating point of Die Closed Time equal to 160 s and Cooling Duration equal to 0 s was used for these scenarios. This operational condition is represents (10 s, -10 s) relative to the baseline operational condition of (150 s, 10 s). Figures 6.9 and 6.10 show the results for no disturbance and worst case disturbance scenarios, respectively. Table 6.3 summarizes the total area of encapsulated liquid and the good casting ratio for open and closed loop start-up of the virtual process for both disturbance scenarios.

Figure 6.9 shows the closed loop and open loop performance of the virtual process without disturbances. The closed loop process takes about 6 cycles to reach the area threshold and 9 cycles to reach the optimal steady state operational condition. For this analysis, the area threshold has been set equal to the area of encapsulated liquid that occurs during baseline operation (150 s, 10 s). The open loop process requires 14 cycles to reach steady state operation and the area threshold. The faster response of the closed loop process relative to the open loop process results in a reduced total area of encapsulated liquid (33.6 %) and an increased good casting ratio of 80.7 % (refer to Table 6.3). The closed loop process also exhibits similar improved performance relative to the open loop process for the worst case disturbance scenario, as shown in Figure 6.10. In this case, the closed loop response requires 8 cycles to reach the area threshold compared to 20 cycles for the open loop to reach the threshold. Unlike the no disturbance case, the control variables vary from cycle-to-cylce during the worst case disturbance scenario.

Table 6.3: Comparison of Sum of Area and Good Casting Ratio between Open Loop and Closed Loop Simulations with Start-up Operation (160s, 0s) under Both no Disturbance and worst case Disturbance Scenario.

Disturbance	Sum of Area (cm^2)			Good Casting Ratio (%)	
Scenario	Open loop	Closed loop	Reduction rate (%)	Open loop	Closed loop
No Disturbance	262.4	174.2	33.6	38.7	80.7
Worst Case	312.7	210.8	32.6	9.7	67.7



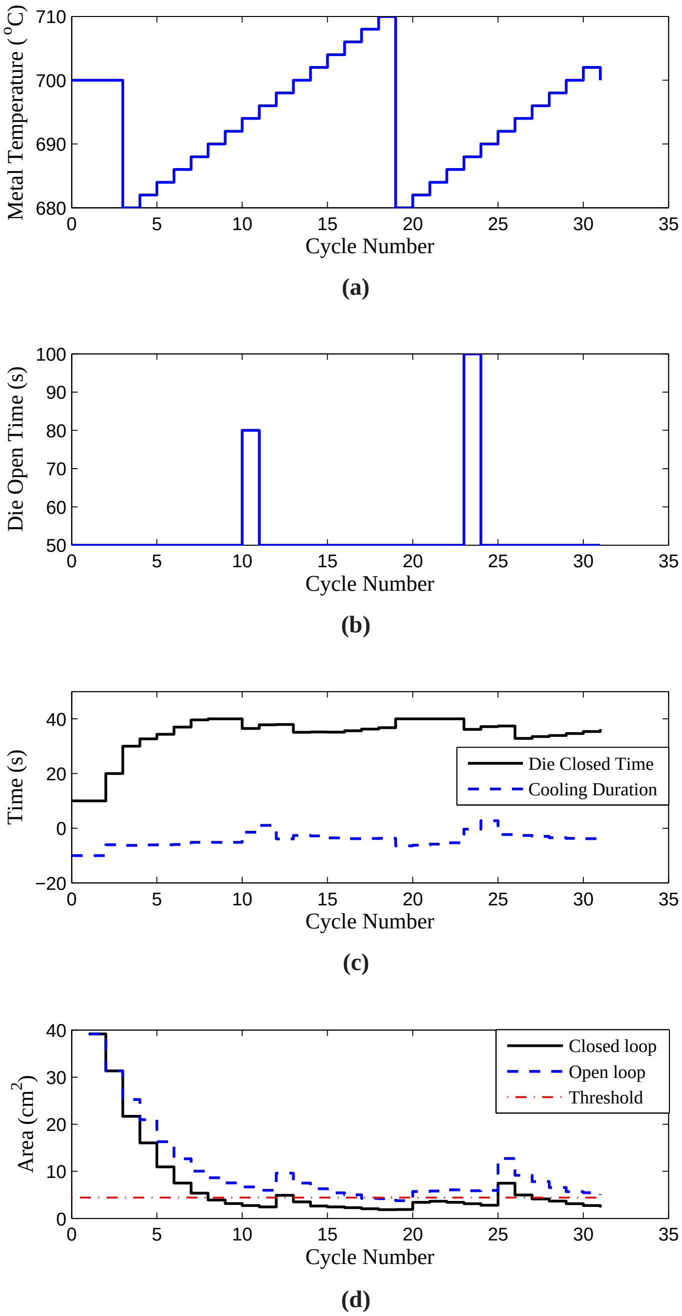
(a)

(b)

(c)

(d)

Figure 6.9: (a)-(b)Virtual Process No Disturbance Scenario and (c) the responses of the Closed Loop Control Variables and (d) the results for Area of Encapsulated Liquid of the Closed Loop and Open Loop with Start-up Operation (160s, 0s).



(a)

(b)

(c)

(d)

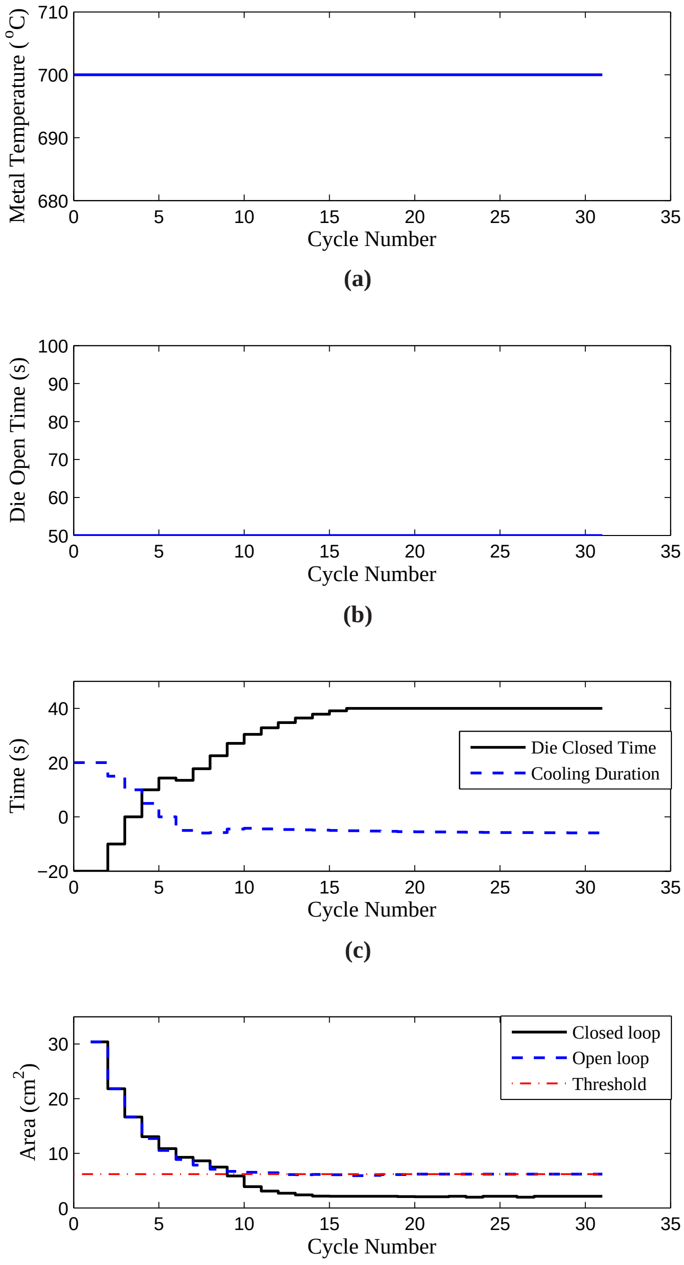
Figure 6.10: (a)-(b)Virtual Process Worst Case Disturbance Scenario and (c) the responses of the Closed Loop Control Variables and (d) the results for Area of Encapsulated Liquid of the Closed Loop and Open Loop with Start-up Operation (160s, 0s).

6.3.2 Start-up With Initial Operating Point of (130 s, 30 s)

The performance of the controller during start-up for a different initial operational condition (130 s, 30 s) was assessed with no disturbances and the worst case disturbance scenario. This initial operational condition is represented as (-20 s, 20 s) relative to the baseline of (150 s,10 s). Figures 6.11 and 6.12 show the results for no disturbance and the worst case disturbance scenario, respectively. Table 6.4 summarizes the total area of encapsulated liquid and the good casting ratio between open loop and closed loop simulations with a start-up condition of (130 s, 30 s) under both no disturbance and worst case disturbance scenarios.

Table 6.4: Comparison of Sum of Area and Good Casting Ratio between Open Loop and Closed Loop Simulations with Start-up Operation (130s, 30s) under Both no Disturbance and Worst Case Disturbance Scenario.

Disturbance Scenario	Sum of Area (cm^2)			Good Casting Ratio (%)	
	Open loop	Closed loop	Reduction rate (%)	Open loop	Closed loop
No Disturbance	258.2	174.0	32.6	64.5	74.2
Worst Case	310.2	214.0	30.7	0	58.1



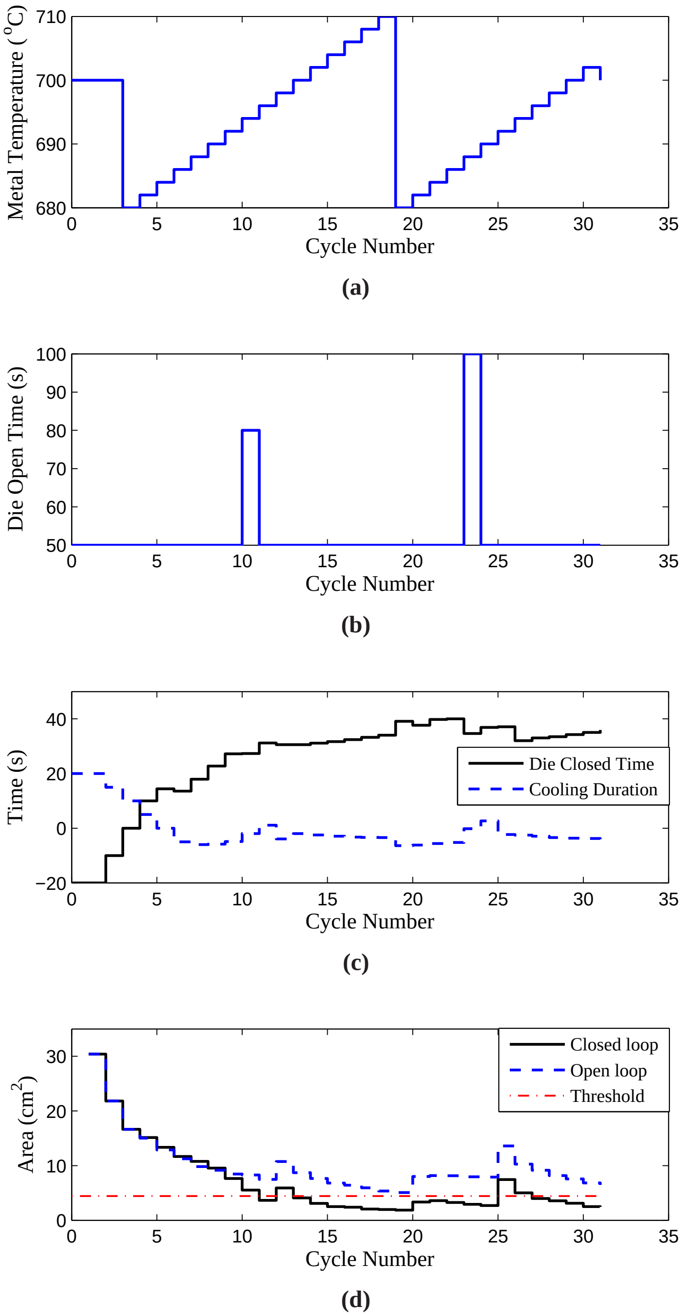
(a)

(b)

(c)

(d)

Figure 6.11: (a)-(b)Virtual Process No Disturbance Scenario and (c) the responses of the Closed Loop Control Variables and (d) the results for Area of Encapsulated Liquid of the Closed Loop and Open Loop with Start-up Operation (130s, 30s).



(a)

(b)

(c)

(d)

Figure 6.12: (a)-(b)Virtual Process Worst Case Disturbance Scenario and (c) the responses of the Closed Loop Control Variables and (d) the results for Area of Encapsulated Liquid of the Closed Loop and Open Loop with Start-up Operation (130s, 30s).

The results of the closed loop and open loop assessment of the virtual process performance without disturbances and starting with initial conditions of (130s,30s) is shown in Figure 6.11. Compared to the 12 cycles required by the open loop process, the closed loop process spends 9 cycles from start-up to reach the threshold. This improvement is relatively small compared to the previous starting condition examined. This trend is also observed in the good casting ratio, summarized in Table 6.4, where the good casting ratio is 64 % and 74 % for open and closed loop operation, respectively. When considering start-up and the worst case disturbance, the closed loop process shows much better performance than the open loop process, as shown in Figure 6.12. In this case, it take 11 cycles during closed loop operation to reach the threshold from start-up, whereas during open loop operation, the area of encapsulated liquid never falls below the threshold during the start-up / disturbance scenario. For the worst case disturbance scenario, the total area of encapsulated liquid is reduced by 30.7 % and the good casting ratio is 58.6 % for the closed loop and 0 for the open loop.

6.3.3 Discussion and Conclusion

The two start-up conditions examined show similar behaviour in terms of the benefits of closed loop operation for total area reduction. For both the no disturbance and the worst case disturbance scenarios, the total area of encapsulated liquid is reduced about 30% for the closed loop operation and the good casting ratio is always larger for the closed loop operation. Thus, based on these metrics, operating the virtual process with NMPC leads to better performance compared to the open loop operation during start-up while experiencing typical disturbances.

When no disturbances are applied, the control variables of the virtual process are gradually driven by the NMPC toward the optimal steady state operating point. When the worst case disturbance scenario is applied, the control variables continue to varying each cycle, and the optimal steady state operating point is not achieved. In all start-up cases examined, the NMPC improves the operation of the virtual process resulting in a reduced number of cycles to reach the area threshold compared to the open loop.

The uncertainty in setting x_0 in the Equation (4.9) did not affect the NMPC performance partly because its effect on the virtual process has been compensated for with the constant output disturbance observer, and partly because the influence of this initial component x_0 on the results decays with increasing cycles.

6.4 Comparison between Nonlinear MPC and Linear MPC

In most control methodologies, a linear model is used for the analysis and design of a control algorithm. Linear model predictive control has been a popular control methodology since the 1970s and consequently, linear MPC theory is quite mature. Important issues such as online computation, the interplay between modelling/identification and control, and stability are well addressed [98–100]. However, many systems are inherently nonlinear and are not well described using linear models. Nonlinear predictive control, the extension of well established linear predictive control to nonlinear systems, appears to be a well suited approach for the current problem. In this section, the advantage of applying NMPC to the nonlinear virtual process will be assessed by comparing the performance of the virtual process under both NMPC and LMPC.

Following the same procedure used to develop the nonlinear MPC controller, a linear MPC controller was developed based on a linear state-space model with a linear static gain function. The procedure applied is as follows,

- p,q are set equal to (2,2) in Equation (4.2), which can be expanded to

$$\begin{aligned}\mathbf{Z}(a_1,a_2) &= \sum_{i=0}^{1}\sum_{j=0}^{1}\theta_{ij}\, a_1^i\, a_2^j \\ &= \theta_{00}+\theta_{01}\, a_2+\theta_{10}\, a_1+\theta_{11}\, a_1\, a_2 \qquad (6.1)\end{aligned}$$

- Let $\theta_{11}=0$, then Equation (6.1) is simplified to

$$\mathbf{Z}(a_1,a_2)=\theta_{00}+\theta_{01}\, a_2+\theta_{10}\, a_1 \qquad (6.2)$$

  Note that Equation (6.2) is the linear expression of the function Z for the two independent variables a_1 and a_2.

- The coefficient θ_{01} and θ_{10} can then be computed using the least-squares method by comparing predicted and virtual process data.

Figure 6.13 compares the relative change in Maximum Die Tempeature using the linear fitting function Z with data generated from the virtual process at a selected die location across the process operational range. The 2D contour of the predicted data shows that the linear response surface does not match the variability exhibited by the virtual process response surface. Thus, the linear function Z does not accurately capture the nonlinear static gain of the virtual process.

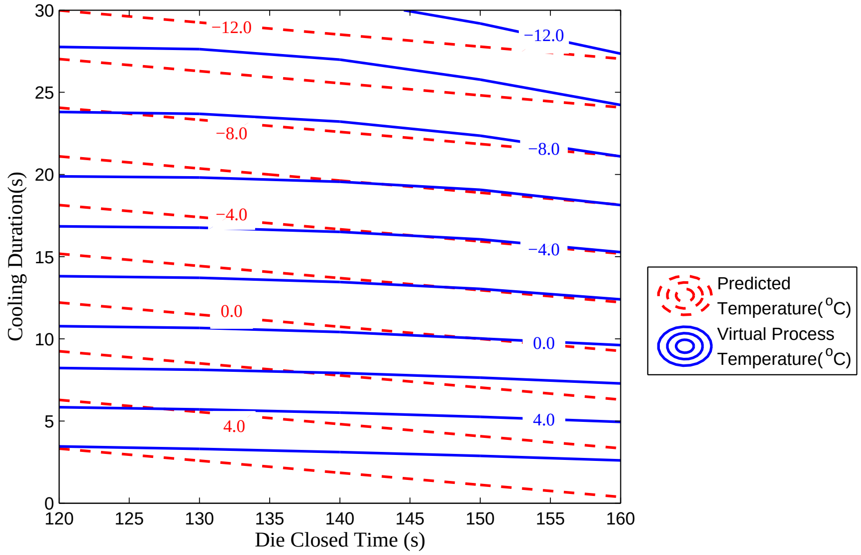
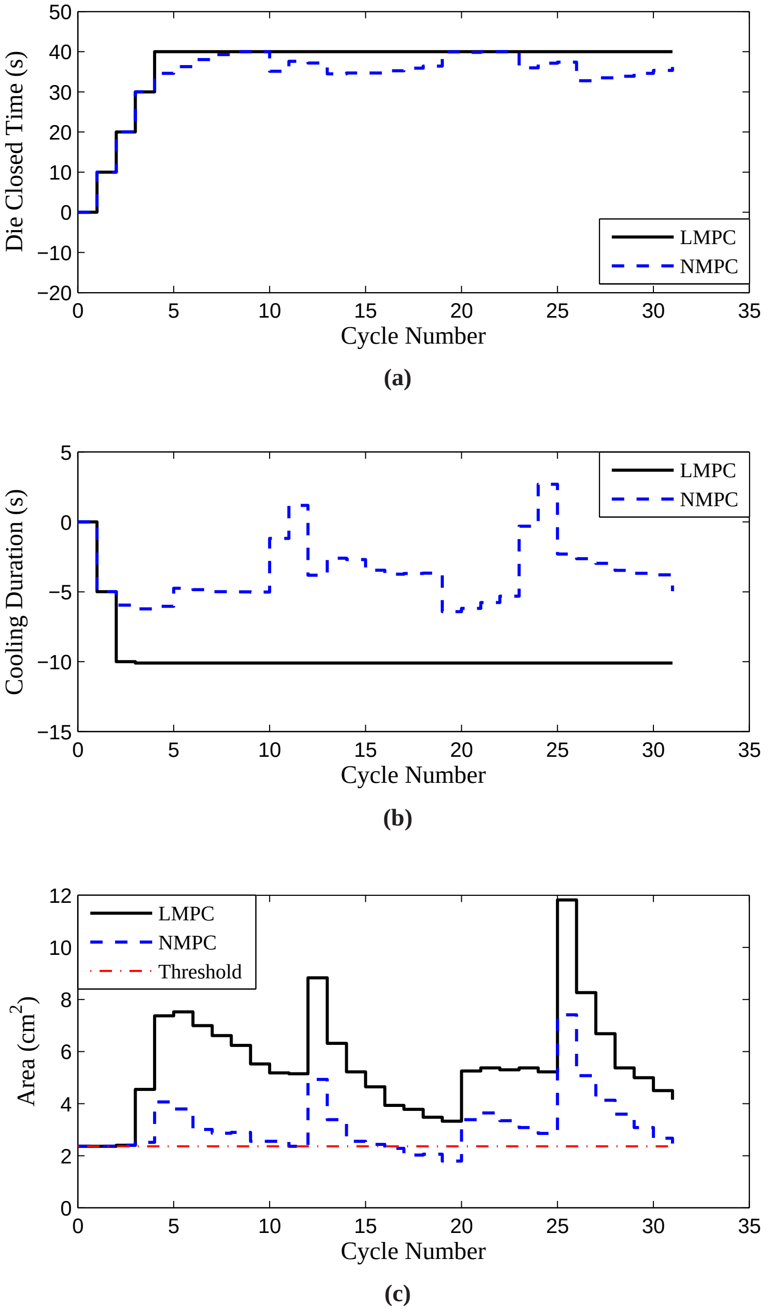


Figure 6.13: 2D Contours of Maximum Temperature for both Linear Static Gain Predicted by Equation (6.2) and Virtual Process Steady State Data over the Operational Range at a Random Die Location.

The linear state-space model was implemented within the MPC framework developed in this research and was used to control the virtual process experiencing the worst case disturbance scenario. Initial operational conditions of (150 s, 10 s) and (130 s, 30 s), selected for the same reason described in Section 6.2, were assessed. The performance of LMPC applied to the virtual process is compared with the NMPC results in Figures 6.14 and 6.15 for initial steady state conditions of (150 s, 10 s) and (130 s, 30 s), respectively. Table 6.5 summarizes the total area of encapsulated liquid and good casting ratio for LMPC and NMPC.

Table 6.5: Comparison of Sum of Area and Good Casting Ratio between LMPC and NMPC Simulations under Worst Case Disturbance Scenario.

Initial	Sum of Area(cm^2)			Good Casting Ratio(%)	
Steady State	LMPC	NMPC	Reduction rate (%)	LMPC	NMPC
(150s,10s)	171.7	97.5	43.2	3.2	19.4
(130s,30s)	205.6	137.8	33.0	48.4	74.2

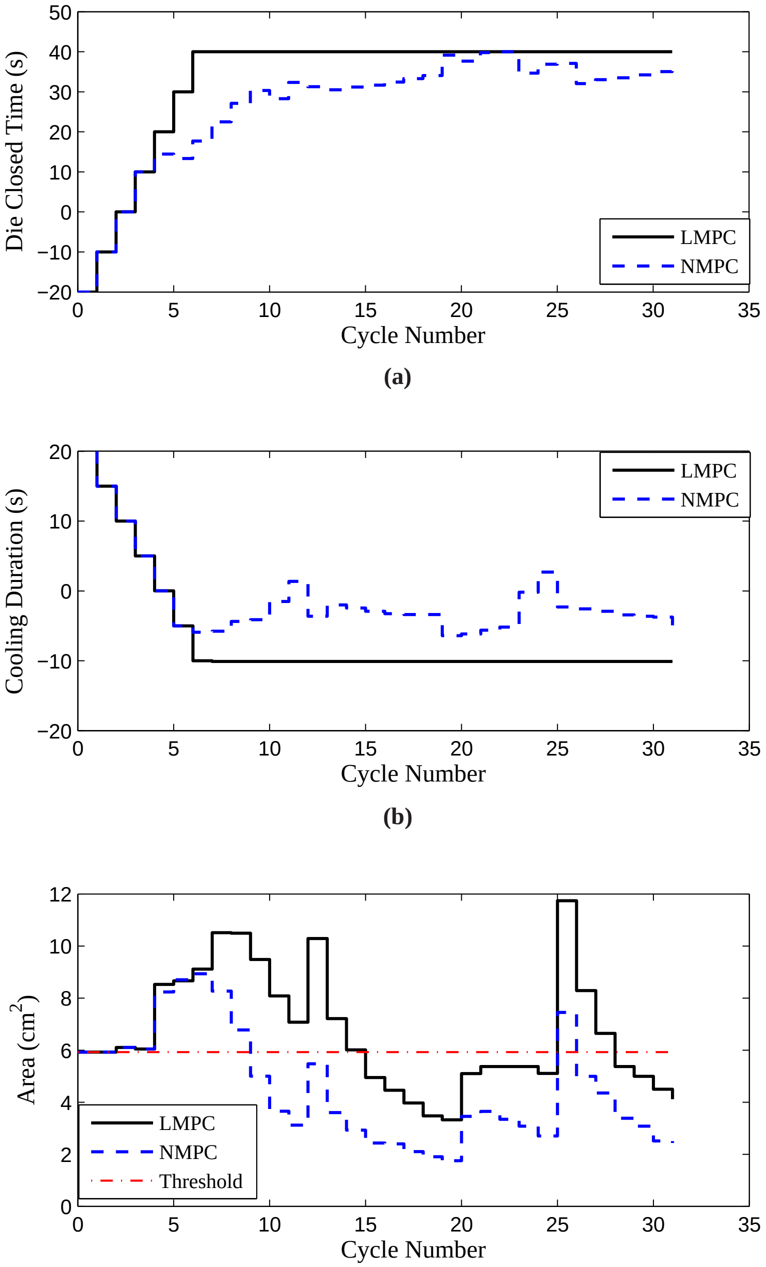


(a)

(b)

(c)

Figure 6.14: NMPC and LMPC Comparison of (a) Die Closed Time and (b) Cooling Duration and (c) the Area of Encapsulated Liquid on the Virtual Process under Worst Case Disturbance Scenario with Steady State Operation (150s,10s).



(a)

(b)

(c)

Figure 6.15: NMPC and LMPC Comparison of (a) Die Closed Time and (b) Cooling Duration and (c) the Area of Encapsulated Liquid on the Virtual Process under Worst Case Disturbance Scenario with Steady State Operation (130s,30s).

The results in Figure 6.14 compare the LMPC and NMPC performance during the worst case disturbance scenario starting from an initial steady state condition of (150 s, 10 s). NMPC results in a reduced number of cycles to achieve the threshold area compared to LMPC. The NMPC results in cycle-to-cycle variation of the control variables throughout the scenario, whereas the control variables computed by LMPC, shown in Figure 6.14a and 6.14b, follow an initial transient until they saturate at the hard constraints where they remain unchanged in remaining cycles. As a result, the virtual process controlled by LMPC does not reach the area threshold during the disturbance scenario.

The results in Figure 6.15 present LMPC and NMPC performance during the worst case disturbance scenario starting from an initial steady state condition of (130s,30s). In this case, the area threshold associated with the steady state condition of (130 s, 30 s) is higher than the previous case because (130s,30s) is further away from the optimal steady state than (150 s, 10 s). Consequently, both the NMPC and LMPC controlled virtual process are able to drive the process below the area threshold for a large portion of the disturbance scenario cycles generating higher good casting ratios than previous case. The NMPC performance for this case also exhibits a faster dynamic response than the LMPC. The control variables computed in LMPC show similar trajectories the previous case where they remain unchanged after reaching the hard constraints.

The improved performance of NMPC over LMPC for controlling the virtual process under worst case disturbance scenario is summarized in Table 6.5. When the process starts from a steady state operating point of (150s,10s), NMPC reduces the area of encapsulated liquid by 43.2 % relative to the LMPC area of 171.7 cm^2. The good casting ratio for NMPC is also larger than that using LMPC. For an initial steady state operating point of (130 s, 30 s), the total area of encapsulated liquid is reduced by 33.0 % when NMPC is used and the good casting ratio is 74.2 % for NMPC versus 48.4 % for LMPC. Based on these two evaluation metrics, the virtual process controlled using NMPC exhibits better performance than LMPC.

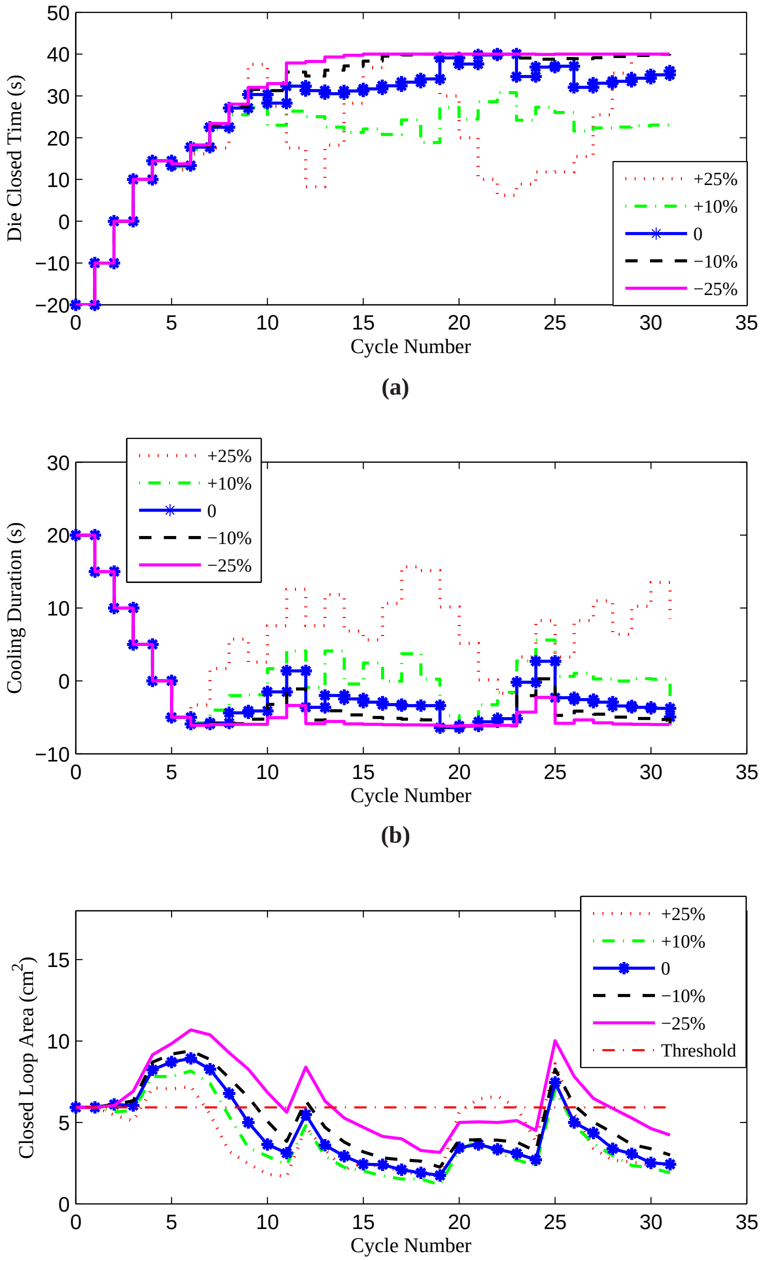
As linear static gain presented in the linear model does not accurately capture the nonlinear static gain of the virtual process, the model inaccuracy thus greatly affects the performance of LMPC on the virtual process. The above data and analysis suffice to hold one point - the virtual process does require a non-linear control solution to minimize macro-porosity during the process operation.

6.5 Robustness of the Control Solution

A control system is robust if it is insensitive to differences between the actual system and the model of the system which was used to design the controller. These differences are referred to as model/process mismatch or simply process uncertainty [101]. The sensitivity of the control system to uncertainty in the process is often used as a means of assessing robustness. In terms of control system performance, it is preferred that the control solution exhibit graceful degradation of performance in the presence of process uncertainty or variability.

Since the removal of heat from the casting determines the solidification sequence, two factors that affect heat transfer in the die have been selected to asses the robustness of the control solution. The selected parameters are the heat transfer coefficient(s) (h_1) along the die-casting interface and the heat transfer coefficient (h_2) in Cooling Channel 1. As noted, these heat transfer coefficients often vary in the industrial environment due to die coating thickness variations or water pressure variations, respectively. Thus, it is important to understand how sensitive the control solution is to uncertainty in these parameters. The steady state operational point for Die Closed Time and Cooling Duration equal to 130 s and 30 s, respectively was selected as the initial condition for the analysis. Both closed and open loop simulations were conducted for h_1 and h_2 varied by $\pm 25\%$ and $\pm 10\%$ from their baseline values described in chapter 2. The worst case disturbance scenario was used to evaluated the process response to each of these changes.

Figures 6.16 to 6.17 show the closed loop and open loop responses of the virtual process to the variation of h_1, while Figures 6.18 to 6.19 show the closed loop and open loop responses of the virtual process to the variation of h_2. Tables 6.6 and 6.7 summarize the total area of encapsulated liquid and the good casting ratio for open and closed loop simulations with varying h_1 and h_2, respectively.



(a)

(b)

(c)

Figure 6.16: The Responses of the Closed Loop Control Variables of (a) Die Closed Time and (b) Cooling Duration and (c) the Area of Encapsulated Liquid to Variation in h_1 under the Worst Case Disturbance Scenario with Steady State Operation (130s,30s).

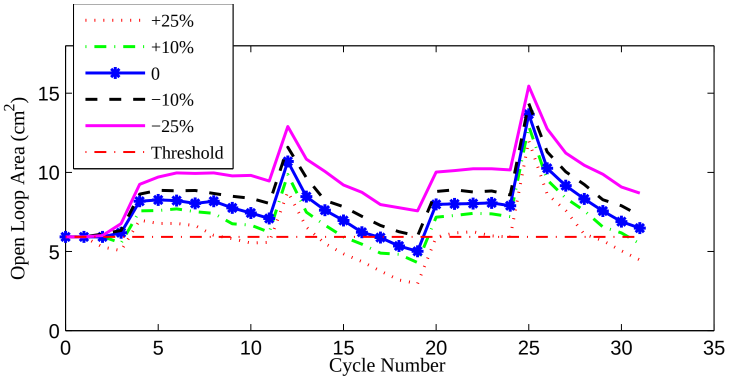
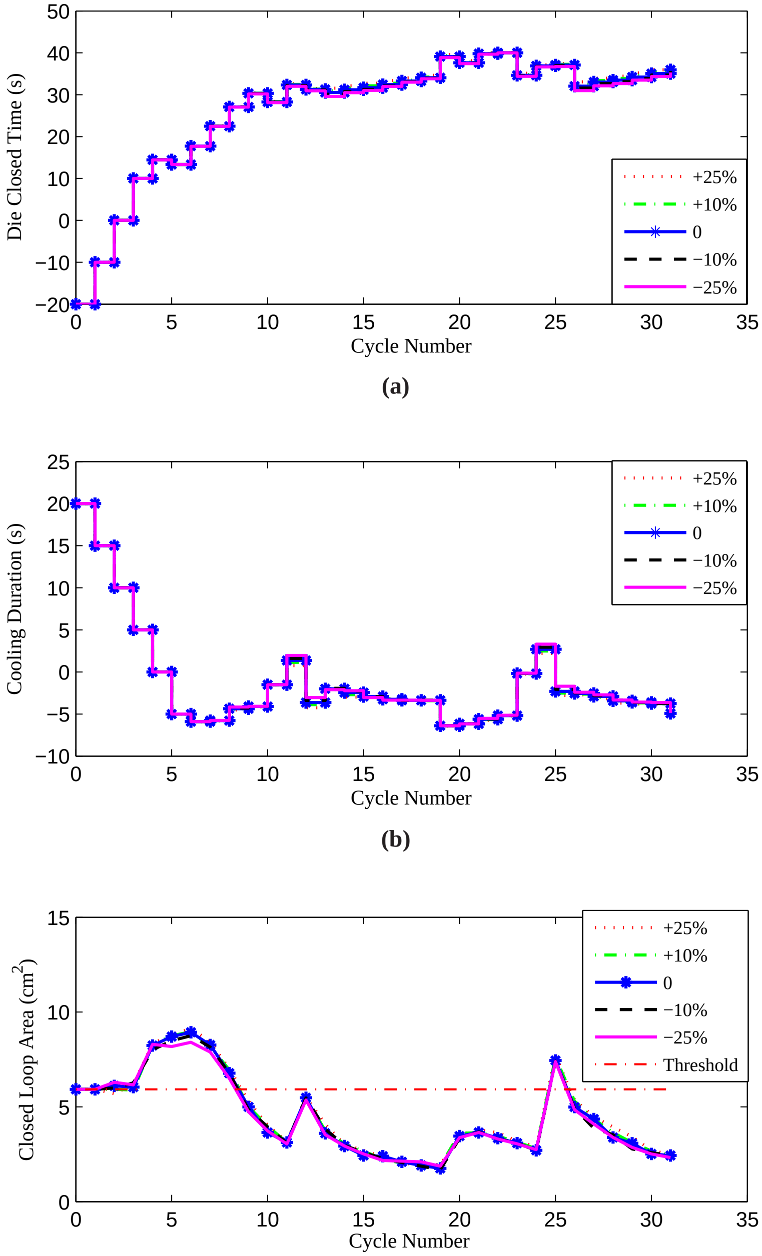


Figure 6.17: The Responses of the Open Loop Area of Encapsulated Liquid to Variation in h_1 under the Worst Case Disturbance Scenario with Steady State Operation (130s,30s).

Table 6.6: Comparison of the Total Area and Good Casting Ratio between Open Loop and Closed Loop Simulations for Variation in h_1 with Initial Steady State Operation (130s, 30s) under the Worst Case Disturbance Scenario.

h_1 varying range		-25%	-10%	0	$+10\%$	$+25\%$
Closed loop	Sum of Area(cm^2)	197.18	158.66	137.80	120.90	129.10
	Good casting ratio(%)	54.84	63.52	74.20	83.87	80.65
Open loop	Sum of Area(cm^2)	299.76	262.63	239.60	217.22	186.00
	Good casting ratio(%)	3.23	3.23	16.1	25.81	54.84



(a)

(b)

(c)

Figure 6.18: The Responses of the Closed Loop Control Variables of (a) Die Closed Time and (b) Cooling Duration and (c) the Area of Encapsulated Liquid to Variation in h_2 under the Worst Case Disturbance Scenario with Steady State Operation (130s,30s).

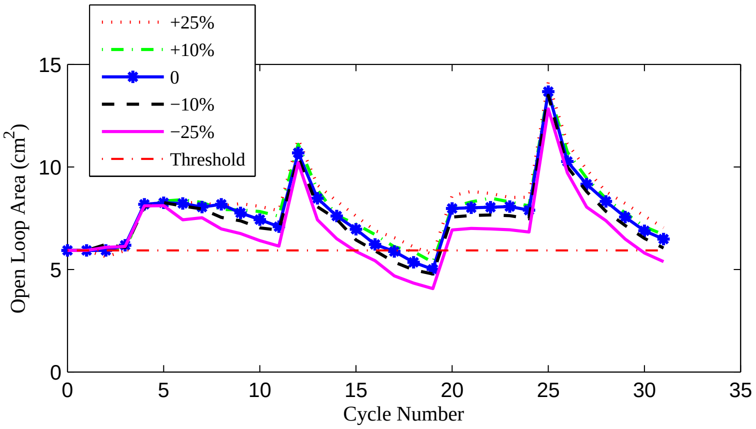
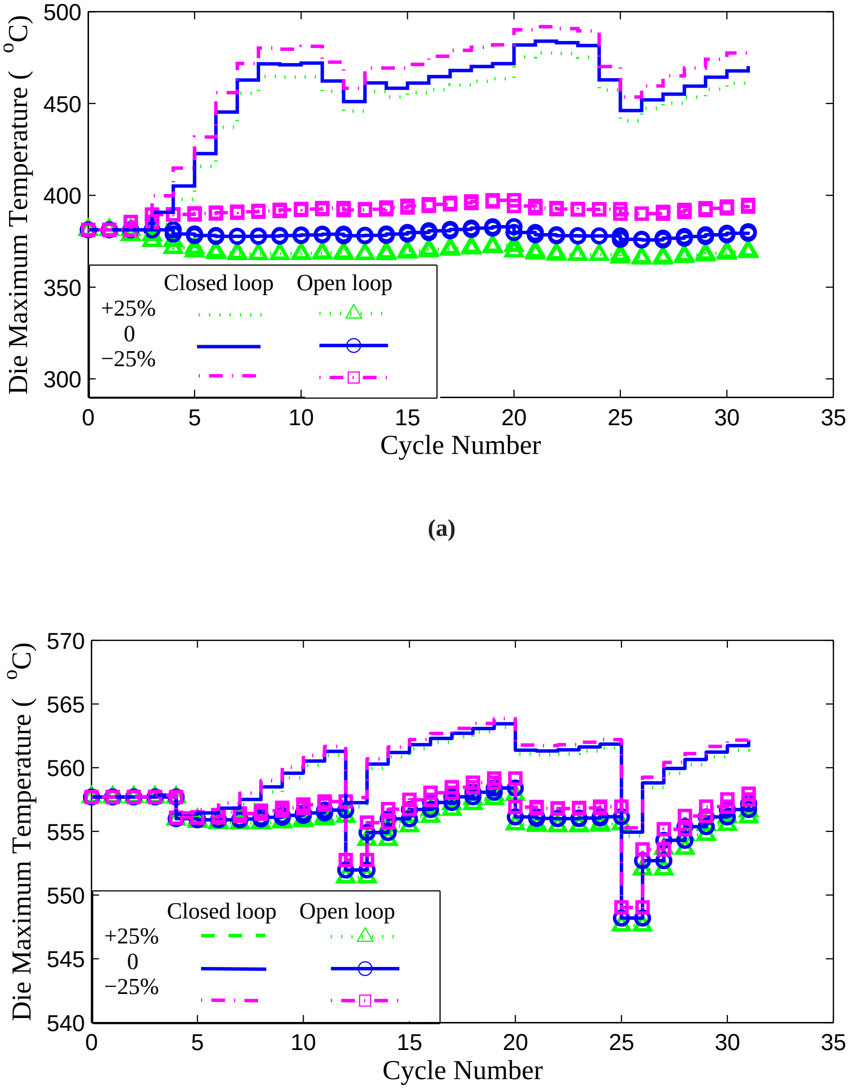


Figure 6.19: The Responses of the Open Loop Area of Encapsulated Liquid to Variation in h_2 on the Virtual Process under Worst Case Disturbance Scenario with Steady State Operation (130s,30s).

Table 6.7: Comparison of the Total Area and Good Casting Ratio between the Open Loop and Closed Loop Simulations for Variation in h_2 with Initial Steady State Operation (130s, 30s) under the Worst Case Disturbance Scenario.

h_1 varying range		-25%	-10%	0	$+10\%$	$+25\%$
Closed loop	Sum of Area(cm^2)	135.28	136.61	137.80	139.75	142.73
	Good casting ratio(%)	74.19	74.19	74.20	77.42	77.42
Open loop	Sum of Area(cm^2)	214.35	230.62	239.60	246.29	254.23
	Good casting ratio(%)	25.81	16.13	16.10	9.68	12.90



(a)

(b)

Figure 6.20: The Maximum Die Temperature Responses During Closed and Open Loop Operation to Variations in h_2 at the Die Locations (a) Closest and (b) Farthest From Cooling Channel 1 under Worst Case Disturbance Scenario with Steady State Operation (130s,30s).

Figures 6.16 and 6.17 show the closed and open loop responses of the virtual process to variations in h_1. When h_1 is varied, the closed loop responses of both control variables, Die Closed Time and Cooling Duration, are initially unchanged from the baseline. This insensitivity occurs as the controller pushes the processes towards its optimal operation condition (near 150 s, 10 s). As the process nears its optimal operational conditions, the control variable responses show sensitivity corresponding to the direction of variation of h_1. Generally, for increasing h_1 which corresponds to improved heat transfer from the casting to the die, Die Closed Time decreases as the less time is required to cool the casting and Cooling Duration increases. For decreasing h_1, Die Closed Time increases and Cooling Duration decreases until it reaches the constraints. The area of liquid encapsulation during closed loop operation increases initially due to the disturbance scenario, but then decrease below the threshold in all cases, as shown in Figure 6.16c, because the controller drives the process towards its optimal operation point.

It is interesting to note that with increased h_1, the open loop response of the process shows decreased sum of area of liquid encapsulation and increased good casting ratio. This suggests that the baseline behaviour of the process could be improved by enhancing interfacial heat transfer. For each h_1 condition, the closed loop response is better than open loop, as shown in 6.6. The closed loop responses present constant improvement with increasing h_1 up to the $+25\%$ condition. This discrepancy is caused by the control variable response which appears to be inconsistent with the baseline response, as shown in Figure 6.16a and 6.16b.

The open and closed loop response show little sensitivity to variations in h_2. In fact, both control variables (Die Closed Time and Cooling Duration) match the baseline throughout the scenario, as shown in Figure 6.18a and 6.18b. The encapsulated liquid areas show small deviations from the baseline for during closed loop operation and small variations during open loop operation (refer to Figure 6.18c and 6.19). During open loop operation, increasing h_2 causes increased liquid encapsulation. The effects of h_2, further summarized in Table 6.7, are clearly minimal.

Although the control variable responses and the area of encapsulated liquid appear to be insensitive to variations in h_2, the temperatures at the 9 locations in the die (refer to Figure 4.1) monitored during process operation exhibit varying degrees of sensitivity. The sensitivity to changes h_2 at each location is dependent on the distance of the location to Cooling Channel 1. To examine these effects, the maximum die temperatures at 2 of the 9 locations (the closest and farthest away form Cooling Channel 1) has been plotted in Figure 6.20 for both closed and open loop conditions while varying h_2. For the die location closest to Cooling Channel 1, the maximum die temperature varies by $\sim 20-30°$C relative to the baseline response. The temperature variation at the location farthest from Cooling Channel 1 shows nearly no change ($\sim 0-2°$C) during the entire disturbance scenario. Overall, the open loop response at each location, corresponding to a 30 s cooling duration, shows larger variations than the closed loop response as the NPMC controller decreases the cooling duration to approach the optimal process condition.

6.6 Summary and Conclusion

In this chapter, the NMPC controller was applied to control the virtual process through a variety of disturbance scenarios and with different starting conditions. Starting from either a steady state or start-up operational condition, the NMPC exhibits superior performance over the open loop operation for all conditions. Further, it demonstrated an ability to automatically adjust control variables from cycle to cycle to drive the closed loop process towards the optimal dynamic or static operating point. This automatic adjustment is significant since in practice the optimal operational condition will vary depending on the state of the process. NMPC also shows significantly improved performance compared to LMPC. The robustness of the control solution was assessed by evaluating its sensitivity to changes in the die-casting interfacial heat transfer and cooling intensity in Cooling Channel 1. The closed and open loop responses were sensitive to the variations in the interfacial heat transfer and insensitive to variations in the cooling intensity of Cooling Channel 1. In all cases, NMPC was shown to perform better than open loop and to consistently seek the optimal operation condition.

Chapter 7

Conclusions

The objective of this research program was to develop an advanced control solution for a low pressure die casting process to compensate for the negative effects of process disturbances and to drive the process towards its optimal operational condition based on minimizing macro-porosity in the casting. In the pursuit of this objective, a 3D FE model of a LPDC demonstration casting, which was purposely designed with a high propensity to form macro-porosity, was developed using ABAQUS and validated with experimental data. This process model predicts temperature distribution within the casting and die during a casting cycle which is then used to determine the volume of encapsulated liquid via a post-programming script. A Perl language wrapper script has been developed to transform the process model into a virtual process in which the communications and functionality of an industrial process are emulated. The virtual process was used as a platform for developing and testing an advanced process control solution. By developing and evaluating the control solution offline from the industrial process, the expenses associated with the extensive plant trials necessary to develop such a system can be avoided.

A methodology has been developed to analyze the correlation between die temperatures and the volume of encapsulated liquid over the operational range for a particular die location. The metrics for evaluating the correlation, defined as the Correlation Index (CI) and the Standard Deviation (STD) of the correlations, have been developed. Calculating these metrics at each location in the die enabled the determination of the optimal location to monitor die temperature for correlation to liquid encapsulation in the casting. Using this methodology, characteristic temperatures within each casting cycle were evaluated and the Maximum Die Temperature was observed to be a good indicator of the volume of encapsulated liquid occurring in the casting. The method for correlating die temperatures to liquid encapsulation and its application to assess die locations is a new and important contribution from this research. A linear regression (LR) expression has also been developed to predict the amount of encapsulated liquid based on the Maximum Die Temperatures at selected locations. The combined use of the CI and LR methods offers a means of selecting and assessing locations to monitor dies and represents a major advancement over the traditional trial-and-error approach regularly used in industry.

To reduce simulation time, a 2D-axisymmetric version of the 3D process model, was developed and used for the system identification (SI) of the virtual process. Using the input-output data from open loop simulations, a state-space model with linear dynamics and nonlinear static gain was identified. The state-space model was shown to accurately predict the temperature response of the die to a range of varying process inputs. A LR expression was also fit to predict the area of encapsulated liquid based on the Maximum Die Temperatures at selected locations. The combination of the state-space model to predict die temperatures and the LR expression to convert these temperatures to a measure of the liquid encapsulation provides a direct link between process inputs and the area of liquid encapsulation. This nonlinear SI approach linking a measure of casting quality to process inputs is a novel and easy to understand contribution from this research.

A nonlinear model-based predictive controller (NMPC) was developed for use in controlling an operational LPDC process. A cost function, containing two weighting terms for the predicted area of liquid encapsulation and the Die Closed Time respectively, was included in the NMPC consistent with the control goals of improving casting quality and increasing production rate. The tuning parameters associated with the cost function such as prediction horizon H_p and control horizon H_u were adjusted to optimize the closed loop performance. The NMPC was intentionally designed to mitigate the negative effects of variations in the Incoming Metal Temperature and Die Open Time on the process.

The performance of the NMPC developed in this research was assessed with the virtual process. The results of the control solution evaluations highlighted that, under any disturbance scenario, starting from either steady state or start-up within the operational range considered, the NMPC was able to automatically adjust the control variables, driving the process towards optimal dynamic or static operating points. The automatic adjustment exhibited by this control method is important as the system's ideal operating point may not be known a priori. In this research, NMPC was shown to be necessary and to provide improved performance compared to linear MPC. The robustness of the control solution was assessed by conducting a sensitivity analysis on the response of the NMPC to variations in the die-casting interfacial heat transfer coefficient and the heat transfer coefficient applied in the cooling channel. Overall, the controlled process was shown to be sensitive to variations in the interfacial heat transfer, but insensitive to changes to the heat transfer in the cooling channel.

This research provides a complete and reliable solution for the development of an advanced control method for Low Pressure Die Casting to minimize macroporosity in a casting by driving the process towards its optimal operational conditions. The solution provides all of the necessary tools required to design a control methodology offline, from the development of a validated FE model, correlation analysis and determination of optimal die locations, nonlinear system identification linked to the LR expression of encapsulated liquid, controller tuning and closed loop performance evaluation. This closed loop solution can then be applied to the real casting process to present better performance than the open loop.

Recommendations for Future Work

The next logical step for this research is the application of this methodology to develop a control solution for an operational low pressure die casting process. This could be done in a phased approach where initially the control system would run in an advisory mode in parallel with the traditional open loop system to provide the operator with calculated control variable changes. Following satisfactory evaluation in this mode, the next step would be to close the loop and automatically change the control variables without operator intervention. The effects of operational failures in the industrial environment on the controller should be assessed. For instance, the malfunction or termination of a control thermocouple signal would have a negative effect on the performance of the control solution. Assuming the malfunction of a thermocouple can be determined, it may be possible to develop a solution to mitigate the loss of thermocouples.

In this research, the correlation metrics and a LR expression were used to identify and quantify the volume of macro-porosity formed during casting based on die temperature. This analysis approach may be extended / adapted to quantitatively identify other defects such as micro-porosity using die temperatures and other process parameters that can be obtained by sensors. The development of such an expression would then enable its use in a control solution to minimize micro-porosity.

In addition to low pressure die casting, the same basic framework could be applied to any other cyclic casting processes which can be modelled using FEM simulation software such as ABAQUS or any other high-fidelity models. To enable the application of the methodology developed in this dissertation to other processes, some effort should be spent on determining and reducing the number of model runs required. In this work, no consideration was given to the number of cycles simulated, but to save controller development costs, shortening simulation time and simplifying procedures of developing a controller should be considered.

Bibliography

[1] B. Zhang, D. M. Maijer, and S. L. Cockcroft. Development of a 3-d thermal model of the low-pressure die-cast (lpdc) process of a356 aluminum alloy wheels. *Materials Science and Engineering A*, 465:295, 2007. → pages xii, 1, 4, 7, 29, 30, 32, 33

[2] F. Bonollo, J. Urban, B. Bonatto, and M. Botter. Gravity and low pressure die casting of aluminium alloys: a technical and economical benchmark. *la metallurgia italiana*, 6:23–32, 2005. → pages 1

[3] B. Andresen. *Die Casting Engineering: a hydraulic, thermal, and mechanical process*. Marcel Dekker, New York, 2005. → pages 1, 2, 132

[4] R. Esdaile, T. Nguyen, G. De-Looze, and M. Murray. Proceedings of the first international non-ferrous processing and technology conference. pages 213–218, Warrendale,PA, 1997. TMS International. → pages

[5] W. Muller and F. Feikus. *AFS Trans*, 104:1111–1117, 1996. → pages 1

[6] B. Zhang, S. L. Cockcroft, D. M. Maijer, J. D. Zhu, and A. B. Phillion. Casting defects in low-pressure die-cast aluminum alloy wheels. *JOM JOURNAL OF THE MINERALS, METALS AND MATERIALS SOCIETY*, 57:36–43, Nov 2005. → pages 1, 7

[7] D. P. K. Singh, G. D. Mallinson, S. M. Panton, and N. Palle. Die design strategy for improved productivity and quality in die casting. *AFS Transactions*, 99(25):127–133, 1999. → pages 1, 11

[8] A. J. Wilkinson, G. E. Wilson, A. Connor, and N. A. Smith. Cim and control of pressure die-casting in the automotive industry. *Computing and Control Engineering Journal*, 5(3):125–130, 1994. → pages 2

[9] T. B. Yang, X. Chen, and H. Hu. A fuzzy pid thermal control system for die casting processes. In *22nd IEEE International Symposium on*

Intelligent Control Part of IEEE Multi-conference on Systems and Control, Singapore, Oct 2007. → pages 4, 15, 20, 21

[10] J. D. Zhu, S. L. Cockcroft, and D. M. Maijer. Modeling of microporosity formation in a356 aluminum alloy casting. *Physical Metallurgy and Materials Science: Metallurgical and Materials Transactions*, 37A(3A):1075, 3 2006. ISSN 1073-5623. → pages 6

[11] C. Pequet, M. Gremaud, and M. Rappaz. Modeling of microporosity, macroporosity, and pipe-shrinkage formation during the solidification of alloys using a mushy-zone refinement method: Applications to aluminum alloys. *METALLURGICAL AND MATERIALS TRANSACTIONS A*, 33A:2095, 7 2002. → pages 6

[12] D. M. Maijer, W. S. Owen, and R. A. Vetter. Development of model predictive control for aluminum wheel casting via a virtual process model. *Journal of Materials Process Technology*, 209(4):1965–1979, Feb 2009. → pages 7, 8, 9, 13, 17, 21

[13] S. Palanisamy, C. R. Nagarajah, K. Graves, and P. Iovenitti. A hybrid signal pre-processing approach in processing ultrasonic signals with noise. *International Journal of Advanced Manufacturing Technology*, 42(7-8):766–771, June 2009. → pages 7

[14] M. Cox and R. A. Harding. Influence of tilt filling on weibull modulus of 2l99 aluminium investment castings. *Materials Science and Technology*, 23(2):214–224, 2007. → pages

[15] L. Jeffery. Non-destructive testing: 5 ways to ensure defect-free deliveries. *Modern casting*, pages 49–52, April 1998. → pages 7

[16] W. Bishenden and R. Bhola. Die temperature control. In *Transaction of the 20th International Die Casting Congress and Exposition*, pages 161–164, Cleveland,USA, 1999. North American Die Casting Association. → pages 7, 20, 48

[17] H. Hu, F. Chen, X. Chen, Y. Chun, and P. Cheng. Effect of cooling water flow rates on local temperatures and heat transfer of casting dies. *Journal of Materials Processing Technology*, 148(1):57–67, 2004. → pages 7, 8

[18] T. B. Yang, H. Hu, X. Chen, Y. L. Chu, and P. Cheng. Thermal analysis of casting dies with local temperature controller. *Int J Adv Manuf Technol*, 33:277–284, 2007. DOI 10.1007/s00170-006-0468-8. → pages 7, 16, 20

[19] H. Shang, J. B. Wiskel, J. F. Forbes, and H. Henein. Exploiting model fidelity to control metals processing. *JOM*, 55(3):41–45, 2003. → pages 8, 17, 21

[20] Z. Q. Sheng, R. Taylor, and M. Strazzanti. Fem-based progressive drawing process design. *The International Journal of Advanced Manufacturing Technology*, 36(3):226 – 236, Mar 2008. ISSN 0268-3768. → pages 8, 9

[21] G. C. Goodwin, S. F. Graebe, and M. E. Salgado. *Control System Design*. Prentice Hall, NJ, 2001. → pages 8, 16, 83, 107

[22] R. I. L. Guthrie. Fluid flows in metallurgy-friend or foe? *Metallurgical and Materials Transactions B*, 35(3):417 – 437, 06 2004. ISSN 1073-5615. → pages 9

[23] M. F. Zhu, C. P. Hong, D. M. Stefanescu, and Y. A. Chang. Computational modeling of microstructure evolution in solidification of aluminum alloys, metallurgical and materials transactions b. *Process Metallurgy and Materials Processing Science*, 38(4):517 – 524, 08 2007. ISSN 1073-5615. → pages 9

[24] S. Tin, P. D. Lee, A. Kermanpur, M. McLean, and M. Rist. Integrated modeling for the manufacture of ni-based superalloy discs from solidification to final heat treatment. *Physical Metallurgy and Materials Science: Metallurgical and Materials Transactions A*, 36(9):2493 – 2504, Sep 2005. ISSN 1073-5623. → pages 9

[25] D. M. Stefanescu. *Science and Englineering of Casting Solidification*. Springer, 2008. pp: 81. → pages 10

[26] M. S. Shadloo and A. Kimiaeifar. Application of homotopy perturbation method to find an analytical solution for magnetohydrodynamic flows of viscoelastic fluids in converging/diverging channels. *Proceedings of the IMechE, Part C: Journal of Mech. Engineering Science*, Jun 2010. Online ISSN 2041-2983. → pages 10

[27] K. W. Morton and D. F. Mayers. *Numerical Solution of Partial Differential Equations: An Introduction*. Cambridge University Press, 2005. → pages 10

[28] O. Rbenkonig. *The Finite Difference Method (FDM) - An introduction*. Albert Ludwigs University of Freibur, 1st edition, Apr 2005. ISSN 0997-7538. → pages 10

[29] W. S. Hwang and R. A. Stoehr. Molten metal ow pattern prediction for complete analysis of near net shape casting. *Materials Science and Technology*, 4:240 –250, 1988. → pages 10

[30] J. N. Reddy. *An Introduction to the Finite Element Method*. New York: McGraw-Hill, 2nd edition, 1993. McGraw-Hill series in mechanical engineering. → pages 11

[31] W. Velez, D. Gomez, and P. Thomson. Finite element model updating. *DYNA-COLOMBIA*, 76:177–189, June 2009. ISSN 0012-7353. → pages 11

[32] A. Buhrig-Polaczek, P. R. Sahm, P. V. Heeke, and D. Grzesik. In A. F. Giami and G. J. Abbaschian, editors, *Modeling of Casting and Welding Processes IV*, pages 617–624, Warrendale, PA, 1988. TMS. → pages 11

[33] P. A. Kobryn and S. L. Semiatin. Determination of interface heat-transfer coefficients for permanent-mold casting of ti-6al-4v. *Metallurgical and Materials Transactions B*, 32(4):685 – 695, Aug 2001. ISSN 1073-5615. → pages 11

[34] J. H. Jeong; and D. Y. Yang. Finite element analysis of filling stage in die-casting process using marker surface method and adaptive grid refinement technique. *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS*, 44:209–230, 2004. DOI: 10.1002/fld.637. → pages 11

[35] M. O. Shabani, M. Alizadeh, and A. Mazahery. Modelling of mechanical properties of cast a356 alloy. *Fatigue and Fracture of Engineering Materials and Structures*, 34(12):1035 – 1040, 12 2011. ISSN 8756-758X. → pages 11

[36] K. S. Chae, S. J. Lee, K. H. Ahn, and S. J. Lee. Coupled approach of analytical and numerical methods for shape prediction in sheet casting process. *Korea-Australia Rheology Journal*, 13(3):131–139, Sep 2001. → pages 11

[37] N. Hofmann, S. Olive, G. Laschet, F. Hediger, J. Wolf, and P. R. Sahm. Numerical optimization of process control variables for the bridgman casting process. *Modelling and Simulation in Materials Science and Engineering.*, 5:23–34, 1997. → pages 11

[38] F. Matrinsen, B. A. Foss, and T. A. Johansen. A control relevant dynamic model of grate sintering. In *International conference on Control Applications*, pages 1294–1299, Hawaii, USA, 1999. Proceedings of the 1999 IEEE. → pages

[39] D. Maijer, M. A. Wells, and S.L.Cockcroft. Mathematical modelling of porosity formation in die casting a356 wheels. *Mathematical Modelling in Metal Processing and Manufacturing, Canadian Institute of Mining, Metallurgy and Petroleum*, page 14, 2000. → pages

[40] D. Constales, J. Kacur, and R. V. Keer. On the optimal cooking strategy for variable-speed continuous casting. *Internation Journal for Numerical Methods in Engineering*, 53(3):125–130, June 1994. → pages

[41] J. R. S. Zabadal, M. T. M. B Vilhena, and S.Q.Bogado Leite. Heating transfer process simulation by finite differences for online control of ladle furnaces. *Ironmaking and Steelmaking*, 31(3):227–234. → pages 11

[42] M. Cross, S. Johnson, and C. lerotheou. The impact of hardware on the modeling and analysis of materials processing operations. In V. R. Voller, M. S. Stachowicz, and B. G. Thomas, editors, *TMS: Materials Processing in the Computer Age*, pages 35–44, Warrendale, PA, 1991. → pages 11

[43] A. Montanaro. On control of linear spring-mass chains. *EUROPEAN JOURNAL OF MECHANICS A-SOLIDS*, 25(3):559–564, 2005. ISSN 0997-7538. → pages 12

[44] K. H. Lundberg. *Linear dual inverted pendulum control,*. PhD thesis, Massachusetts Institute of Technology,Electrical Engineering and Computer Science, Aug 2005. → pages 12

[45] J. H. Lee and A. W. Dorsey. Monitoring of batch processes through state-space models. *AlChE Journal*, 50:1198–1210, 2004. issue:6. → pages 13

[46] C. McNabb. Projection based mimo control performance monitoring: I-covariance monitoring in state space. *Journal of process control*, 13(8):739, 2003. → pages

[47] B. Nie, Q. Liu, D. Liao, and J. Ding. Statistical process control based on state space model. In *Information Science and Engineering (ICISE), 2010 2nd International Conference*, pages 1–8, Hangzhou, china, 12 2010. ISBN: 978-1-4244-7616-9. → pages 13

[48] H. A. Nielsen and H. Madsen. *Predicting the Heat Consumption in District Heating Systems using Meteorological Forecasts*. Department of Mathematical Modelling, Technical University of Denmark. → pages 13

[49] H. A. Nielsen and H. Madsen. Modelling the heat consumption in district heating systems using a grey-box approach. *Energy and Buildings*, 38(1):63–71, 2006. → pages 13

[50] Y. Shmaliy. *Continuous-Time Systems*. springer, 2007. → pages 13

[51] E. Gershon, U. Shaked, and I. Yaesh. 4 continuous-time systems: Tracking control, h 8-control and estimation of state-multiplicative linear systems. *Lecture Notes in Control and Information Sciences*, 318:55–73, 2005. → pages 13

[52] A. Ramakalyan, P. Kavitha, and S. H. Vijayalakshmi. Discrete-time systems. In *Resonance*, volume 5 of *2*, pages 39–49, Feb 2000. ISSN 0971-8044. → pages 13

[53] Z. H. Wu and R. Pei. Constrained discrete time-delay system robust model predictive control. In *Control Conference,27th Chinese*, pages 516 – 520, Kunming, china, Jul 2008. ISBN: 978-7-900719-70-6. → pages 13

[54] M. Lazar, W. P. M. H. Heemels, S. Weiland, and A. Bemporad. Stabilizing model predictive control of hybrid systems. *IEEE Trans on Automatic Control*, 51(11), Nov 2006. → pages 13

[55] P. Kachroo. Chattering reduction and error convergence in the sliding-mode control of a class of nonlinear systems. *IEEE Trans on Automatic Control*, 51(11), Nov 2006. → pages 13

[56] J. P. Hespanha. *Linear systems theory*. Princeton, N.J: Princeton University Press, 2009. → pages 13

[57] A. E. Bryson. *Applied linear optimal control: examples and alogrithms*. Cambridge University Press, 1st edition, Sep 2002. → pages 13

[58] S. P. Bhattacharyya, A. Datta, and L. H. Keel. *Linear control theory: structure, robustness, and optimization, Automation and control engineering*. CRC Press, 1st edition, Jan 2009. → pages 13

[59] I. Rivals and L. Personnaz. Black-box modeling with state-space neural networks,neural adaptive control technology. *World Scientific*, pages 237–264, 1996. → pages 14

[60] H. N. Al-Duwaish. Identification of wiener model using genetic algorithms. In *5th IEEE GCC Conference and Exhibition*, pages 1–4, 2009. ISBN:9781424438853. → pages 14

[61] E. W. Bai. Frequency domain identification of wiener models. *Automatica*, 39:1521–1530, 2003. → pages 15

[62] Y. Li, Z. Z. Mao, and Y. Wang. Bcrls identification method for hammerstein-wiener model. In *2010 International Conference on Measuring Technology and Mechatronics Automation*, volume 1, pages 745 – 748, 2010. ISBN1424450012. → pages 15

[63] C. A. Bode, B. S. Ko, and T. F. Edgar. Run-to-run control and performance monitoring of overlay insemiconductor manufacturing. *Control Engineering Practice*, 12(7):893–900, 2004. → pages 15, 20

[64] Y. Abe, M. Konishi, J. Imai, R. Hasagawa, M. Watanabe, and H. Kamijo. Pid gain tuning method for oil refining controller based on neural networks. *International Journal of Innovative Computing Information and Control*, 4(10):2649 – 2662, Oct 2008. ISSN 1349-4198. → pages

[65] F. Farahmand. *Alignment, modeling and iterative adaptive robust control of cross-directional processes*. PhD thesis, UBC, 2009. dissertation for Doctor of Philosophy. → pages 15

[66] S. Skogestad. Simple analytic rules for model reduction and pid controller tuning. *Journal of Process Control*, 13(4):291–309, 2003. ISSN 0959-1524. → pages 16

[67] K. Warwick and D. Rees. *Industrial digital control systems*. Institution of Electrical Engineers Stevenage, UK, 1988. → pages 16

[68] A. G. Wills and W. P. Heath. Application of barrier function based model predictive control to an edible oil refining process. *JOURNAL OF PROCESS CONTROL*, 15(2):183 – 200, 2004. ISSN 0959-1524. → pages 17

[69] R. Silva and W. Kwong. Nonlinear model predictive control of chemical processes. *BRAZILIAN JOURNAL OF CHEMICAL ENGINEERING*, 16(1):83 – 99, 03 1999. ISSN 0104-6632. → pages

[70] M. Mercangoz and F. Doyle. Model-based control in the pulp and paper industry. *IEEE Control Systems Magazine*, 26(4):30 – 39, 08 2006. ISSN 1066-033X. → pages

[71] A. Poloski and J. Kantor. Application of model predictive control to batch processes. *COMPUTERS and CHEMICAL ENGINEERING*, 27(7):913 – 926, 07 2003. ISSN 0098-1354. → pages 17

[72] J. M. Maciejowski. *Predictive Control with Constraints*. Prentice Hall, 2002. → pages 18, 100, 108, 119

[73] K. B. Ariyur and M. Krstic. *Real-Time Optimization by Extremum-Seeking Control*. A JOHN WILEY and SONS INC., 2003. → pages 18

[74] K. J. Astrom and B. Wittenmark. *Adaptive control*. Addison Wesley, Reading, MA, 2nd edition, 1995. → pages 18

[75] G. C. Goodwin and K. S. Sin. *Adaptive Filtering Prediction and Control*. Englewood Cliffs, NJ: Prentice-Hall, 1984. → pages

[76] A. Ioannou and J. Sun. *Stable and Robust Adaptive Control*. Englewood Cliffs, NJ: Prentice-Hall, 1995. → pages 18

[77] M. Krstic, I. Kanellakopoulos, and P. V. Kokotovic. *Nonlinear and adaptive control design*. New York: Wiley, 1995. → pages 18

[78] E. Visser, B. Srinivasan, S. Palanki, and D. Bonvin. A feedback-based implementation scheme for batch process optimization. *Journal of Process Control*, 10(5), 399-410 2000. → pages 20

[79] F. E. Thomas. Control of unconventional processes. *Journal of Process Control*, 6(2-3):99–110, 1996. → pages

[80] M. Huzmezan, W. A. Gough, and S. Kovac. Advanced control of batch reactor temperature. In *World Batch Forum*, Woodcliff Lake,USA, 2002. → pages 20

[81] F. Xaumier, M. L. Lann, M. Cabassud, and G. Casamatta. Experimental application of nonlinear model predictive control: Temperature control of an industrial semi-batch pilotplant reactor. *Journal of Process Control*, 12(6):687–693, 2002. → pages 20

[82] K. S. Lee and J. H. Lee. Convergence of constrained model-based predictive control for batch processes. *IEEE Transactions on Automatic Control*, 45(10), 2000. → pages 20

[83] T. Yang, H. Hu, X. Chen, Y. Chu, and P. Cheng. On-line thermal management system for die casting processes. In National

Research Council Canada, editor, *Proceedings of 5th International Workshop on Advanced Manufacturing Technologies*, pages 189–196, 2005. → pages 20

[84] S. Thompson, S. L. Cockcroft, and M. A. Wells. Advanced light metals casting development: solidification of aluminium alloy a356. *Materials Science and Technology*, 20:194–200, 2004. → pages 29

[85] Source. *ASM Metals Reference*. American Society for Metals, Metals Park, Ohio 44073, 2nd edition, 1984. → pages x, 29

[86] K. C. Mills. *Recommended Values of Thermophysical Properties for selected Commercial Alloys*. Woodhead Publishing, Cambridge,UK, 2002. → pages x, 29, 34

[87] M. Drosg. *Dealing with uncertainties: a guide to error analysis*. Berlin : Springer, 2nd edition, 2009. → pages 40

[88] S. Ghosh and C. R. Rao. *Design and Analysis of Experiments. Handbook of Statistics*. North-Holland, 1996. → pages 48, 72

[89] D. H. Wu and M. S. Chang. Use of taguchi method to develop a robust design for the magnesium alloy die casting process. *Materials Science and Engineering*, 379:366–371, 2004. → pages 48

[90] R. N. Kackar. Off-line quality control,parameter design and the taguchi method. *J.Qual.Technol.*, 17(4):176–188, 1985. → pages

[91] N. Logothetis. The role of data transformation in the taguchi analysis. *Quality and Reliability Engng Int.*, 4:49–61, 1988. → pages 48

[92] L. Ljung. *System Identification: Theory for the user*. Upper Saddle River, NJ: Prentice Hall, 2nd edition, 1999. → pages 67

[93] P. V. Overschee and B. D. Moor. N4sid: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30:75–93, 1994. → pages 67, 87

[94] P. T. Boggs and J. W. Tolle. Sequential quadratic programming. *Acta Numerica*, 4:1–51, 1995. → pages 108

[95] M. J. D. Powell. A fast algorithm for nonlinearly constrained optimization calculations. *Numerical Analysis,Lecture Notes In Mathematics*, 630, 1978. → pages 108, 109

[96] M. J. D. Powell. The convergence of variable metric methods for nonlinearly constrained optimization calculations. In O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, editors, *Nonlinear Programming 3*. Academic Press, 1978. → pages 109

[97] M. I. Ribeiro and A. R. Pais. Kalman and extended kalman filters: Concept, derivation and properties. online, 2004. → pages 109

[98] J. H. Lee and B. Cooley. Recent advances in model predictive control and other related areas. In J. C. Kantor, C. E. Garcia, and B. Carnahan, editors, *Fifth International Conference on Chemical Process Control C CPC V*, page 201C216. American Institute of Chemical Engineers, 1996. → pages 154

[99] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M . Scokaert. Constrained model predictive control: stability and optimality. *Automatica*, 26(6):789–814, 2000. → pages

[100] M. Morari and J. H. Lee. Model predicitve control: Past, present and future. *Comp. and Chem. Eng.*, 23(4/5):667–682, 1999. → pages 154

[101] S. Skogestad and I. Postlethwaite. *Multivariable Feedback Control*. John Wiley and Sons Ltd., 1996. ISBN 0-471-94330-4. → pages 160

Appendix A

Virtual Process Code

A.1 Perl Script

```
#!/usr/bin/perl
use processtagexrefine2D;
use processfiles2D;
use File::Copy;
use File::Path;
use File::Glob;

my $ENDTIMESTEPTIME=0.1;$STEP3=130;$STEP5=30;
my $OPENTIME=50;$OPENTIMESTEPTIME=2;
my $CASTTEMP=700;$DIETEMP=400;
my $HCOEFFICIENT=50;
my $HCVARIABLE=0;
my $PERLMARK=0;
my $MARK1=0;
my $DC11=0;$DC12=0;
my $DC21=301;$DC22=300;
my $DC31=301;$DC32=300;
my $DC41=301;$DC42=300;


($ncycle,$startcyclenum)=@ARGV;

#————Create Directories to store the Log file and Data Files
 print "Creating Data Directories\n";
mkpath("/tmp/MPCOctCase215inletT/output");
mkpath("/data2/xinmei/MPCOctCase215inletT");
```

```
#———simpath is where to copy the ABAQUS model and simulate.
#datapath is where the simulation definition files are.
$simpath=”/tmp/MPCOctCase215inletT/”;
$datapath=”/data/xinmei/test/”;

#———Change the directory to the specified data path
chdir($datapath);

#Open a New Log file
open(CYCLOG, ”>cycleMPCOctCase215inletT.log”);
open(DATAGET, ”>DATAMPCOctCase215inletT.log”);
print DATAGET ”num,9 pvs,dieclosetime,dc1–Td,metaltemperature,dieopentime,volumemin,
volumemax,starttime,endtime,totalsolidtime\n”;


#Copy the model files to the simpath
print ”Copying Model Files\n”;
@modelfiles = glob(’model2Dof3DJune2bottom/∗’);
foreach (@modelfiles) {
        print CYCLOG ”$_\n”;
        copy(”$_”,”$simpath”);
}
        copy(”$datapath/test1”,”$simpath”);

#All the work is now done in the simpath
chdir($simpath);

#———Start Cycling through the Simulation
for (my $i = $startcyclenum; $i <= $ncycle; $i++) {
        print CYCLOG ”∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗∗\n”;
        print CYCLOG ”Start casting cycle #$i of $ncycle\n”;
          print ”Start casting cycle #$i of $ncycle\n”;

#———————————additional 5 step variables setting————————
    ($STEP3,$STEP5,$DC11)=(140,10,20); # die closetime1, die closetime2, DC1–Ts.
    # ————————uf1,uf2 setting————————————
                        if ($i == 11){
                                $OPENTIME=50+30;
                        }elsif ($i == 20){
                               $OPENTIME=50+20;
                        }else{
                               $OPENTIME=50;
                        }

                        if ($i <= 3){
                          $CASTTEMP=700;
                    }elsif ($i <= 19){
                         $CASTTEMP=680+($i-4)*2;
                    }elsif ($i <= 31){
                         $CASTTEMP=680+($i-20)*2;
                    }else{
                         $CASTTEMP=700;
                    }

#-------get parameters from different input files-------------------------
     #--Get Simulation Mode Parameters
     @SIMULATIONMODE = getfileparams("$datapath/input/simulationmode.input", $i);

#-------Get Setpoints
     my @SETPOINTS = getfileparams("$datapath/input/setpoints9.input", $i);

#-------extract last step number and increment number from previous.sta
      $signaltemp2=generatetempdef();

#-------Get Process Variables and choke volume
     @PROCESSVARS=getprocessvars();
       @volume=getchokevolume();

#-------Get Control Variables
     @CONTROLVARS = getfileparams("$datapath/input/controlvariables.input", $i);

#-------Get Feedforward Variables
     my @FEEDFORWARDS;
   $FEEDFORWARDS[0]=$CASTTEMP-700;
   $FEEDFORWARDS[1]=$OPENTIME-50;

       my @PROCESSVARS1=@PROCESSVARS[1,2,3,4,5,6,7,8,9];
#-------Writes the variables to the communication file.
      writetocommfile($i, $datapath,\@SIMULATIONMODE, \@PROCESSVARS1, \@SETPOINTS,
                \@CONTROLVARS, \@FEEDFORWARDS);


      print CYCLOG "the results from getpvs\n";
      print CYCLOG "$PROCESSVARS[1],$PROCESSVARS[2],$PROCESSVARS[3],$PROCESSVARS[4]\n";
      print CYCLOG "$PROCESSVARS[5],$PROCESSVARS[6],$PROCESSVARS[7],$PROCESSVARS[8],
              $PROCESSVARS[9]\n";
      print CYCLOG "$PROCESSVARS[10],$PROCESSVARS[11],$PROCESSVARS[12]\n";
      print CYCLOG "the results from choke\n";
      print CYCLOG "$volume[1],$volume[2],$volume[3],$volume[4],$volume[5]\n";
      print CYCLOG "the results from simulationmode,setpoint,controlvars,feedforward\n";
      print CYCLOG "@SIMULATIONMODE\n";
      print CYCLOG "@SETPOINTS\n";
```

```
        print CYCLOG ”@CONTROLVARS\n”;
        print CYCLOG ”$FEEDFORWARDS[0],$FEEDFORWARDS[1]\n”;


#-----Read the CV’s from the controller if it is closed loop
        if ($SIMULATIONMODE[0] eq ”1”) {
                @CONTROLVARS = getcontrolvars();
        print CYCLOG ”getcontrolvars:\n”;
        print CYCLOG ”@CONTROLVARS\n”;
        }

#-----Writes the variables to their respective output files.
        writetooutputfiles($i, $datapath, \@SIMULATIONMODE, \@PROCESSVARS, \@SETPOINTS,
                        \@CONTROLVARS, \@FEEDFORWARDS, \@DISTURBANCE);

#----------------#Process the Tag File--------------------------
#-----$DC12 is DC1-Td; $DC31 is DC3-Ts; $DC32 is DC3-Td.

       $STEP3=@CONTROLVARS[0]+150;
           $DC12=@CONTROLVARS[1]+20+10;

       my $return= &compileinputfile($i,$CASTTEMP,$DIETEMP,$STEP3,$STEP5,$OPENTIME,
          $OPENTIMESTEPTIME,$ENDTIMESTEPTIME);
       my $return1= &compilemainfile($DC11,$DC12,$DC21,$DC22,$DC31,$DC32,$DC41,$DC42);


#-----Run the Abaqus simulation
   $abaout=‘/opt/abaqus/64bit/Commands/abq675 j=sim inp=cycle user=main interactive‘;

#-----Save the .fil and .sta files for starting up the next simulation cycle
        copy(”sim.fil”,”previous.fil”);
            copy(”sim.sta”,”previous.sta”);

        #Save the .fil and .odb files as our data files
        ‘gzip sim.fil‘;
          copy(”sim.fil.gz”,”output/cycle_$i.fil.gz”);
       ‘gzip sim.odb‘;
      copy(”sim.odb.gz”,”output/cycle_$i.odb.gz”);
       ‘gzip sim.sta‘;
          copy(”sim.sta.gz”,”output/cycle_$i.sta.gz”);

        #Delete all of the files from the simulation that we don’t need anymore
        unlink(<sim.*>);

#--------------------

        print CYCLOG $abaout;
        print CYCLOG "Complete casting cycle #$i of $ncycle\n";
        print CYCLOG "*********************\n\n";
        print DATAGET "$i,$PROCESSVARS[1],$PROCESSVARS[2],$PROCESSVARS[3],$PROCESSVARS[4],
    $PROCESSVARS[5],$PROCESSVARS[6],$PROCESSVARS[7],$PROCESSVARS[8],$PROCESSVARS[9],
    $STEP3,$STEP5,$CASTTEMP,$OPENTIME,$DC11,$DC12,$volume[1],$volume[2],$volume[3],
    $volume[4],$volume[5]\n";
}


print CYCLOG "execution of programme is done\n";
close(CYCLOG);
close(DATAGET);

#———Copy the model files to the simpath
print "Copying Result Files to home directory\n";
@modelfiles = glob('output/*');
foreach (@modelfiles) {
        copy("$_","/data2/xinmei/MPCOctCase215inletT/");
}
copy("cycle_x.tag","/data2/xinmei/MPCOctCase215inletT/");
copy("cycle.inp","/data2/xinmei/MPCOctCase215inletT/");
copy("main_x.tag","/data2/xinmei/MPCOctCase215inletT/");
copy("main.f","/data2/xinmei/MPCOctCase215inletT/");
copy("test1","/data2/xinmei/MPCOctCase215inletT/");
copy("$datapath/DATAMPCOctCase215inletT.log","/data2/xinmei/MPCOctCase215inletT/");

#———delete all the temp files
   chdir("/tmp/");
  `rm -rf MPCOctCase215inletT`;
```

A.2 processtagexrefine2D.pm

```
package processtagexrefine2D;
require Exporter;
@ISA = qw(Exporter);
@EXPORT = qw(compileinputfile);
#----------------------------------------------------------------------
#Compiles the Input file from the .tag file for Abaqus
sub compileinputfile {
   my ($startnumber,$CASTTEMP,$DIETEMP,$STEP3,$STEP5,$OPENTIME,
       $OPENTIMESTEPTIME,$ENDTIMESTEPTIME) = @_;

   open(TAGFILE, "cycle_x.tag");
   open(WRTINP, ">cycle.inp");

   while (<TAGFILE>) {
       if(/<CASTDIETEMPS>/) {
          if ($startnumber!=1){
                        print "startnumber is $startnumber\n";
                                        my $step;
                                        my $inc;
                print open(STAFILE, "previous.sta");
                              while(<STAFILE>) {
                           @STADATA = /(\d+)\D+(\d+)\D+/;
                        if (length($STADATA[0]) > 0) {
                                    $step = $STADATA[0];
                      $inc = $STADATA[1];
                                                }
                                     }
                                   close(STAFILE);

        print WRTINP "*INITIAL CONDITIONS, TYPE=TEMPERATURE, FILE=previous,
                      STEP=$step, INC=$inc\n";
                print WRTINP "*INITIAL CONDITIONS, TYPE=TEMPERATURE\n";
           print WRTINP "NCAST, $CASTTEMP\n";
           print WRTINP "NSPRUE, $CASTTEMP\n";

           }else{
                print WRTINP "*INITIAL CONDITIONS, TYPE=TEMPERATURE\n";
                print WRTINP "NCAST, $CASTTEMP\n";
                print WRTINP "NSPRUE, $CASTTEMP\n";
                print WRTINP "NDIE, $DIETEMP\n";
                }
       }elsif(/<DIECLOSEDTIME1>/) {
                         print WRTINP "0.1, $STEP3, 0.000001, $ENDTIMESTEPTIME\n";
       }elsif(/<DIECLOSEDTIME2>/) {
```

```
                        print WRTINP ”0.1, $STEP5, 0.000001, 2\n”;
         }elsif(/<DIEOPENTIME>/) {
                        print WRTINP ”0.1, $OPENTIME, 0.000001, $OPENTIMESTEPTIME\n”;
         }elsif(/<BOUNDARYCASTTEMP>/) {
                        print WRTINP ”NCAST,    11,,         $CASTTEMP\n”;
                        print WRTINP ”NSPRUE,    11,,         $CASTTEMP\n”;
         }elsif(/<SIMPLEBDFORSPRUE>/) {
                        print WRTINP ”0.0,$CASTTEMP,34.0,600.0\n”;
         }else{
                        print WRTINP $_;
              }
        }

    close(TAGFILE);
    close(WRTINP);
    return 1
}
```

A.3 processfiles2D.pm

```
package processfiles2D;
require Exporter;
@ISA = qw(Exporter);
@EXPORT = qw(compilemainfile getprocessvars getchokevolume generatetempdef
getfileparams getfileparamscomma writefileparams getcontrolvars
writetocommfile writetooutputfiles);
#Compiles the Input file from the .tag file for Abaqus
sub compilemainfile {
        my ($D11,$D12,$D21,$D22,$D31,$D32,$D41,$D42) = @_;

   open(TAGFILE, ”main_x.tag”);
   open(WRTINP, ”>main.f”);

   while (<TAGFILE>) {
       if(/<HEATCOEFFICIENT1>/) {
                    print WRTINP ”IF ((TIME(2).GE.$D11).AND.(TIME(2).LE.$D12)) THEN\n”;
       }elsif(/<HEATCOEFFICIENT2>/) {
                    print WRTINP ”IF ((TIME(2).GE.$D21).AND.(TIME(2).LE.$D22)) THEN\n”;
       }elsif(/<HEATCOEFFICIENT3>/) {
                    print WRTINP ”IF ((TIME(2).GE.$D31).AND.(TIME(2).LE.$D32)) THEN\n”;
       }elsif(/<HEATCOEFFICIENT4>/) {
                    print WRTINP ”IF ((TIME(2).GE.$D41).AND.(TIME(2).LE.$D42)) THEN\n”;
       }else{
                   print WRTINP $_;
       }
   }

   close(TAGFILE);
   close(WRTINP);
   return 1
}

#———Gets the process variables
       #———add getpvs, if no path, means the same folder as test1
sub getprocessvars {
        #Execute getpvs.exe to extract the pvs from the previous.fil file
        $_=‘/opt/abaqus/64bit/Commands/abq675 /data/xinmei/test/getpvs2d‘;

 my     @RETURNDATA = split /\s+/;

        #We don’t want to return the 0 element because it is an empty string.
        #This results from the way the string returned from getpvs.exe is structured.
        return @RETURNDATA[0..$#RETURNDATA];
}
```

```
#———Gets the 2 choke volumes
sub getchokevolume {
        #Execute getpvs.exe to extract the pvs from the previous.fil file
       $_=‘/opt/abaqus/64bit/Commands/abq675 /data/xinmei/test/choke2d‘;

        my @RETURNDATA = split /\s+/;

        return @RETURNDATA[0..$#RETURNDATA];
}

#———generate temp.def file
sub generatetempdef {
   open(TEMPDEF, ”>temp.def”);
                        my $step;
                        my $inc;
                open(STAFILE, ”previous.sta”);
              while(<STAFILE>) {
                   @STADATA = /(\d+)\D+(\d+)\D+/;
                if (length($STADATA[0]) > 0) {
                    $step = $STADATA[0];
              $inc = $STADATA[1];
                                }
                     }
                   close(STAFILE);
                   print TEMPDEF ”$step, $inc\n”;
   close(TEMPDEF);
   return 1;
}

#———generate temp.def file
sub CopyandDel {
        copy(”sim.fil”,”previous.fil”);
        copy(”sim.sta”,”previous.sta”);

        #Save the .fil and .odb files as our data files
        ‘gzip sim.fil‘;
          copy(”sim.fil.gz”,”output/cycle_$i.fil.gz”);
       ‘gzip sim.odb‘;
        copy(”sim.odb.gz”,”output/cycle_$i.odb.gz”);
       ‘gzip sim.sta‘;
        copy(”sim.sta.gz”,”output/cycle_$i.sta.gz”);

#———Delete all of the files from the simulation that we don’t need anymore
        unlink(<sim.*>);
   return 1;
}
```

```
#———This subroutine requires the following parameters
#1. Input Parameter File Name $filename
#2. The Current Cycle Number $cyclenum

#———Simulation Parameters from the specified file
sub getfileparams {
          my ($filename, $cyclenum) = @_;

          my @RETURNDATA;

          open(PARAMFILE, $filename);

        while (<PARAMFILE>) {
          my @PARAMDATA = split /\s+/;

                    if (@PARAMDATA[0] <= $cyclenum) {
                              @RETURNDATA = @PARAMDATA[1..$#PARAMDATA];
                    }
        }

    close(PARAMFILE);
    return @RETURNDATA;
}

#———This subroutine requires the following parameters
#1. Input Parameter File Name $filename
#2. The Current Cycle Number $cyclenum

#———Simulation Parameters from the specified file
sub getfileparamscomma {
          my ($filename, $cyclenum) = @_;

          my @RETURNDATA;

          open(PARAMFILE, $filename);

        while (<PARAMFILE>) {
          my @PARAMDATA = split /,+/;

                    if (@PARAMDATA[0] <= $cyclenum) {
                              @RETURNDATA = @PARAMDATA[1..$#PARAMDATA];
                    }
        }

    close(PARAMFILE);
    return @RETURNDATA;
```

```
}

#———This subroutine requires the following parameters
#1. Output Parameter File Name $filename
#2. The Current Cycle Number $cyclenum
#3. The parameters to be written to the file

#———Simulation Parameters from the specified file
sub writefileparams {
        my ($filename, $cyclenum, @PARAMDATA) = @_;

        my $nextline = $cyclenum;
        foreach (@PARAMDATA) {
        $nextline = $nextline.” ”.$_;
        }

      open(PARAMFILE, ”>>$filename”);
        print PARAMFILE ”$nextline\n”;
      close(PARAMFILE);
}


#———Gets the controller variables based on the Controller Mode to the BWDD
sub getcontrolvars {
        my @RETURNDATA;
        my $done = 0;
        my $cvsready;

        until ($done) {
                ’touch –c /data/xinmei/test/simulation1.comm’;
                open(COMMFILE, ”/data/xinmei/test/simulation1.comm”);
                $cvsready = <COMMFILE>;

                        #If the CVS are ready then just assign return data to every line
                        #as the last line is the Control variables.

                        if (($cvsready == ”1”) || (1*$cvsready ==1)) {
                                $done = 1;
                        while (<COMMFILE>) {
                                @RETURNDATA = split /\s+/;
                        }
                        }
                close(COMMFILE);
            sleep 1;
        }

        return @RETURNDATA;
```

```
}

#———The format of the COMM file is:
#CVS Ready
#CycleNum
#Simulation Mode
#PVS
#SPS
#FFS
#CVS
sub writetocommfile {
        my ($cyclenum, $datapath,$SIMULATIONMODE, $PROCESSVARS,
                $SETPOINTS, $CONTROLVARS, $FEEDFORWARDS) = @_;
#The Arrays are passed as references and hence must be reassigned to the following variables.
        @MODE = @$SIMULATIONMODE;
        @PVS = @$PROCESSVARS;
        @SPS = @$SETPOINTS;
        @CVS = @$CONTROLVARS;
        @FFS = @$FEEDFORWARDS;

                ’touch −c /data/xinmei/test/simulation1.comm’;
      open(COMMFILE, ”>$datapath/simulation1.comm”);

#This is the Control Variables Ready Bit.
#Used when retrieving the Control Variables from the Controller.
        #The controller must set this to 1 when the CV’s are ready.
        print COMMFILE ”0\n”;

        print COMMFILE ”$cyclenum\n”;

        foreach (@MODE) {
        print COMMFILE $_.” ”;
        }
        print COMMFILE ”\n”;

        foreach (@PVS) {
        print COMMFILE $_.” ”;
        }
        print COMMFILE ”\n”;

        foreach (@SPS) {
        print COMMFILE $_.” ”;
        }
        print COMMFILE ”\n”;

        foreach (@FFS) {
        print COMMFILE $_.” ”;
        }
        print COMMFILE ”\n”;

        print COMMFILE ”1\n”;

        foreach (@CVS) {
        print COMMFILE $_.” ”;
        }
        print COMMFILE ”\n”;

      close(COMMFILE);


}

sub writetooutputfiles {
        my ($cyclenum, $datapath, $SIMULATIONMODE, $PROCESSVARS,
               $SETPOINTS, $CONTROLVARS, $FEEDFORWARDS, $DISTURBANCE) = @_;

#The Arrays are passed as references.
        @MODE = @$SIMULATIONMODE;
        @PVS = @$PROCESSVARS;
        @SPS = @$SETPOINTS;
        @CVS = @$CONTROLVARS;
        @FFS = @$FEEDFORWARDS;
        @DIST = @$DISTURBANCE;

        writefileparams(”$datapath/output/simmode.output”, $cyclenum, @MODE);
        writefileparams(”$datapath/output/processvariables.output”, $cyclenum, @PVS);
        writefileparams(”$datapath/output/setpoints.output”, $cyclenum, @SPS);
        writefileparams(”$datapath/output/controlvariables.output”, $cyclenum, @CVS);
        writefileparams(”$datapath/output/feedforwards.output”, $cyclenum, @FFS);
        writefileparams(”$datapath/output/disturbances.output”, $cyclenum, @DIST);
}
```

A.4 simulation1.comm

```
1
17
1
499.997  450.173  466.204  510.672  542.953  531.433  530.962  565.298  562.392
494.634  443.000  453.357  508.235  542.530  531.458  531.161  565.087  561.874
6  0
1
33.9272  −3.5261
```

Appendix B

MATLAB Code

B.1 MPC_controller_deltaU.m

```
%global software_path;
 clc;
 clear;
 clear global;
 close all;
 global A1 A2 A3 A4 A5 A6 A7 A8 A9 betaSI alfa1 beta1 alfaano1 betaano1 gamaano1
 global A B Cf D num_dist num_cvs uf uv xcurrent kk Hp Hu Q R DataRef yerr%delaysum
 %=================1. MPC ini setting=========================================
% delay is [1 0 0 1];
%--------A,B,C,D setting-------------------------------
 load Model2Dof3DdataAug9.mat;
%-------------augmented model with delay involved-------------
Au=[0 0 0 0 0 0
    1-a11 a11 0 0 0 0
    1-a22 0 a22 0 0 0
    0 0 0 a33 0 0
    0 0 0 0 a44 0
    0 0 0 0 0 a55];
Bu=[1 0
    0 0
    0 0
    0 1-a33
    0 1-a44
    0 1-a55];
  Af=[Af Bf(:,2)
       0 0 0];
```

```
   Bf=[Bf(:,1) [0;0]
           0 1];
   A=diagmx(Au,Af);
   B=[Bu zeros(6,2);zeros(3,2) Bf];
  Cf=[Cf zeros(9,1)];
  D=zeros(9,4);
%——————parameters setting——————
Hu=2;Hp=15;
num_dist=2; num_cvs=2;
Q=1*2500*eye(1);R=0*eye(1); % case1: focus on minimized vol.

  u_lim=[-40,40;
          -9.9,30;];
   deltau_lim=[10;
                5];
baseline=[150,10];
%—————inequilities Aine,bine setting——————————
  [Aineadd,bineadd]=funABineadd(Hu,deltau_lim);
  Aine=Aineadd;
  bine=bineadd;
%———————equility Aeq,Beq setting——————————
 Aeq=[];beq=[];
 vlb=[];vub=[];
    vlbtemp=u_lim(:,1);vubtemp=u_lim(:,2);
   for i=1:Hu;
      vlb=[vlb;vlbtemp]; vub=[vub;vubtemp];
   end


%=============2.running preparation===============
sim_path = '/data/xinmei/test/';
cycle = 0; % cycle # in matlab, used to compare with real cycle# from perl script.
Cvompute_Act = 0; % activating signal for cv computation.
cv_ready=0; % judge if cv is ready to be sent to perl for merging into inp file.
matlab_run=1; % control matlab running.
totaltime=200;
%—————ref baseline:ubaseline=[150,10].
 ubase=[150;10];
 uv=zeros(totaltime,2);uf=zeros(totaltime,2);
 ymea=zeros(totaltime,9);ypre=zeros(totaltime,9);
 xcurrent=zeros(9,totaltime); x0=zeros(2*Hu,1);
 Uvpredict=zeros(num_cvs*Hu,1);% set initial Uv10,Uv20.
%————read last time xstate value from xstate.def————————
    FID=fopen([sim_path,'xstate.def']);
    if (FID > 0)
        fclose(FID);
          para = dlmread([sim_path,'xstate.def']);
```

```
         lastNum=para(1);
      xcurrent(:,lastNum)=para(2:10);
    else
       disp('xstate.def file cannot be found');
    end

%=============2.loop=============
while(matlab_run==1)

        FID=fopen([sim_path,'simulation1.comm']);
    if (FID <= 0)
       disp('simulation1.comm file cannot be found');
    else
         fclose(FID);
%-------------a. read the communications file -------------
          comm = dlmread([sim_path,'simulation1.comm']);
          CycleofPerl=comm(2,1);
          SimuMode=comm(3,1);
          pv=comm(4,1:9);
          sp=comm(5,1:9);
          ff1=comm(6,1);ff2=comm(6,2);
          matlab_run=comm(7,1);
%-------------b. compute the signal on whether to compute CV-----
      if ((SimuMode==1) & (cycle < CycleofPerl))
            Cvompute_Act=1;
            disp('New Cycle');cycle = CycleofPerl
      end
%-------------c. the compution of CV, i needs to be >2-------------
      if Cvompute_Act==1
                 kk=cycle;
                 ymea(kk,:)=pv-DataRef;  yerr=ymea(kk,:)-ypre(kk,:);
                 uf(kk,:)=[ff1 ff2];

          %%compute uv(i)
                    qx1=-30:10:10;  qy1=30:-10:-9.9;
                    [QX,QY]=meshgrid(qx1,qy1);
                  temp=size(QX);
                  Ntotal=temp(1)*temp(2);
                  QX=reshape(QX,Ntotal,1);  QY=reshape(QY,Ntotal,1);
                for j=1:Ntotal;
                  x0(1:2:end)=QX(j);x0(2:2:end)=QY(j); % initial value for predicted u.
                  [x(:,j),fval(j),exitflag,output]=fmincon(@funf,x0,Aine,bine,Aeq,beq,
                                                vlb,vub,'LPDCcon');
            end

         index=find(fval==min(fval));
         display(['run #=',num2str(kk)]);
```

```
             if mod(length(index),2)==1
                 Uvpredict=x(:,index((1+end)/2)); % x is predicted u
                 uv(kk,:)=x(1:num_cvs,index((1+end)/2))';
             else
             Uvpredict=x(:,index(end/2)); % x is predicted u
             uv(kk,:)=x(1:num_cvs,index(end/2))';
             end


                  bine=[bine(1:end-4);
                deltau_lim(1)+uv(kk,1);
                        deltau_lim(1)-uv(kk,1);
                deltau_lim(2)+uv(kk,2);
                deltau_lim(2)-uv(kk,2);];

         %%write uv(i) into simulation1.comm.
         MPC_write_comm(uv(kk,:),sim_path,'simulation1.comm');

         %%compute xcurrent(i+1),ypre(i+1)
         [ypre(kk+1,:),xcurrent(:,kk+1)]=Simulationmodel(xcurrent(:,kk),uv(kk,:),uf(kk,:));
         MPC_write_xstate(kk+1,xcurrent(:,kk+1),sim_path,'xstate.def','xstatestore.def');
         Cvompute_Act=0;
     end
   end
       pause(1);
end
 disp('finished running');
```

B.2 funf.m

```
% x is the vector, involving u(k|k)...u(k+Hu-1|k). total number is hu*2.
%x(2*n-1),x(2*n) stand for nth u vectors DC1-Td and DC3-Ts.

function f=funf(x);
global A1 A2 A3 A4 A5 A6 A7 A8 A9 betaSI alfa1 beta1 alfaano1 betaano1 gamaano1
global A B Cf D num_dist num_cvs uf uv xcurrent kk Hp Hu Q R DataRef yerr%delaysum

[num_rows_B,num_cols_B] = size(B);
B_inputs = B(:,1:num_cols_B-num_dist);
B_dist = B(:,num_cols_B-num_dist+1:end);

[Psi,Upsilon,Theta,Xi,Xiuv,C_matrix,Q_matrix,R_matrix] =
                                      get_mpc_matricesXinmei(Hu,Hp,A,B_inputs,B_dist,Cf,D,Q,R);

SQ = sqrt(Q_matrix);
SR = sqrt(R_matrix);


uv_predicted=x; %%% new
for i=(Hu+1):Hp
     uvHu=x(end-num_cvs+1:end);
     uv_predicted=[uv_predicted;uvHu];
end

dist_predicted=[];
dist=uf(kk,:)';
for i = 1:Hp
     dist_predicted = [dist_predicted;dist];
     dist(2,1) = 0;
end

%------------XX(k) compution------------
XX=Psi*xcurrent(:,kk) + Xiuv*uv_predicted+Xi*dist_predicted;
%------------YY(k) compution------------
   for i=1:Hp
          temp1=XX(9*(i-1)+2,1);temp2=XX(9*(i-1)+3,1);
          temp4=XX(9*(i-1)+4,1);temp5=XX(9*(i-1)+5,1);temp6=XX(9*(i-1)+6,1);
            for j=1:length(alfa1);
               s=sprintf('AA=A%d;',j);
                eval(s);
               x1both(j,1)=alfa1(j)*temp1+beta1(j)*temp2;
               x2both(j,1)=alfaano1(j)*temp4+betaano1(j)*temp5+gamaano1(j)*temp6;
               YY1(j,1)=[1 x1both(j,1) x1both(j,1)^2]*AA...
                          *[1;x2both(j,1);x2both(j,1)^2;x2both(j,1)^3];
```

```
        end

      YY2=Cf*[XX(9*(i-1)+7:9*(i-1)+9,1)];
      YY=YY1+DataRef'+YY2+yerr'; %model output T(9,1) with constant output observer.
      VolPre(i)=[1 YY']*betaSI;
end

  %----------cost function format:------------------------------
  term1=VolPre;
  term2=uv_predicted([1:2:end])+150; % uv1 absolute values along Hp horizon.
  f=Q*term1*term1'+R*term2'*term2;
  %f=sum(term1);
```

B.3 funABineadd.m

```
function [Aineadd,bineadd]=funABineadd(Hu,deltau_lim);
uv1Array=[];uv2Array=[];
basicArray1=zeros(2,Hu*2);
basicArray1(:,1:3)=[1 0 -1;
                    -1 0 1];
%-----------------------------------------
    temp=basicArray1;
    for i=1:Hu-1;
        uv1Array=[uv1Array;temp];
        temp=[zeros(2,2) temp(:,1:end-2)];
    end
%-------------------------------------------
    temp=[zeros(2,1) basicArray1(:,1:end-1)];
    for i=1:Hu-1;
        uv2Array=[uv2Array;temp];
        temp=[zeros(2,2) temp(:,1:end-2)];
    end
%-------------------------------------------
temp=[1 0;-1 0;0 1;0 -1];
    lastArray=[temp zeros(4,Hu*2-2)];
%---------------------------------------
Aineadd=[uv1Array;uv2Array;lastArray];

%---------------------------------------
if Hu==1
   bineadd=[deltau_lim(1)*ones(2,1);
            deltau_lim(2)*ones(2,1);];
else
   bineadd=[deltau_lim(1)*ones(Hu*2-2,1);
          deltau_lim(2)*ones(Hu*2-2,1);
          deltau_lim(1)*ones(2,1);
          deltau_lim(2)*ones(2,1);];
end
```

B.4 get_mpc_matricesXinmei.m

```
function [Psi, Upsilon, Theta, Xi, Xiuv, C_matrix, Q_matrix, R_matrix] =
        get_mpc_matricesXinmei(num_control_moves, prediction_length ,A,B,Bd,C,D,Q,R);

Psi = [];
for i = 1:prediction_length
    Psi = [Psi;mpower(A,i)];
end

Upsilon = [];
for i = 1:prediction_length
    temp = zeros(size(B));
    for j = 1:i
        temp = temp + mpower(A,j-1)*B;
    end
    Upsilon = [Upsilon;temp];
end

[num_rows_B, num_columns_B] = size(B);

temp_Theta_column = Upsilon;

Theta = [temp_Theta_column];
for i = 2:num_control_moves
    temp_Theta_column = [zeros(num_rows_B, num_columns_B);
    temp_Theta_column(1:num_rows_B*(prediction_length -1),:)];
    Theta = [Theta, temp_Theta_column];
end

[num_rows_Bd, num_columns_Bd] = size(Bd);

temp_Xi_column = [];
temp_Xiuv_column = [];%% new

for i =1:prediction_length
    temp_Xi_column = [temp_Xi_column;mpower(A,i-1)*Bd];
    temp_Xiuv_column = [temp_Xiuv_column;mpower(A,i-1)*B];%% new
end

Xi = temp_Xi_column;
Xiuv = temp_Xiuv_column;%% new

for i = 2:prediction_length
    temp_Xi_column = [zeros(size(Bd));temp_Xi_column];
    Xi = [Xi, temp_Xi_column(1:length(Bd)*prediction_length ,:)];
    temp_Xiuv_column = [zeros(size(B));temp_Xiuv_column];  %% new
    Xiuv = [Xiuv,temp_Xiuv_column(1:length(B)*prediction_length,:)];%%  new
end

C_matrix = [];
[num_rows_C,num_columns_C] = size(C);

for i = 1:prediction_length
    temp = zeros(num_rows_C,num_columns_C*prediction_length);
    column_index = (1 + (i-1)*num_columns_C);
    temp(1:num_rows_C,column_index:column_index+num_columns_C-1) = C;
    C_matrix = [C_matrix;temp];
end

Q_matrix = zeros(prediction_length*num_rows_C,prediction_length*num_rows_C);

for i = 1:prediction_length
    index = 1 + (i-1)*length(Q);
    Q_matrix(index:index+length(Q)-1,index:index+length(Q)-1) = Q;
end

R_matrix = zeros(num_control_moves*num_columns_B,num_control_moves*num_columns_B);

for i = 1:num_control_moves
    index = 1 + (i-1)*length(R);
    R_matrix(index:index+length(R)-1,index:index+length(R)-1) = R;
end
```

B.5 LPDCcon.m

```
function [g,ceq]=LPDCcon(x)
  g=[];
  ceq=[];
```

B.6 Simulationmodel.m

```
function [yout,xupd]=Simulationmodel(xpre,uinput,udist);
global A1 A2 A3 A4 A5 A6 A7 A8 A9 alfa1 beta1 alfaano1 betaano1 gamaano1
global A B Cf num_dist

[num_rows_B,num_cols_B] = size(B);
B_inputs = B(:,1:num_cols_B-num_dist);
B_dist = B(:,num_cols_B-num_dist+1:end);

xupd=A*xpre+B_inputs*uinput'+B_dist*udist';

for j=1:length(alfa1);
          s=sprintf('AA=A%d;',j);
           eval(s);
          x1both(j,1)=alfa1(j)*xupd(2)+beta1(j)*xupd(3);
          x2both(j,1)=alfaano1(j)*xupd(4)+betaano1(j)*xupd(5)+gamaano1(j)*xupd(6);
          yout1(j,1)=[1 x1both(j,1) x1both(j,1)^2]*AA...
                 *[1;x2both(j,1);x2both(j,1)^2;x2both(j,1)^3];
end

yout=yout1+Cf*xupd(7:9);
```

B.7 MPC_write_comm.m

```
%  MPC_write_comm(uv(kk),sim_path,'simulation1.comm');
function MPC_write_comm(cvs,sim_path,filename)
%read each line in the communications file
FID=fopen([sim_path,filename],'r');

cvs_ready=fgets(FID);
cycle_num=fgets(FID);
sim_mode=fgets(FID);
pv_read=fgets(FID);
sp_read=fgets(FID);
ff_read=fgets(FID);
matlabrunning_read=fgets(FID);
cv_read=fgets(FID);
fclose(FID);

%rewrite the communications file with CVS Ready=1, and the new CVS

FID=fopen([sim_path,filename],'w+');

fprintf(FID,'1'); % write cvs ready
fprintf(FID,'\n');
fprintf(FID,cycle_num);
fprintf(FID,sim_mode);
fprintf(FID,pv_read);
fprintf(FID,sp_read);
fprintf(FID,ff_read);
fprintf(FID,matlabrunning_read);
fprintf(FID,num2str(cvs(1)));
fprintf(FID,'\t');
fprintf(FID,num2str(cvs(2)));
fprintf(FID,'\n');
fclose(FID);
```

B.8 MPC_write_xstate.m

```
 %MPC_write_xstate(kk+1,xcurrent(:,kk+1),sim_path,'xstate.def','xstatestore.def');
function MPC_write_xstate(cycleN,x,sim_path,filename1,filename2)
% write data into xstate.def.
fid=fopen([sim_path,filename1],'w+');
data = [cycleN x'];
fprintf(fid,'%3.0f %12.8f %12.8f %12.8f %12.8f %12.8f %12.8f %12.8f %12.8f %12.8f\n',data);
fclose(fid);
% write data into xstatestore.def.
fid=fopen([sim_path,filename2],'a');
data = [cycleN x'];
fprintf(fid,'%3.0f %12.8f %12.8f %12.8f %12.8f %12.8f %12.8f %12.8f %12.8f %12.8f\n',data);
fclose(fid);
```