FEDERAL UNIVERSITY OF SANTA CATARINA
TECHNOLOGY CENTER
CONTROL AND AUTOMATION ENGINEERING

Leonardo Pavan Rocha

Simulation-based optimization of a precision glass molding thermal model

Aachen
2020

Leonardo Pavan Rocha

Simulation-based optimization of a precision glass molding thermal model

Report was judged in the context of the discclipline **DAS5511: Final Project Work** and APPROVED in its final form by the Control and Automation Engineering Course.
Advisor: Prof. Dr. Rodolfo César Costa Flesch

Aachen
2020

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Rocha, Leonardo Pavan
Simulation-based optimization of a precision glass molding thermal model / Leonardo Pavan Rocha ; orientador, Rodolfo César Costa Flesch, 2020.
66 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro Tecnológico, Graduação em Engenharia de Controle e Automação, Florianópolis, 2020.

Inclui referências.

1. Engenharia de Controle e Automação. 2. Precision glass molding. 3. Parameter identification. 4. Simulation based optimization. I. Flesch, Rodolfo César Costa . II. Universidade Federal de Santa Catarina. Graduação em Engenharia de Controle e Automação. III. Título.

Leonardo Pavan Rocha

Simulation-based optimization of a precision glass molding thermal model

This monography was judged in the context of the discipline DAS5511: Final Work Project and approved on its final form by the following members of the examination board:

____

Gang Liu, Dr.-Ing.
Company Supervisor
Fraunhofer-Institut für Produktionstechnologie

____

Prof. Rodolfo César Costa Flesch, Dr.
University Advisor
Federal University of Santa Catarina

____

Prof. Fabio Luiz Baldissera, Dr.
Evaluator
Federal University of Santa Catarina

ACKNOWLEDGEMENTS

“If I have seen further it is by standing on the shoulders of giants”. With this Isaac Newton’s famous quote, I begin the most important part of this work. The most important because without the support from the following people, this work would not have been possible.

I would like to show my great gratitude towards Gang Liu and Jan-Helge Staasmeyer for the opportunity and the guidance during the development of this work. This gratitude is also extended to my colleagues at IPT, Stefanie Carvalho and Dominik Stollenwerk, who helped me during the experimental part of this work.

I also want to thank my teachers and professors that taught me not only theories, but also how to apply them in real-life problem solving situations. In this matter, I want to specially thank Professor Rodolfo César Costa Flesch, for the support and help during the development and writing of this work. In many times where I felt that the solutions would be too complex, he always provided me great advice and supporting words.

A special thanks to Henrique, Ivan, Igor, João Paulo and Otávio, who were my family in Aachen. The gatherings in Pontstraße and the parks of Aachen will always be remembered kindly.

At last, but not least, I want to thank my family, my friends and my girlfriend, for always providing me full support, believing in my dreams and proving that closeness has nothing to do with distance.

These are the giants whose shoulders I have been lucky to stand on, and for that I am forever grateful.

RESUMO

Historicamente, o processo convencional de produção de lentes consistia na moagem e no polimento de vidro de forma a atingir os requisitos de projeto. No entanto, o esforço de fabricação de lentes por meio do processo convecional aumenta significativamente com a complexidade de projeto das lentes. Devido à grande demanda do mercado por lentes com cada vez mais alta precisão, novos processos de manufatura tornaram-se mais relevantes.

A moldagem de precisão de vidro emergiu como um desses processos, oferecendo uma alternativa economicamente viável para a produção em massa de componentes ópticos de alta precisão. O processo consiste em uma prensa inserida em um ambiente com temperatura controlada, onde um molde é utilizado para moldar o vidro sujeito a condições específicas de temperatura e pressão. Após o vidro ser moldado, o mesmo passa por uma etapa de resfriamento. O resultado do processo é um componente óptico pronto para ser utilizado.

Embora o processo ofereça muitas vantagens, alguns desafios devem ser abordados, como vazão imprevista de vidro para fora do molde e o encolhimento da lente dentro do molde, que podem resultar em má qualidade óptica. As soluções *ad-hoc* para esses problemas consistem no projeto repetitivo das lentes e na manufatura iterativa de moldes, representando, portanto, um alto custo tanto financeiro quanto de mão-de-obra. A alternativa para resolver essas questões de forma menos custosa consiste em utilizar simulação por elementos finitos para melhor entender como o vidro se comporta durante o complexo processo de moldagem.

A simulação atual do processo consiste de um modelo mecânico e um modelo térmico. O primeiro permite realizar uma análise de tensão-deformação do vidro, enquanto o segundo permite avaliar a distribuição de temperatura dentro da máquina. O modelo térmico envolve um conjunto the fenômenos termodinâmicos complexos que ocorrem simultaneamente. Alguns desses fenômenos podem ser modelados matematicamente, no entanto com parâmetros desconhecidos. A fim de permitir a otimização do processo, a modelagem correta desses fenômenos térmicos é necessária.

Este relatório aborda a identificação de parâmetros térmicos para a simulação do processo de moldagem de precisão de vidro utilizando otimização baseada em simulação juntamente com dados experimentais. A arquitetura proposta para identificação desses parâmetros consiste em integrar o *software* de simulação com um módulo de otimização, responsável por alterar os arquivos de simulação com os parâmetros a serem testados. Primeiramente, um plano de experimentos foi feitos de forma a identificar os parâmetros desconhecidos do processo. Foram definidos experimentos de resfriamento à vácuo, resfriamento por nitrogênio, e experimentos de aquecimento.

Após isso, utilizou-se a simulação atual como base para o desenvolvimento de uma nova simulação que descreve os experimentos feitos. De forma a integrar a simulação com o módulo de otimização, um projeto de *software* foi realizado utilizando a metodologia de desenvolvimento baseado em funcionalidades. O resultado foi um

script na linguagem *Python* responsável por integrar o pacote de otimização *Nevergrad* com o software de simulação *Abaqus*.

Como existem múltiplos algoritmos de otimização que podem ser utilizados nesse caso, foi proposta uma comparação entre múltiplos algoritmos de otimização sem utilização de derivada, visto que a simulação não computa esses valores. Após definir um algoritmo a ser utilizado para os experimentos, a identificação dos parâmetros foi feita utilizando diferentes hipóteses de modelagem de cada experimento.

Para os experimentos de resfriamento à vácuo, os fenômenos térmicos existentes são: perda de calor por condução pelas flanges, perda de calor por radiação para o ambiente, e condução na interface entre as placas de resfriamento e o molde. Analisou-se duas hipóteses de modelagem a partir disso. A primeira buscou identificar apenas o coeficiente de condução pelas flanges e a emissividade de radiação para o ambiente, assumindo que a condutância na interface mantinha-se constante. A segunda hipótese assumia que a condutância na interface também deveria ser identificada. A segunda hipótese foi capaz de explicar os dados experimentais com baixo erro.

Os experimentos de resfriamento por nitrogênio apresentam os seguintes fenômenos térmicos: perda de calor por condução/convecção, perda de calor por radiação para o ambiente, convecção na lateral do molde, e condução na interface. Múltiplas hipóteses foram analisadas, no entanto apenas uma foi capaz de explicar bem os dados experimentais. A mesma assumiu que o coeficiente de convecção superior e inferior, o coeficiente de convecção lateral e a condutância na interface deveriam ser identificadas. Além disso, múltiplos pontos de operação (vazão de nitrogênio) foram analisados e tiveram seus respectivos coeficientes identificados utilizando a mesma hipótese de modelagem. Posteriormente uma análise de regressão foi feita de forma a encontrar uma função que descreva como as variáveis identificadas variam com o ponto de operação.

A partir dos parâmetros identificados e da análise de regressão, uma avaliação experimental do método foi feita tanto em pontos de operação conhecidos quanto desconhecidos. A avaliação em um ponto de operação conhecido consistiu em repetir um experimento de resfriamento por nitrogênio e utilizar os parâmetros identificados previamente para verificar se a simulação ainda era capaz de explicar os dados do novo experimento. A avaliação para o ponto de operação de 20 l/min retornou bons resultados.

A avaliação experimental do método em pontos de operação desconhecidos também foi feita. A partir dos modelos de regressão, os parâmetros térmicos para uma vazão de 30 l/min foram obtidos e avaliados. Os parâmetros obtidos geraram uma curva de simulação que não teve boa concordância com a curva experimental. Como a análise de regressão havia sido feita com poucas observações, assumiu-se que as curvas ajustadas foram afetadas por ruído e podiam não ser estatisticamente significativas. Dessa forma, o processo de identificação foi feito próximo à região apontada pela regressão e atingiu bons resultados. Isso mostrou que a regressão estava direcionando os parâmetros para a região correta, no entanto, para o ponto errado. A partir disso,

concluiu-se que mais observações são necessárias para que a regressão seja efetiva.

Por fim, a abordagem sugerida provou ser viável e apresentou bons resultados para experimentos de resfriamento à vácuo e resfriamento por nitrogênio. Os resultados estão de acordo com a teoria e os dados experimentais. A condução na interface identificada mostrou concordância com artigos que abordam o mesmo tópico, porém utilizando diferentes abordagens de identificação. Futuros trabalhos envolvem realizar a identificação dos modelos relacionados aos experimentos de aquecimento, e a avaliação experimental das funções que descrevem as variáveis identificadas nos experimentos de resfriamento por nitrogênio nos pontos de operação restantes.

Palavras-chave: Moldagem de precisão de vidro. Identificação de parâmetros. Otimização baseada em simulação.

ABSTRACT

Precision glass molding emerged as a powerful and economically viable alternative for the mass-production of high-precision optical components. Even though the process offers many advantages, some challenges must be addressed, such as unpredicted glass flow in the machine and shrinkage of the lens inside the mold. In order to solve these problems, finite element simulation is used. The current simulation of precision glass molding consists of mechanical and thermal models. The thermal simulation involves a set of complex thermodynamic phenomena which take place simultaneously. Some of the phenomena can be mathematically modelled, but the coefficients of the model are not trivial to be obtained. In order to allow process optimization, the correct modeling of these thermal phenomena is required. This report addresses the identification of thermal parameters for the precision glass molding simulation using simulation-based optimization together with experimental data. To accomplish this, a study of the thermal model, development of a FEM simulation and a software project to integrate the simulation software with the optimization architecture, as well as the analysis of the identification approach for two cooling scenarios (heat losses to the environment and nitrogen cooling) were carried out. The results proved that the proposed approach is feasible, the identified parameters agree with theory and experimental data. The identified gap conductance for vacuum experiments showed an agreement with papers addressing the same topic, however using different identification approaches.

Keywords: Precision glass molding. Parameter identification. Simulation-based optimization.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

API	Application Programming Interface
BO	Bayesian Optimization
CAGR	Compound Anual Growth Rate
CFD	Computational Fluid Dynamics
CNC	Computer Numerical Control
FDD	Feature-Driven Development
FEA	Finite Element Analysis
FEM	Finite Element Method
HTTP	HyperText Transfer Protocol
IPT	Institute for Production Technology
IT	Information Technology
MAE	Mean Absolute Error
MSE	Mean Squared Error
PGM	Precision Glass Molding
PID	Proportional-Integral-Derivative
PLC	Programmable Logic Controller
PSO	Particle Swarm Optimization
RMSE	Root Mean Squared Error
UML	Unified Modeling Language

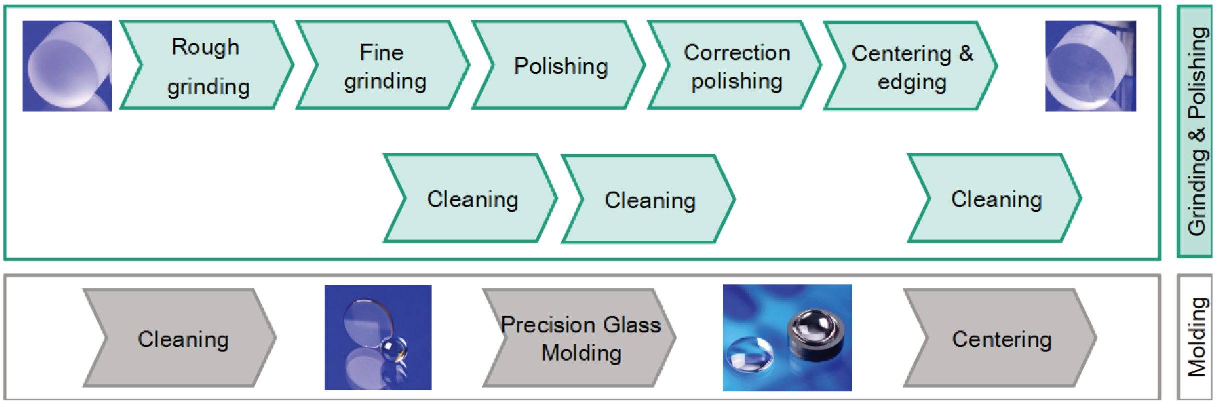
CONTENTS

1 INTRODUCTION

Optical components have been used by society for several millennia, from reading stones, to glasses, microscopes, telescopes, cameras and endoscopes (WINDHOEK, 2015). The idea of "bending" light has always been an important vector for the development of new technologies. 24chemicalresearch (2019) states that the global market for precision glass molding has an expected Compound Anual Growth Rate (CAGR) growth of 33% by 2025. An increasing demand for more complex optical components offering higher precision and lower costs is the cause of this expected growth.

Historically, the conventional process for production of lenses consisted of grinding and polishing glass (YAN et al., 2009). However, following the market demands, new manufacturing processes have become more relevant for the production of complex optical components. PGM emerged as a powerful and economically viable alternative to conventional processes. This is due especially to the fact that PGM relies on a small process chain compared to the conventional production process chain. Figure 1 compares the two process chains.

Figure 1 – Process chain comparison between conventional glass lens manufacturing and precision glass molding.

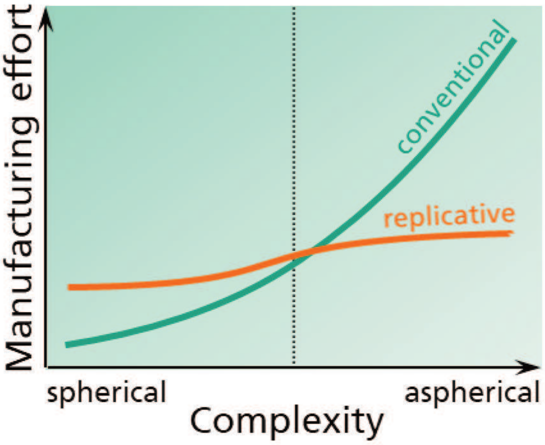


Source – Fraunhofer IPT

Also, in the conventional manufacturing of lenses, the grinding and polishing steps are performed in each lens. If the shape of the designed lens is complex, these steps take great effort. As for PGM, the complexity of the lens depends completely on the mold shape, which can be used to produce many lenses. Therefore, for mass-production of high-precision optical components, the replicative approach offered by PGM reduces significantly the manufacturing effort, as shown in Figure 2, therefore reducing the total costs.

Even though PGM offers many advantages, some challenges must be addressed, such as the shrinkage of the lens inside the mold or the unpredicted glass flow in the machine, what results in bad quality of imaging. The ad-hoc solutions for these problems,

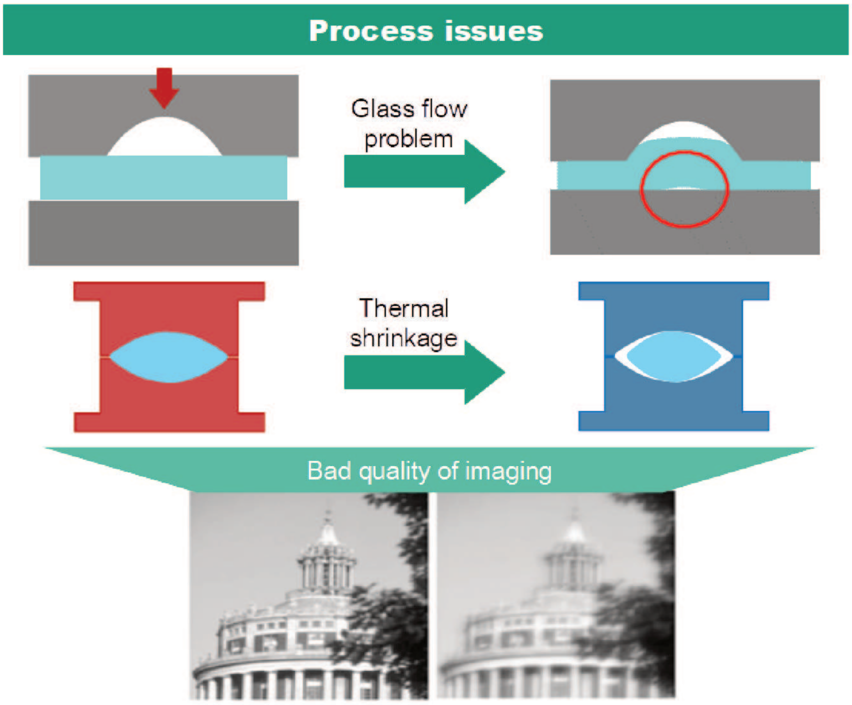
Figure 2 – Comparison between complexity of lens design with its manufacturing effort in replicative and conventional lens production processes.



Source – Fraunhofer IPT

depicted in Figure 3, are the repetitive mould design and iterative mold manufacturing. However these solutions are cost and labor intensive. Since the molding process cannot be directly measured - that is, the state of the glass during the process is not observable - FEM simulation provides a manner of understanding the complex molding process, hence offering universal guidelines for RD activities (LIU, 2018).

Figure 3 – Challenges for PGM, such as shrinkage of the lens inside the mold and unpredicted glass flow in the machine.



Source – Fraunhofer IPT

The thermal simulation of the PGM consists of a complex thermodynamic problem comprising interactions between radiative, convective and conductive thermal transport phenomena. Some of these phenomena can be mathematically modeled, but the models rely on unknown coefficients. The importance of correctly modeling these phenomena represents a great increase in the accuracy of the simulation, therefore enabling process optimization. Ostrouchov et al. (2011) addresses the identification of gap conductance between mold and glass in the PGM simulation. Kannan (2009) uses Computational Fluid Dynamics (CFD) to model the cooling phase of PGM.

1.1 OBJECTIVES

1.1.1 General Objective

Given the many advantages of improving the thermal simulation of PGM, the general objective of the present work is to analyze the viability of using both experimental and finite element simulation data to identify the unknown thermal parameters of the process.

1.1.2 Specific Objectives

- Analysis of the process and current simulation
- Proposal of an identification architecture
- Design of experiments for the identification
- FEM simulation design
- Implementation of a software for integrating the identification module with the FEM simulation
- Identification of the parameters for each proposed experiment
- Analysis of the results

1.2 THE RESEARCH INSTITUTE

This present work was developed inside Fraunhofer IPT (Figure 4), an institute of the Fraunhofer Society for the promotion of applied research. Founded in 1949, the Fraunhofer Society currently operates 74 institutes and research institutions throughout Germany (IPT, 2020).

Located in the state of North Rhine-Westphalia, more specifically in the city of Aachen, Fraunhofer IPT focuses on developing production system solutions for all fields

Figure 4 – Fraunhofer IPT Building.

Source – (WIKIPEDIA, 2010)

in industry. Within IPT, research is divided between 4 main areas: Process Technology, Production Machines, Production Quality and Metrology, and Technology Management.

Inside the area of process technology, the department of fine machining and optics focuses on ultra-precision grinding and polishing, diamond machining and the molding of high precision glass components.

The present report describes the state of the art (Chapter 2), the current environment and the conceptual solution (Chapter 3), the software project for enabling the identification of coefficients (Chapter 4), the analysis of the results (Chapter 5) and the summary and outlook works (Chapter 6).

2 STATE OF THE ART

This chapter presents the main concepts and background theories used in the development of this work from section 2.1 to 2.6.

2.1 FINITE ELEMENT METHOD

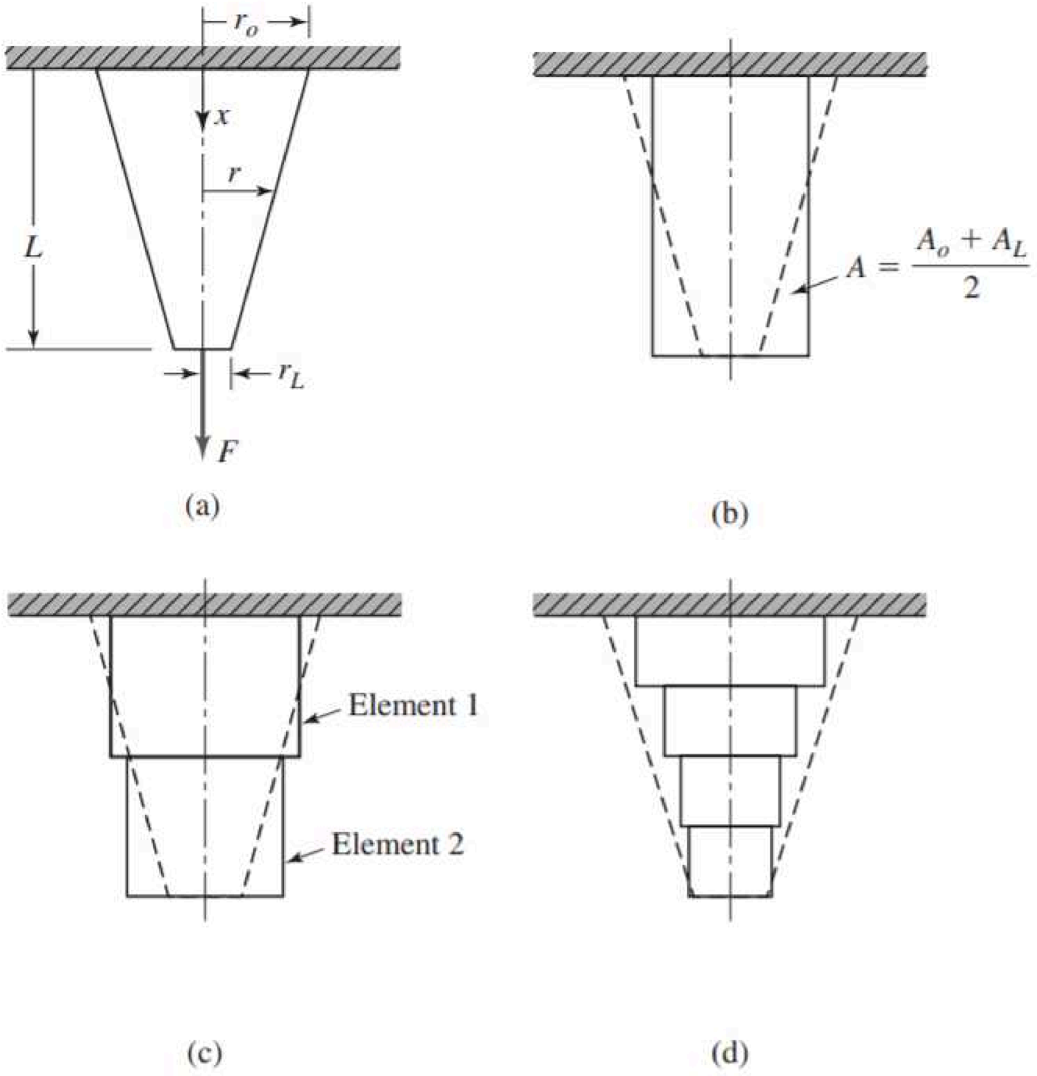
In Mathematics, many problems can be solved by approximating a continuous domain as a discrete domain with a finite number of parts. This strategy has been extensively used throughout history. For example, the "Archimedes' Problem" dated from 250 B.C. consisted of approximating the ratio of the circumference of a circle to its diameter by approximating a circle as a polygon with a large, but finite number of sides (FELIPPA, 2004). The same principle is used to calculate the area of curves.

The FEM, sometimes referred to as Finite Element Analysis (FEA), is a computational technique used to obtain approximate solutions of boundary value problems in engineering. In simple terms, a boundary value problem is a mathematical problem in which variables must satisfy a differential equation everywhere in a defined domain, as well as satisfying conditions on the boundary of this domain (HUTTON, 2003).

In other words, FEM is used to numerically solve partial differential equations in complex domains, where the approach is to divide a complex domain into a set of finite simpler sub-domains. Each sub-domain, denominated element, is solved by differential equations which yield an approximate solution for that element. Elements are connected by nodes, which guarantee the continuity of the field variable across inter-element boundaries. Nodes avoid physically unacceptable solutions containing gaps or voids in the domain. In heat transfer, a gap would represent different temperatures at the same location (HUTTON, 2003).

To exemplify the method, Figure 5.a represents a tapered cylinder fixed in one side and subject to tensile loading on the other side. In order to solve the displacement for this problem, FEM is used. Figure 5.b shows the problem discretized by one uniform cylinder. Figure 5.c uses two elements with the same length. Improving the mesh, Figure 5.d uses four elements to describe the tapered cylinder. The result of using FEM to solve the problem with a different number of elements is depicted in Figure 6.a by comparing the exact and simulated displacement at the end of the cylinder for each discretization. An increase in the number of elements, increased the accuracy of the model. The four-element model is also able to well describe the displacement throughout the whole length of the cylinder, as seen in Figure 6.b.

Figure 5 – (a) Tapered cylinder under tensile loading: $r(x) = r_0 - (x/L)(r_0 - r_L)$. (b) Tapered cylinder as a single element using an average area, while actual cylinder geometry is shown as dashed lines. (c) Tapered cylinder modelled as two, equal-length, finite elements. (d) Tapered circular cylinder modeled as four, equal-length, finite elements.

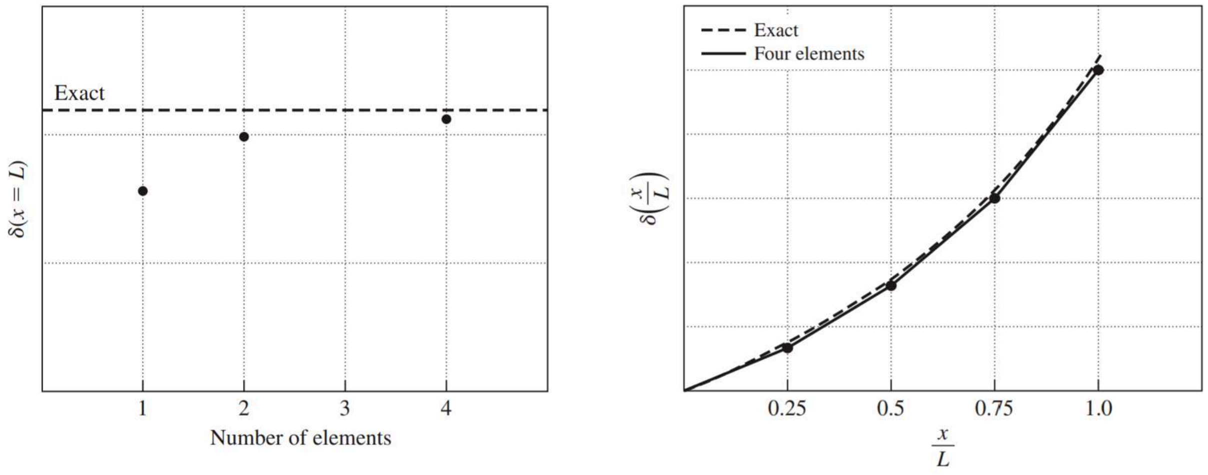


Source – (HUTTON, 2003)

2.2 HEAT TRANSFER CONCEPTS

This section presents some of the heat transfer concepts used throughout this document. These concepts are presented from section 2.2.1 to 2.2.3 and comprise the topics of conduction, convection, radiation and gap conductance.

Figure 6 – FEM results for tapered cylinder in tension of Figure 5. Displacement at x/L for each discretization compared with the exact displacement (left). Comparison of the exact solution and the four-element solution throughout the length of the cylinder (right).



Source – (HUTTON, 2003)

2.2.1 Conduction Heat Transfer

Conduction describes the energy transfer due to direct molecular contact at different temperatures (LEVENSPIEL, 2014). This phenomena was described by Fourier‘s Law (see equation 1), which states that the conduction heat flux in a certain direction is proportional to the negative temperature gradient with respect to that direction.

$$\dot{q_x} = -kA\frac{dT}{dx}. \tag{1}$$

In equation (1), $\dot{q_x}$ [W] represents the heat in direction x, k [$\frac{\mathrm{W}}{\mathrm{mK}}$] represents the thermal conductivity of the material, A [m^2] represents the area normal to the x direction and $\frac{dT}{dx}$ [$\frac{\mathrm{K}}{\mathrm{m}}$] is the temperature gradient in the x direction (LEVENSPIEL, 2014).

The complete equation for steady-state heat conduction in any arbitrary direction through an isotropic material, without heat generation, is:

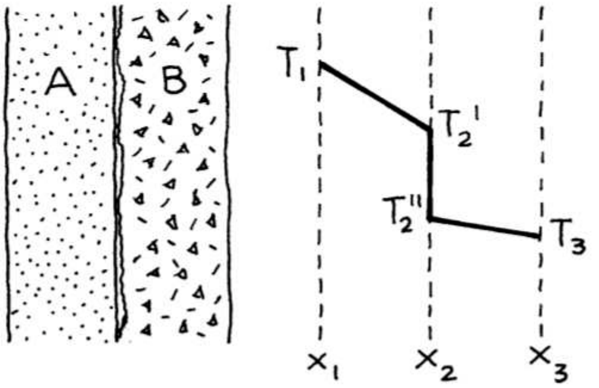
$$\dot{q} = -kA\Delta T. \tag{2}$$

Related to conduction heat transfer, a phenomena called gap conductance or gap resistance is found when heat flows across two touching plane walls. Due to the irregularities in the contact of both surfaces, an additional resistance is found at the interface, causing a temperature drop between the surfaces (LEVENSPIEL, 2014). The heat flow across the interface of surfaces A and B (Figure 7) can be described as

$$\dot{q} = -h_c A(T_2'' - T_2'), \tag{3}$$

where h_C is the contact heat transfer coefficient. The value of h_C is highly dependent on the conditions of the interface. In cases where vacuum is present, the heat transfer in the interface occurs via conduction through the real contact area between the surfaces, which consists of a set of micro-contacts. In cases where gas is present at the interface, the heat transfer occurs through the contact area, as well as conduction through the gas filling the interface. In this case, the value of h_C is also dependent on factors such as contact pressure, micro-hardness, surface roughness, gas pressure and temperature, and the the thermal conductivity of the gas (SONG; YOVANOVICH; NHO, 1992).

Figure 7 – Schematic of gap resistance between surfaces A and B.



Source – (LEVENSPIEL, 2014)

2.2.2 Convection Heat Transfer

Convection describes the energy exchanged by the movement of fluids in a macroscopic scale. Convection can manifest itself as either forced convection or natural convection. The latter is caused by an external factor, such as an imposed flow of fluid. When the motion of the fluid is not caused by an external factor, only by difference in density due to temperature variations, the phenomena is called natural convection (LEVENSPIEL, 2014). Figure 8 depicts both of these phenomena.

The convection heat transfer can be modelled by

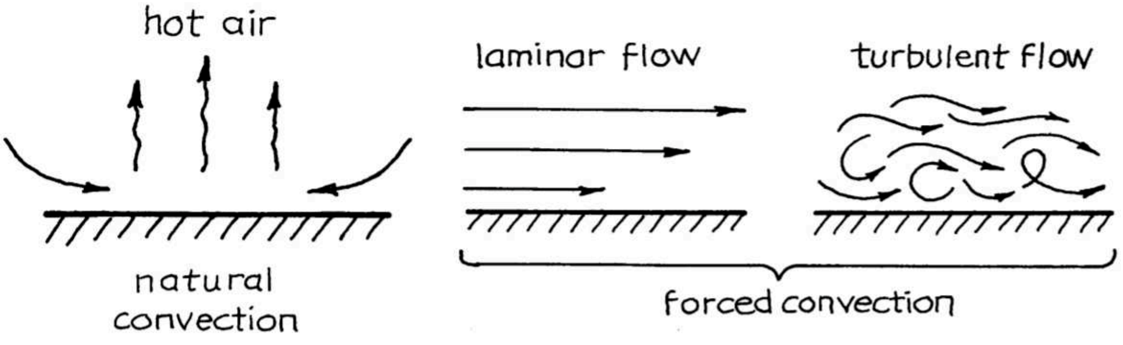
$$\dot{q} = h_C A \Delta T, \tag{4}$$

where h_C $[\frac{\mathrm{W}}{\mathrm{m^2K}}]$ represents the convection coefficient of heat transfer.

2.2.3 Radiation Heat Transfer

Different from conduction and convection, radiation heat transfer does not depend on a medium to occur. Radiation represents the thermal energy emitted from any body in the form of electromagnetic waves and is caused by the oscillations of electrons.

Figure 8 – Comparison between natural and forced convection modes.



Source – (LEVENSPIEL, 2014)

These oscillations are caused by the internal energy, hence the temperature, of matter (BERGMAN et al., 2011). The heat emitted by radiation from a black body (also known as ideal emitter) can be modelled by

$$\dot{q} = \sigma A T^4, \tag{5}$$

where σ represents the Stefan-Boltzmann constant and T the body temperature, in Kelvin.

When analyzing the thermal radiation between a surface and a black-body, the following equation can be used:

$$\dot{q} = \sigma A \epsilon (T_1^4 - T_2^4). \tag{6}$$

where ϵ represents the emissivity of the surface. This value ranges from 0 to 1 depending on how close the surface is to a black body (GANJI; SABZEHMEIDANI; SEDIGHI-AMIRI, 2018). For example, an object with emissivity 0 reflects all radiation, while an object with emissivity 1 absorbs all radiation it receives.

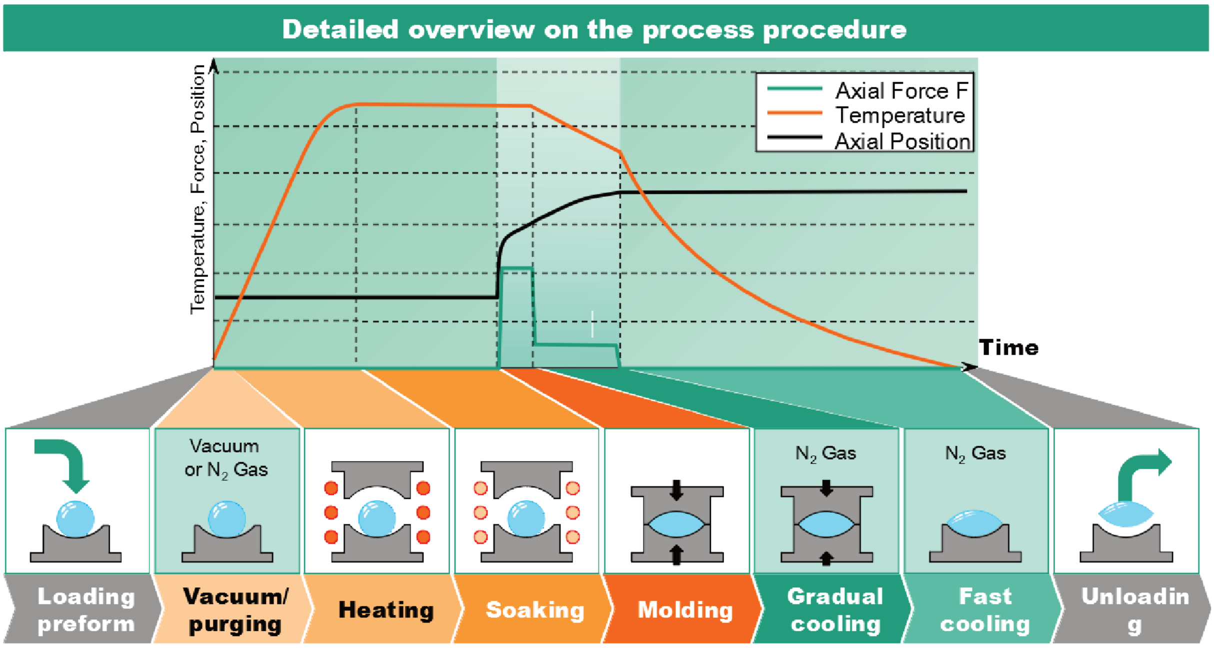
2.3 PRECISION GLASS MOLDING

2.3.1 Precision Glass Molding Process

As introduced in chapter 1, precision glass molding emerged as an alternative to roughing and grinding processes in the mass production of high-complex optical components. PGM consists of a sequence of steps that have as result a ready-to-use glass optic.

The process begins with the loading of the preform in the machine. Right after, in order to prevent oxidization of the mold at high temperatures, oxygen is removed by nitrogen purging or vacuum. Then the machine is heated up to a specific molding temperature (above the glass transition temperature). To guarantee isothermal temperature condition between glass preform and molding tools, the molding process is delayed for a couple of time in the step called "soaking". After molding is complete, the temperature in the machine decreases at a constant rate due to the purge of nitrogen together with the controlled lamp system. This step is important for it prevents any internal stresses in the lens that might cause distortion or cracking (IQBAL, 2009). After the residual stresses in the glass were already controlled, the machine is cooled down by a greater flow of nitrogen. At this point, the glass returns to a solid state and the glass component is taken out of the mold and further machining operations are no longer necessary. Figure 9 depicts the detailed overview of the precision glass molding process and its corresponding steps.

Figure 9 – Detailed overview of the precision glass molding process.



Source – Fraunhofer IPT

2.3.2 Precision Glass Molding Simulation

Even though precision glass molding has been applied in industry for many years, the absence of observation methods during the molding process strongly limits the knowledge of the viscoelastic glass material flow, influence of process parameters and shrinkage of glass lenses during molding. For this reason, many process design decisions can only be done by trial-and-error (LIU, 2018). This approach is extremely time and resource-consuming. For this reason, FEM simulation is used to better understand the process and how the glass behaves under extreme molding conditions.

The simulation of PGM is divided in mechanical and thermal models. The first model results in a stress-strain analysis in the glass. The second results in the temperature distribution inside the machine and glass. Many studies have been conducted in order to study the mechanical model of the precision glass molding, such as Liu (2018), Li (2016) and Farghaly (2019), who studied the fracture behavior of glass within precision glass molding.

The current thermal precision glass molding simulation is done by using the finite element method to solve the partial derivative heat transfer equations together with the boundary conditions that describe the heat transfer phenomena present in the process. The following phenomena are considered:

- Conduction between the parts in the mold assembly;
- Convection between the nitrogen flow and the upper and cooling plates;
- Radiation between infrared lamps and mold assembly;
- Contact conductance between cooling plates and mold assembly and glass preform and mold;

The most important thermal steps in the process are heating and cooling. For this reason, transient simulation for both these steps are done. A geometrical representation of the simulation model is presented in Figure 10.

2.4 PARAMETER IDENTIFICATION PROBLEM

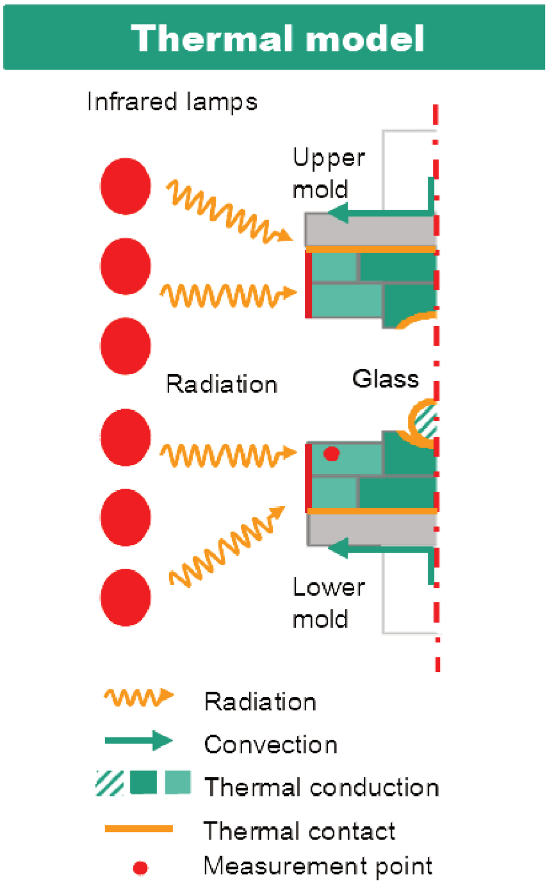
In order to better understand the parameter identification problem, it is necessary to first introduce the concepts of direct and inverse problems. Generally, modeling consists of finding a relation between a set of observation data *d* and physical parameters characterizing a model, *m*. Assuming the fundamental physics are adequately understood, and that a function *G* may be specified relating *m* and *d*, such as:

$$G(m) = d. \tag{7}$$

A direct problem, or forward problem, can be defined by finding *d* given *m*. This typically consists of solving a differential equation or an integral equation. Alternatively, an inverse problem consists of finding *m* given *d*. The task of finding *G*, given examples of *m* and *d* is defined in the literature as model identification problem (ASTER; BORCHERS; THURBER, 2018).

Parameter identification or parameter estimation can be classified as an inverse problem, where the objective is to find a discrete set of parameters that characterize a model. These parameters can represent physical properties of the modeled system

Figure 10 – Thermal model of precision glass molding process.



Source – Fraunhofer IPT

(such as density, heat transfer coefficients) or can represent abstract concepts (ASTER; BORCHERS; THURBER, 2018).

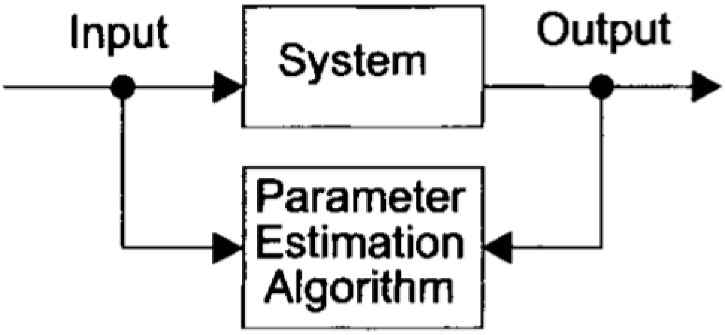
For parameters that represent a physical property of a system, in some cases parameter identification is not necessary due to theoretical or experimental determination of the parameters. However, in many cases not all system parameters are known a priori and cannot be determined by conventional techniques. Unknown parameter values can be determined through experiments with the real-world system. This can, in principle, be done through evaluation of the data measured at the system input and output by the use of the parameter-identification method. Moeller (2004) classifies the method into two architectures: direct (Figure 11) and indirect (Figure 12).

From the indirect approach, it is possible to define the parameter identification problem as an optimization problem with the objective of reducing the residual error between system output and model output:

$$\min_{p} E(Y_{system}, Y_{model}(p)), \tag{8}$$

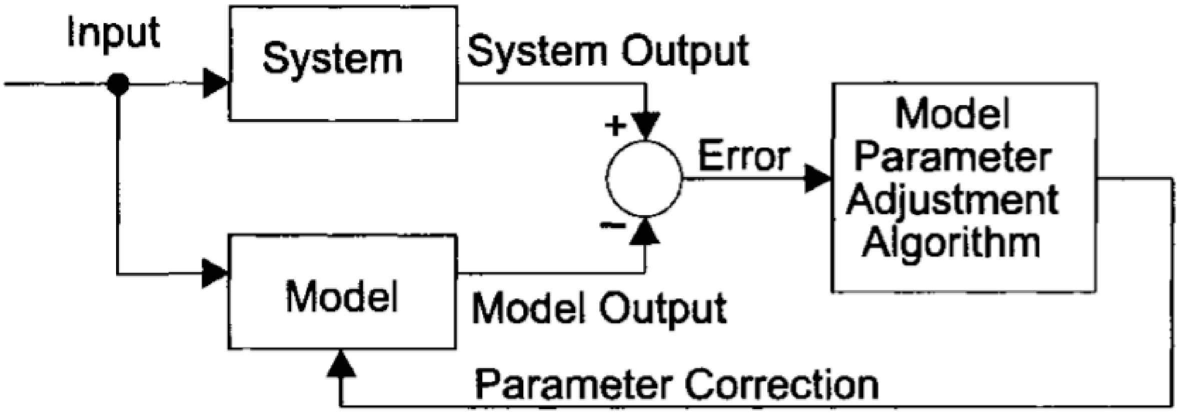
where p represents a vector of parameters, Y_{system} represents the output of the system, Y_{model} represents the model output, that is a function of p, and E represents a function

Figure 11 – Direct parameter identification method.



Source – (MOELLER, 2004)

Figure 12 – Indirect parameter identification method.



Source – (MOELLER, 2004)

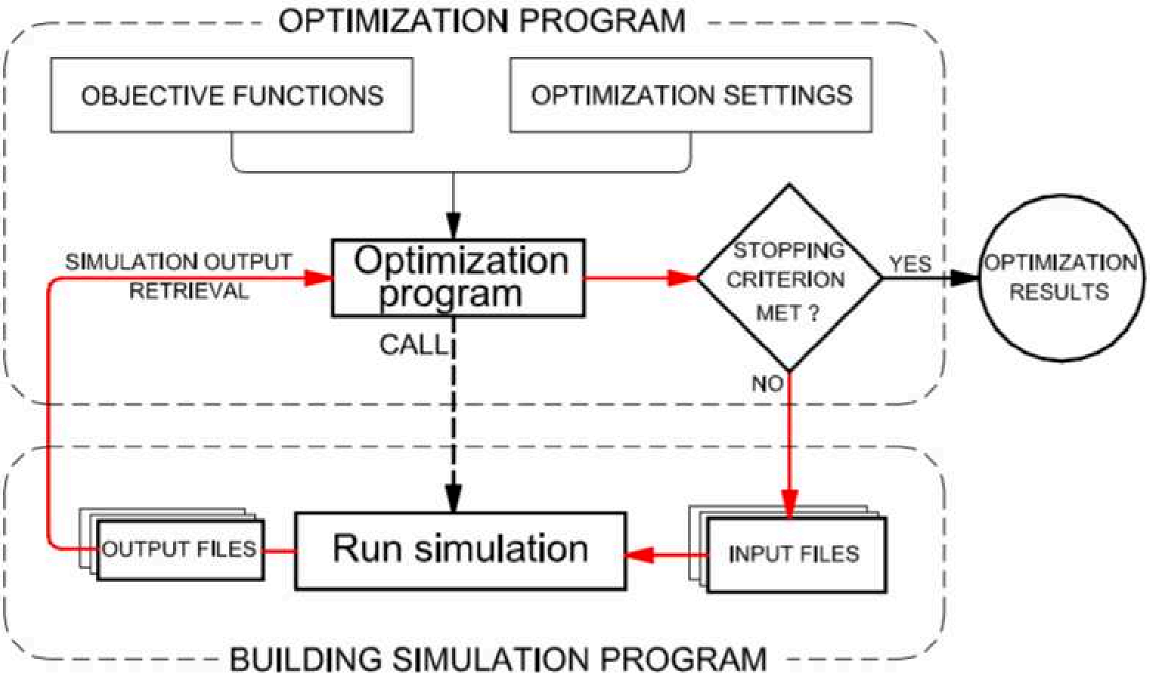
that computes the error between two values.

2.5 SIMULATION-BASED OPTIMIZATION

Computer simulations are broadly used to model real systems and evaluate them. Although choosing optimal simulation parameters can lead to improvements, the process of configuring them presents itself as a challenging problem. Historically, choosing simulation parameters consisted of selecting the best from a set of candidate parameters. Simulation-based optimization seeks to integrate optimization techniques into simulation analysis (DENG, 2007).

In a conventional optimization scenario, this process is usually automated by coupling a simulation program and an optimization module (NGUYEN; REITER; RIGO, 2014). The most typical architecture can be seen in Figure 13, having an optimization module and a module for the dynamic generation of simulations.

Figure 13 – Simulation-based optimization loop.



Source – (NGUYEN; REITER; RIGO, 2014)

Nguyen, Reiter, and Rigo (2014) presents the major phases for simulation-based optimization. These phases are preprocessing, which consists in the formulation of the optimization problem; the optimization phase, which consists of running the optimization algorithms, monitoring convergence and possible errors; and the post-processing phase, consisting of the analysis of results.

Analogously to the concept of indirect parameter identification presented in Figure 12, simulation-based optimization also presents itself as an optimization problem. However, simulation-based optimization is used as a general approach to optimize a simulation using any objective function. Assuming that the objective function is the residual between experimental and simulation data, that is the error presented in Figure 12, it is possible to use simulation-based optimization to identify unknown parameters. This approach is used in Kleinermann and Ponthot (2003) to identify the material parameters in metal forming simulation using experimental and FEM simulation data. Lecompte et al. (2007) also uses the same strategy to identify the four in-plane orthotropic engineering constants of composite plate materials.

2.6 OPTIMIZATION ALGORITHMS

Since simulation-based optimization presents itself as an optimization problem, optimization algorithms need to be used. In order to choose a set of algorithms, the objective function needs to be taken into account.

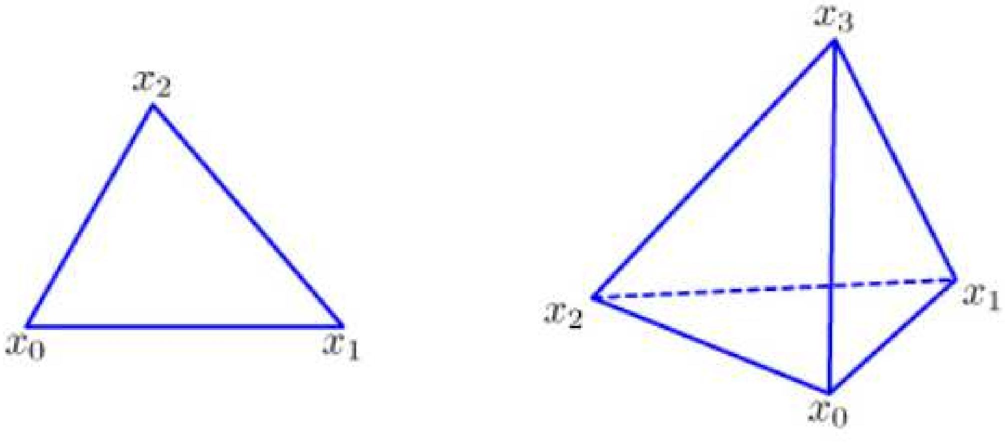
Within simulation-based optimization, the objective function is an associated measurement of an experimental simulation. Due to the complexity of the simulation, the objective function may be difficult and expensive to evaluate. Also derivative information is typically unavailable in FEM simulation software (DENG, 2007). For this reason, derivative-free optimization algorithms are typically applied.

2.6.1 Nelder-Mead

Nelder-Mead or Downhill Simplex is a derivative-free direct search optimization algorithm invented in 1965 by Nelder and Mead. The method is based in a structure called simplex. A simplex in $\mathbb{R}^n$ is defined as the convex frame of n+1 vertices $x_0, \ldots, x_n \in$ *Rn*. For instance, a simplex in $\mathbb{R}^2$ is a triangle, and a simplex in $\mathbb{R}^3$ is a tetrahedron (Figure 14) (SINGER; NELDER, 2009). The general algorithm is given by

- construct an initial simplex S;
- repeat the following steps until the termination test is satisfied:
  - calculate the termination test information;
  - if the termination test is not satisfied then transform the working simplex;
- return the best vertex of the current simplex S and the associated function value.

Figure 14 – Simplex in $\mathbb{R}^2$ (left) and in $\mathbb{R}^3$ (right).



Source – (SINGER; NELDER, 2009)

The transformation of the simplex consists of substituting the worst vertex by using reflection, expansion or contraction with respect to the best vertex (SINGER; NELDER, 2009).

2.6.2 Particle Swarm Optimization

Particle Swarm Optimization (PSO) algorithm is an evolutionary meta-heuristic algorithm proposed in Kennedy and Eberhart (1995). In PSO a number of particles, or candidate solutions, traverse through the search space, and each evaluates the objective function at its current location. Each particle then determines its movement through the search space by combining global and individual bests. Since both global and individual bests are taken into account, eventually the swarm is likely to move close to an optimum of the fitness function (POLI; KENNEDY; BLACKWELL, 2007).

Each particle is described by its position in the search space and its velocity. Changes in a particle velocity are governed by:

$$v_{id} = v_{id} + c_1 rand()(p_{id} - x_{id}) + c_2 rand()(p_{gd} - x_{id}), \tag{9}$$

$$x_{id} = x_{id} + v_{id}, \tag{10}$$

where v_{id} represents the velocity of a certain particle, p_{id} the best previous position, x_{id} the current position, p_{gd} the best previous global position, c_1 and c_2 are constants, and *rand*() represents a function that returns a random number in the range [0,1].

In order to reach good results with the algorithm, the choice of the parameters c_1 and c_2 are really important since they decide how much the particle‘s velocity will be affected by its own best and the population global best. Also, the choice of population size is an important step for the right tuning of PSO (SHI; EBERHART, 1999).

2.6.3 Bayesian Optimization

Bayesian Optimization (BO) is a stochastic optimization algorithm applied extensively to solve problems where the objective function is expensive to evaluate and the goal is to find a global optimum. BO consists of two main components: a Bayesian statistical model for modeling the objective function (typically a Gaussian Process), and an acquisition function for deciding where to sample next (FRAZIER, 2018). The main steps of BO are:

- Initial points are sampled from the search space and evaluated by the cost function;

- A surrogate model of the cost function is generated using the evaluated points, therefore creating the posterior model;

- The acquisition function is used to decide where to sample next based on the posterior model;

- New points are sampled based on the surrogate model;

The tuning of BO is done by defining the parameters for the statistical model and the acquisition function. Further analysis of the choice of parameters is developed in (FRAZIER, 2018).

3 CURRENT ENVIRONMENT AND RESEARCH APPROACH

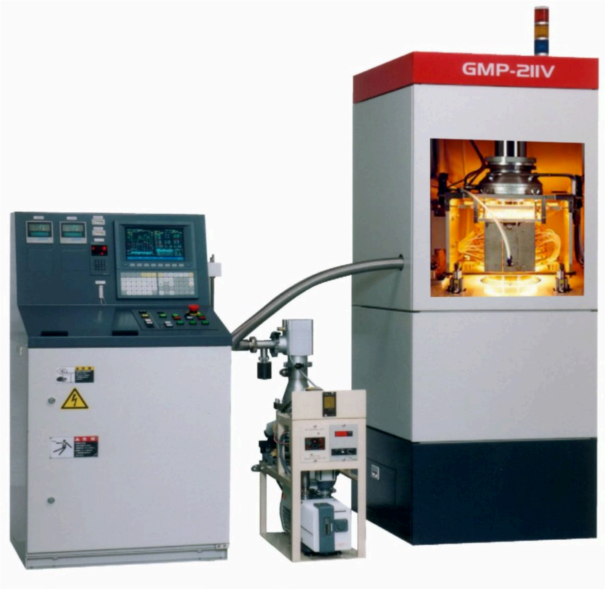
This chapter presents the current PGM environment inside of IPT, as well as the process data collection architecture, followed by the opportunities for the thermal simulation model and the research approach for parameter identification.

3.1 CURRENT ENVIRONMENT

3.1.1 Glass Molding Machine: Toshiba GMP-211V

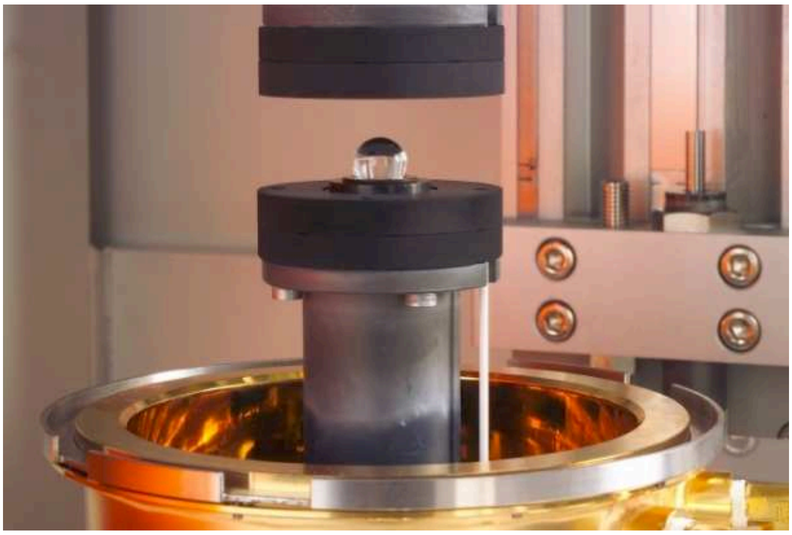
Precision glass molding is normally done using dedicated machines that implement the whole process. At the research institute, the Toshiba GMP-211V machine is used for the molding of small lenses. The machine consists of a Computer Numerical Control (CNC) that controls the main axis, therefore controlling the pressing, and a Programmable Logic Controller (PLC) that is responsible for the discrete logic of the process, as well as for the reading of sensors. The PGM molding machine can be seen in Figure 15. The structure in the left of the figure is the machine control unit, where molding programs can be inserted and the visualization of the process variables can be done. The right structure is where the molding occurs. Figure 16 depicts with more detail the inside of the chamber before the molding process. It is possible to see the glass preform standing on top of the mold assembly. The chamber during molding can be seen in Figure 17. The main specifications for the GMP-211V can be seen in Figure 18.

Figure 15 – Toshiba GMP-211V machine.



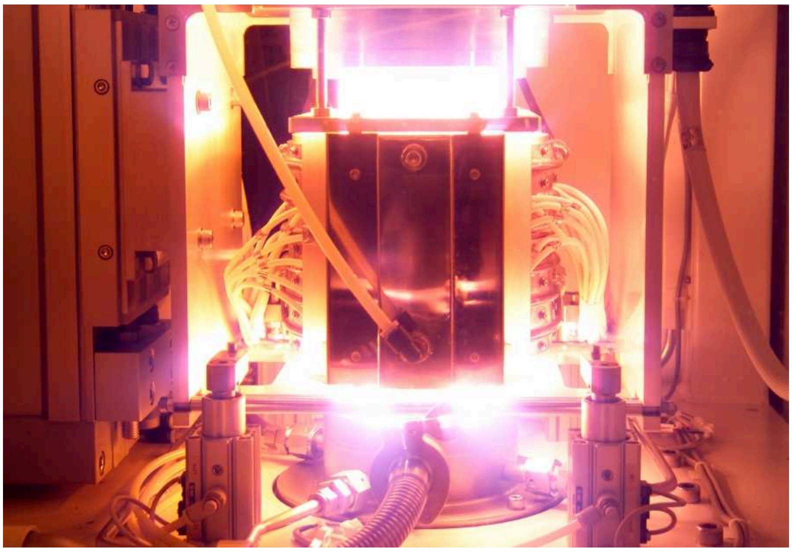
Source – (KLOCKE; STAASMEYER, 2015)

Figure 16 – Toshiba GMP-211V before molding.



Source – (KLOCKE; STAASMEYER, 2015)

Figure 17 – Machine during molding. The intense glow is due to the infrared lamps.



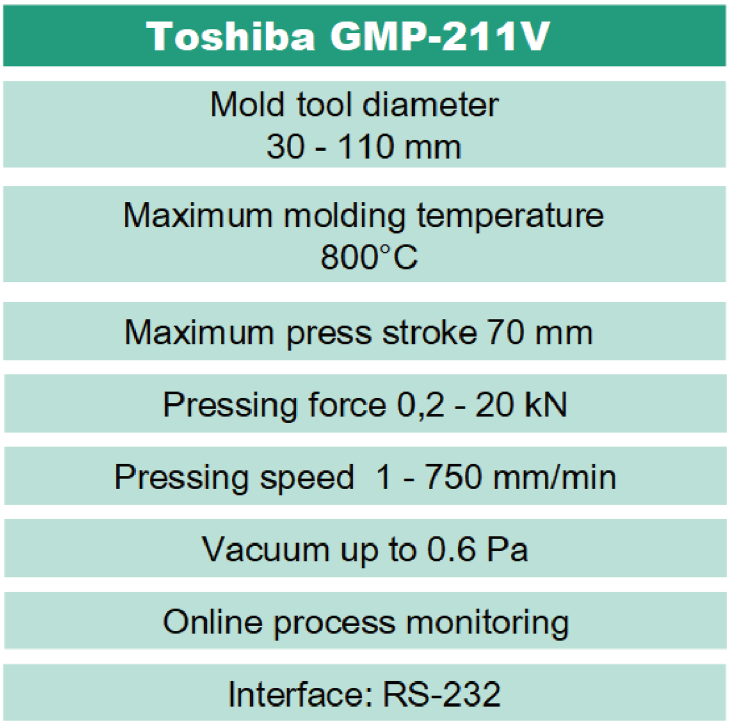
Source – (KLOCKE; STAASMEYER, 2015)

3.1.2 Machine Data Collection

In order to provide support for the PGM process, a complete Information Technology (IT) architecture is set up to read the machine information and store it into a database. The complete architecture can be divided into three software projects: Serial Reader, WebAPI and WebApp.

Serial Reader is a piece of software deployed in a notebook and connected to the serial port of the molding machine. This software is responsible for reading the sensor

Figure 18 – Toshiba GMP-211V machine specifications.



Toshiba GMP-211V

Mold tool diameter
30 - 110 mm

Maximum molding temperature
800°C

Maximum press stroke 70 mm

Pressing force 0,2 - 20 kN

Pressing speed 1 - 750 mm/min

Vacuum up to 0.6 Pa

Online process monitoring

Interface: RS-232

Source – Original

data from the machine using serial communication via Modbus and RS232. The read data are then sent to the WebAPI via HyperText Transfer Protocol (HTTP) requests.

The WebApi is an Application Programming Interface (API) deployed in an internal server at the research institute. The read data are sent to the API, which is responsible for storing the data in a database. The structure of the database can be seen in Figure 19. Many tables exist in the database, however the table that is used in this project is "Press_Logs", which stores the internal machine sensor data.

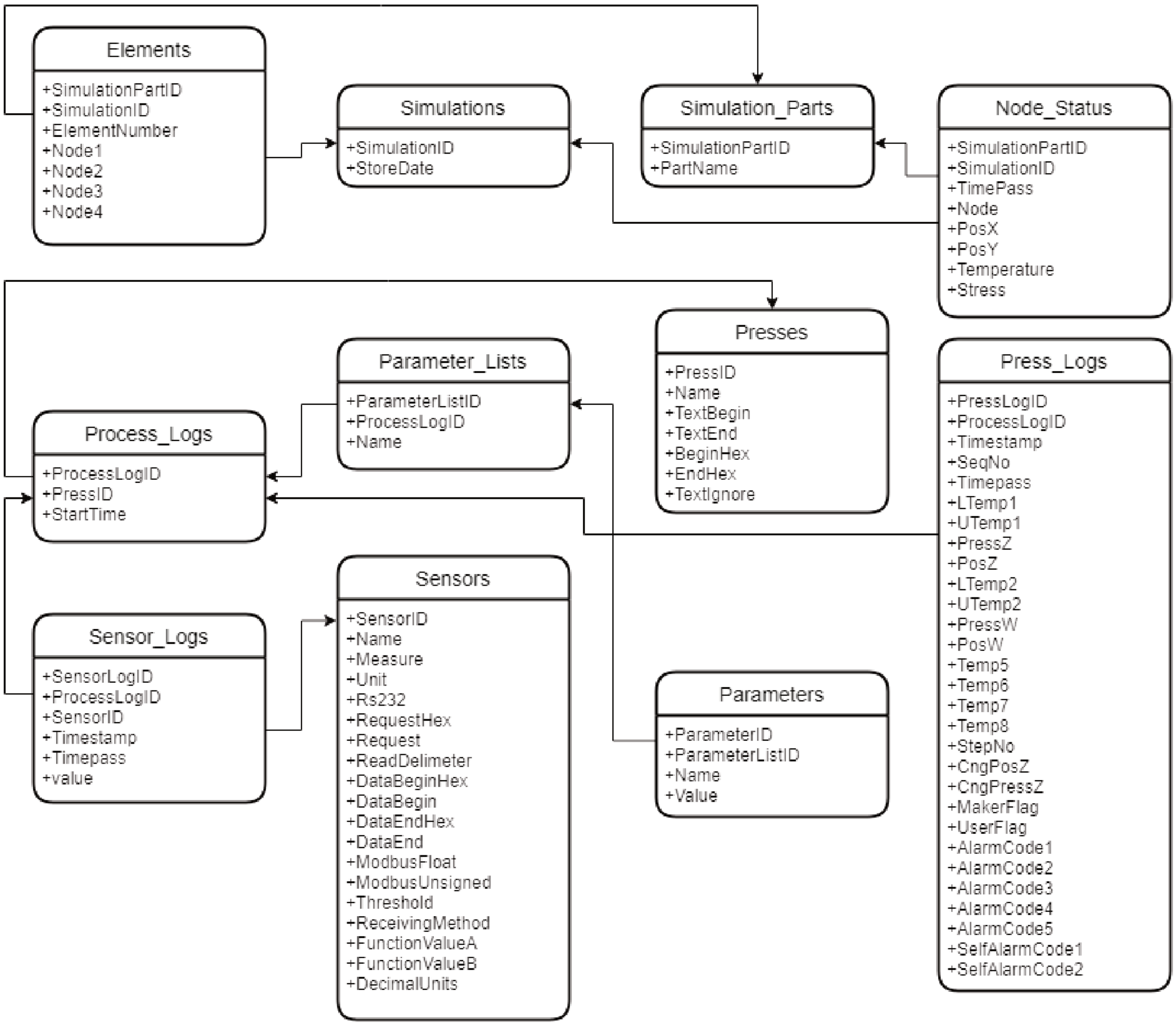
Finally, the WebApp is a web application used for visualization of live and stored process data. It is normally used by machine operators and engineers to analyze past and live processes. Figure 20 shows the visualization of a molding process in the WebApp.

3.2 OPPORTUNITIES IN PGM SIMULATION

As described in section 2.3.2, the present thermal phenomena in the machine are well known and can be described mathematically, though with undetermined coefficients. That is the case for the convection coefficient in the machine, the gap conductance and the emissivity factor of the radiation heat transfer.

However, for the lamp system, there is no mathematical description defined. In the simulation, the radiative heat flux in the boundary is calculated by the following

Figure 19 – Database schema.



Source – Fraunhofer IPT

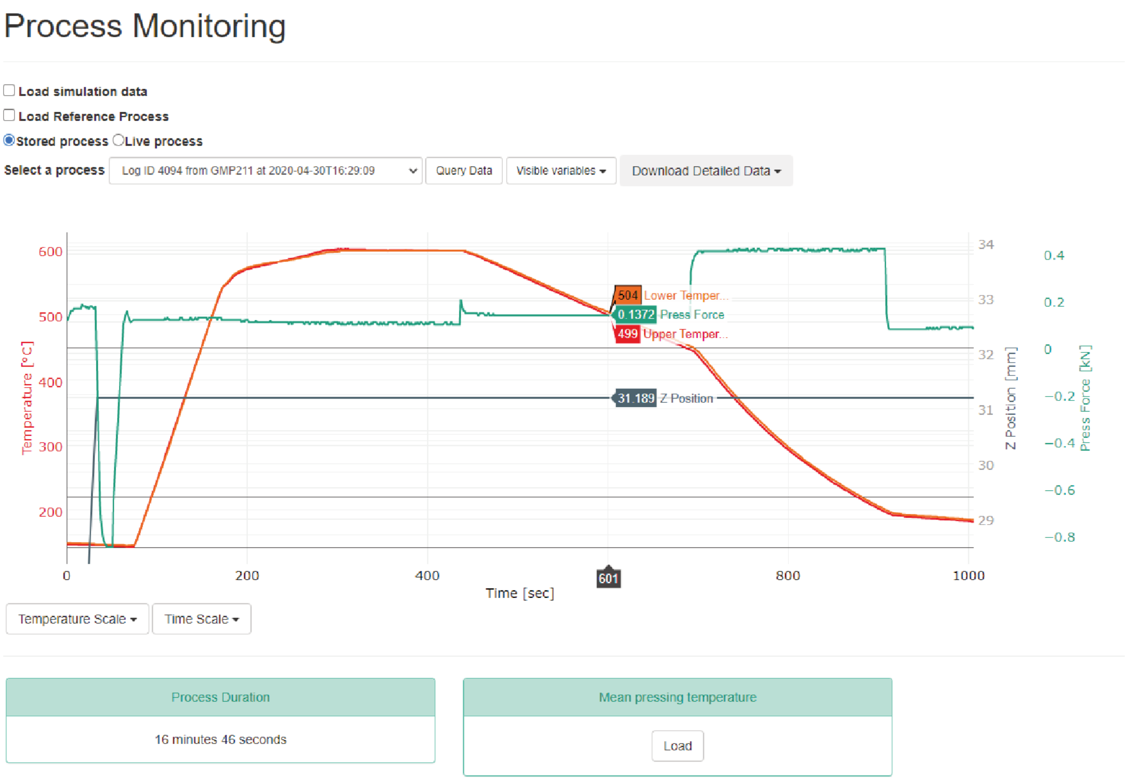
equation:

$$q = \sigma\epsilon(T_{lamp}^4 - T_{mold}^4), \tag{11}$$

being ϵ the emissivity, σ the Stefan-Boltzmann constant, T_{lamp} the estimated temperature in the lamp, and T_{mold} the temperature measured in the measuring point (as shown in Figure 10).

In order to calculate the heat flux in the boundary, the lamp temperature is necessary. However, it is not possible to measure the temperature of the lamp because the filament of the lamp is inside a quartz tube. The said temperature of lamps is actually the temperature of the filament, which is physically isolated due to the quartz tube. Since the quartz tube is transparent, the surface of the tube does not absorb the heat radiated by the filament, therefore measuring the temperature in the surface of the tube is also not possible. For this reason, the approach to estimate the heat flux

Figure 20 – Process data visualization in the WebApp.



Process Monitoring

☐ Load simulation data
☐ Load Reference Process
● Stored process ○ Live process

Select a process: Log ID 4094 from GMP211 at 2020-04-30T16:29:09

Process Duration: 16 minutes 46 seconds

Mean pressing temperature: Load

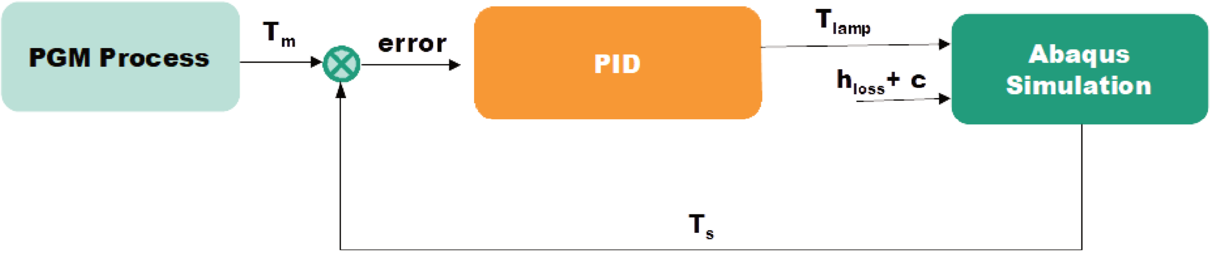
Source – Original

is by using a PID controller that uses the error between simulation temperature in the measuring point and the experimental measured temperature as input is considered. Since the physical heat lamp system consists of a PID controller actuating on the power applied to the lamps, the approach seems reasonable. However, this approach adds dynamical error to the simulation. Also, modeling errors affect the control action of the controller. That means that the controller will try to compensate any modeling errors seeking to decrease the error by changing the lamp temperature. For example, Figure 21 shows the PID controller that is used for estimating the temperature of the lamps, where T_m represents the measured temperature in the mold, T_{lamp} the estimated lamp temperature, h_{loss} the convection coefficient, and T_s the simulation temperature. Assuming that c represents modeling errors in the convection coefficient, the controller will try to compensate this error by changing the estimated lamp temperature, even though the convection coefficient modeling error is independent of the lamp system. For this reason, a set of parameters/process variables needs to be identified.

- System heat losses

The current simulation does not take into account the heat losses to the environment. Therefore, it is necessary to identify: the conduction coefficient between mold system

Figure 21 – PID architecture for estimating the lamp temperature.



Source – Original

and environment through upper and lower flanges; the radiation to ambient through lamp system (the emissivity coefficient); and the gap conductance.

- Nitrogen flow related losses

The current simulation uses only one value for convection coefficient, which is independent of the nitrogen flow used in the process. Therefore, it is necessary to identify the convection coefficient, as well as how it changes based on the nitrogen flow operating point.

- Temperature of infrared lamps

The infrared lamp system is probably the most complex one. The emissivity needs to be identified, as well as a model that relates the lamp temperature and power applied.

3.3 RESEARCH APPROACH

In order to identify the unknown thermal parameters of the precision glass molding model, simulation-based optimization is used. This section describes the identification architecture, as well as the design of experiments.

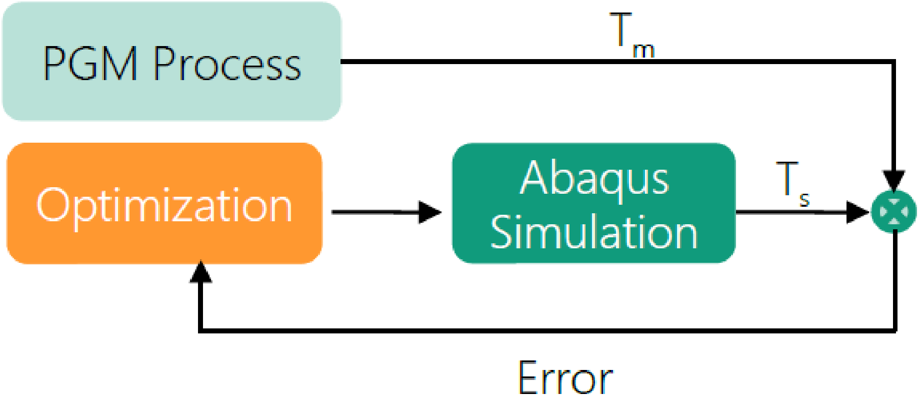
3.3.1 Architecture

The proposed architecture for identifying the unknown parameters is based in Figure 12. The core idea is to compute the error between the temperature curve solved from the FEM software and the experimental temperature curve measured by a temperature sensor installed in the machine (see Figure 22). Therefore, the idea is to use the data from a temperature sensor located in the mold to identify the thermal model. If the temperature obtained in the simulation is the same as the temperature measured by the real sensor, the model is assumed to be correct. Following this approach, the following optimization problem is proposed:

$$\min_{x} \frac{1}{n} \sum_{i=1}^{n} (T_m - T_s(x))^2, \tag{12}$$

where n represents the number of points measured in the experiment, T_m represents the temperature measured by the sensor, T_s represents the temperature given by the simulation, and x represents the unknown thermal parameters.

Figure 22 – Identification Architecture.



Source – Original

This architecture enables finding a vector x that minimizes the error between simulation and experimental data. The approach to identify the parameters is to decouple the many thermal phenomena present in the process into different experiments (see section 3.3.2), model the experiments in the FEM simulation software following some hypotheses of which thermal phenomena occur in the experiment, identify the parameters using the experimental and simulation data through optimization, and evaluate the results.

The main limitation of this approach is the difficulty of creating a model that imitates the dynamic behaviour of the system in a way that it can be considered good enough for its representation. In other words, it is necessary that the designed model is able to describe the process dynamics. A great advantage of this process is the fact that it requires no extra instrumentation in the machine, hence making it a cost-effective identification method.

The greatest challenge in this approach is the fact that FEM simulations can take minutes. Therefore, in order to have an acceptable training time, the convergence time (number of iterations for convergence) is reduced. Also, gradient information is not available from the simulation, thus forcing the use of derivative-free optimization algorithms, which take longer to converge.

3.3.2 Design of Experiments

The precision glass molding process consists of many heat transfer phenomena happening simultaneously, which makes the identification process much harder. In order to solve this, the approach is to decouple the heat transfer phenomena of the machine through well-designed experiments. Having identified the decoupled models/parameters, these can be later re-coupled in order to simulate the whole process. After introducing the approach, the experiments to be conducted are:

- Vacuum cooling

This experiment consists in heating the machine (without glass preform) up to a high temperature. Afterwards, the machine is cooled down by vacuum. The main goal of this experiment is the identification of the system heat losses model.

- Nitrogen cooling

This experiment consists in heating the machine (without glass preform) up to a high temperature. Afterwards, the machine is cooled down by nitrogen (different nitrogen flow setups will be used). The main goal of this experiment is the identification of the convection heat losses, as well as their relation with the nitrogen flow operating point.

- Heating

This experiment consists in heating the machine (without glass preform) up to a high temperature. The heating temperature curve will be used to identify the infrared lamp model, as well as the emissivity of the radiation heat transfer.

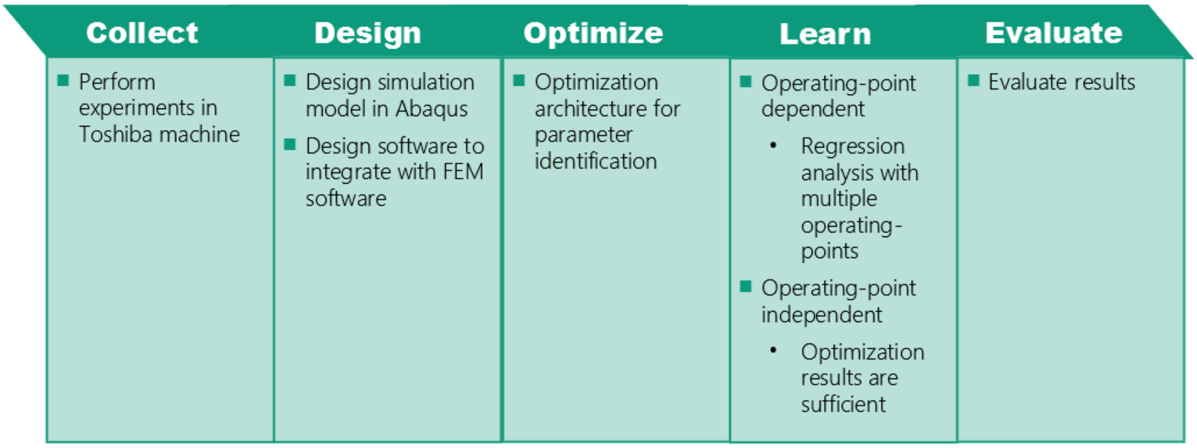
The identification architecture presented in 3.3.1 is used to identify the unknown parameters from the experiments described in the items above. However, in some of the experiments, these parameters can vary depending on the operating-point of the machine. For example, in the nitrogen cooling experiments, the nitrogen flow is responsible for the cooling rate in the machine. Therefore, the cooling parameters will vary depending on the operating-point. For this reason, an extended approach will be used in order to identify operating-point related parameters.

The extended approach consists of using the identification approach in multiple operating-points. Having multiple observations, it is possible to apply regression techniques to identify an expression that relates each unknown parameter with the operating-point. The proposed method for the identification can be seen in Figure 23.

3.3.3 Software Requirements

To allow the use of the architecture presented in Figure 22, the optimization module needs to be created. The general software methodology for the development

Figure 23 – Proposed method for identifying the unknown parameters in PGM.



Collect	Design	Optimize	Learn	Evaluate
■ Perform experiments in Toshiba machine	■ Design simulation model in Abaqus ■ Design software to integrate with FEM software	■ Optimization architecture for parameter identification	■ Operating-point dependent • Regression analysis with multiple operating-points ■ Operating-point independent • Optimization results are sufficient	■ Evaluate results

Source – Original

of this module was Feature-Driven Development (FDD), an iterative and incremental software development process based on agile methods. This methodology was chosen from the nature of the optimization module, which has clear sequential features that must be developed to correctly integrate with the FEM software.

The process consists of breaking the project into multiple features that can be individually developed and later integrated. The goal of the optimization module is to generate simulation jobs based on the error between experimental and simulation values. From this description, the following features are defined:

- Read experimental data from database
- Read simulation temperature curves from the simulation software
- Generate simulation jobs dynamically
- Calculate the error between experimental and simulation curves
- Interface between simulation software and optimization algorithms

Chapter 4 describes how these features were developed and integrated with the FEM simulation.

4 DEVELOPMENT

This section comprises the description of how the finite element simulation was modeled in the Abaqus simulation software and how the connection between FEM simulation and optimization module was developed.

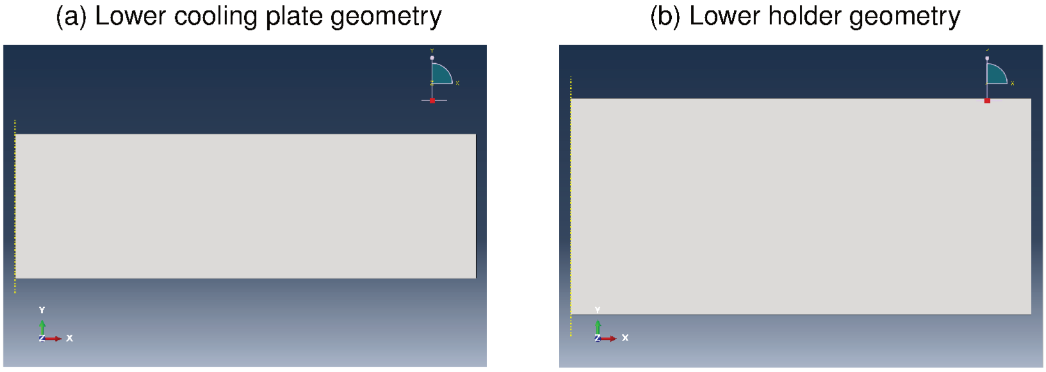
4.1 FINITE ELEMENT MODEL IMPLEMENTATION IN ABAQUS

The cooling model of the experiments described in section 3.3.2 is a subset of the complete simulation of PGM. For this reason, the current model is used and only a few changes were made. The main change is the removal of glass preform from the simulation. A summary of the modeling steps is presented in the following sections.

4.1.1 Parts

This step consists in creating the geometry of the parts used in the simulation. Since no glass preform is used in the experiments, this part was removed. The remaining parts are only the upper and lower cooling plates and the upper and lower holders.

Figure 24 – Geometry of lower parts.



(a) Lower cooling plate geometry

(b) Lower holder geometry

Source – Original

4.1.2 Materials

In the materials section, the thermal and mechanical parameters for each material are defined. The materials considered for the cooling plate and mold die (holder) were FHR96 and TJF03, respectively. The material parameters were taken from the material‘s datasheet (FUJILLOY, 2008) and from Kannan (2009), and can be seen in Table 1 and Table 2.

Table 1 – Material parameters for FHR96

Property	Value	Unit
Conductivity	54	$Wm^{-1}K^{-1}$
Density	17200	kgm^{-3}
Young‘s Modulus	350	GPa
Poisson‘s Ratio	0.28	-
Expansion Coefficient	5.4E-06	MK^{-1}
Specific Heat	400	$Jkg^{-1}K^{-1}$

Source – Original

Table 2 – Material parameters for TJF03

Property	Value	Unit
Conductivity	43	$Wm^{-1}K^{-1}$
Density	15400	kgm^{-3}
Young‘s Modulus	670	GPa
Poisson‘s Ratio	0.17	-
Expansion Coefficient	4.5E-06	MK^{-1}
Specific Heat	214	$Jkg^{-1}K^{-1}$

Source – Original

4.1.3 Assembly

The assembly section defines which parts are contained in the model and how they are geometrically disposed. Also, contact surfaces can be defined, which are used to define where the heat exchange occurs. Since the machine is symmetrical, it is possible to use an axis-symmetric boundary. This reduces the amount of elements in the simulation, therefore reducing total simulation time. The complete assembly can be seen in Figure 25.

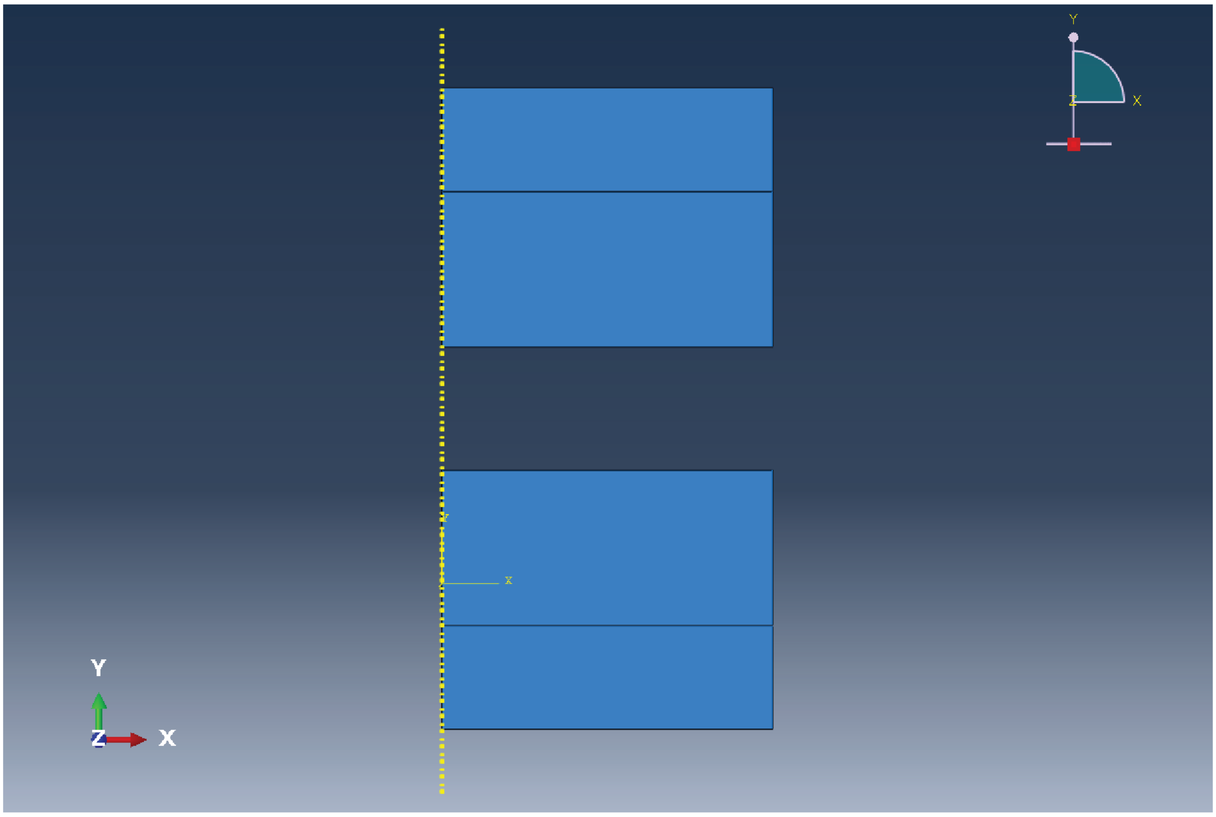
4.1.4 Interactions

Interactions in Abaqus represent any type of physical effects happening between two or more surfaces. In the model, conduction (upper and lower), convection (upper and lower), radiation and contact conductance (upper and lower) interactions are defined. The location of these interactions can be seen in Figure 10.

4.1.5 Steps

Steps represent a convenient phase of the process, in which an analysis is done. The division of a simulation into steps enables sequential analyses of different types (e.g. a transient heat analysis followed by a static stress analysis). In the current simulation, the steps are: initial, heating and cooling. Since the first experiments defined in section 3.3.2 consist of cooling only, the defined steps are initial and cooling. The

Figure 25 – Assembly.



Source – Original

initial step is responsible for initializing the temperature distribution in the assembly and any other initial conditions or interactions. During the cooling step, the interactions are created.

After modeling, the analysis job can be performed. Under the hood, all of the configuration and modeling done in the last steps is summarized in an input file. This file contains all information about the model and can be submitted to the Abaqus solver, which runs the job and generates an output file with the simulation output.

4.2 DEVELOPMENT OF THE OPTIMIZATION MODULE

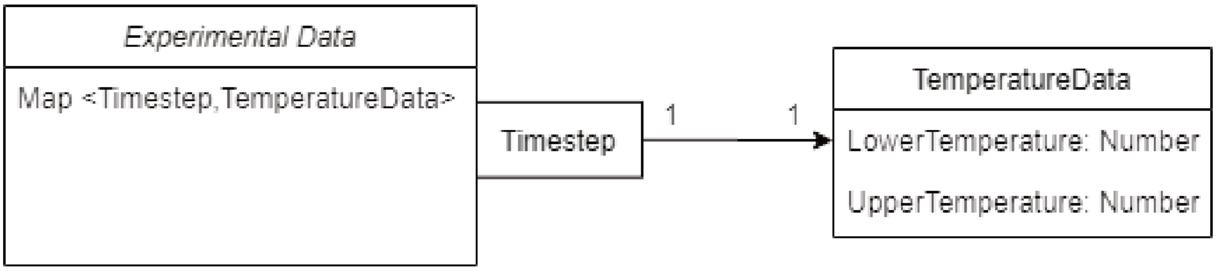
Simulation-based optimization applied to parameter identification depends on the existence of an interface between the simulation software (solver) and the optimization module. The core concept is that the optimization module provides a vector x containing the simulation parameters. After the simulation, the error between the experimental curve and the simulation curve is later used to adjust the vector x. In section 3.3.3, a set of features was defined. Following the FDD methodology, each feature was implemented individually.

The language chosen to implement the optimization module was Python, a general-purpose high-level programming language. Python offers native support for file manipulation, extensive support libraries such as database drivers and optimization packages. In addition, its simple syntax and scripting characteristic enables the fast development of features, as the developed code is not compiled.

4.2.1 Read experimental data from database

As presented in section 3.1, the experimental data is stored in a SQL database. Therefore, the process of reading the data is quite straightforward and consists of performing a query informing the process ID and the upper and lower temperature fields. However, the query result returns a list of tuples containing the time-step and the temperature values. This format is inefficient and hard to manipulate. For this reason, the database data are processed and stored in a new data structure. Figure 26 depicts the UML class diagram representing the data structure that stores the experimental data. This data structure is interesting for the application due to the fact that values from a specified time-step can be retrieved from the structure in constant time complexity.

Figure 26 – UML class diagram representing the data structure to store the experimental data.



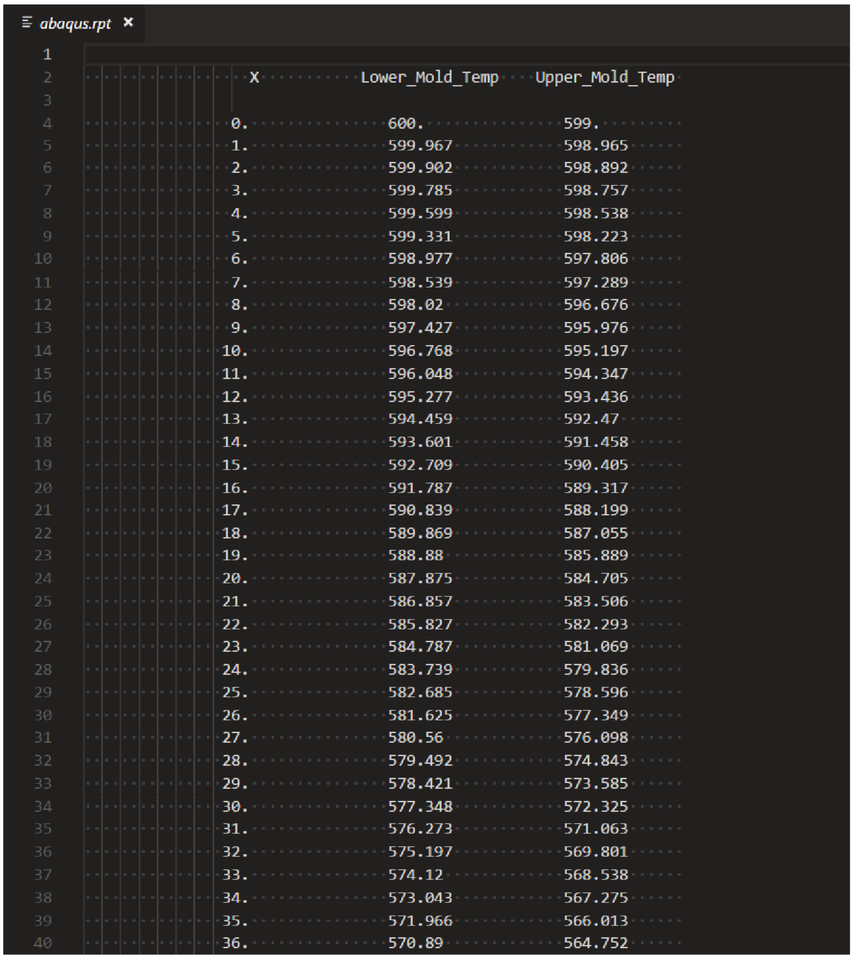
Source – Original

4.2.2 Read simulation output

After running an analysis in Abaqus, it is possible to export the simulation output to a report file (abaqus.rpt). This file contains the temperature measured in the sensor position at every time-step of the simulation. The format of the report file can be seen in Figure 27.

The approach used to transform the content from the report file into a format that can be easily understood and manipulated by computer scripts is to read the file line by line, parse the values and insert them into a data structure. Parsing is done using regular expressions, a sequence of characters that define a search pattern. This regular expression selects every number that it can find (including integers and decimals). These values are stored in the same data structure presented in Figure 26.

Figure 27 – Abaqus report file format.



```

              X          Lower_Mold_Temp   Upper_Mold_Temp

             0.               600.              599.
             1.               599.967           598.965
             2.               599.902           598.892
             3.               599.785           598.757
             4.               599.599           598.538
             5.               599.331           598.223
             6.               598.977           597.806
             7.               598.539           597.289
             8.               598.02            596.676
             9.               597.427           595.976
            10.               596.768           595.197
            11.               596.048           594.347
            12.               595.277           593.436
            13.               594.459           592.47
            14.               593.601           591.458
            15.               592.709           590.405
            16.               591.787           589.317
            17.               590.839           588.199
            18.               589.869           587.055
            19.               588.88            585.889
            20.               587.875           584.705
            21.               586.857           583.506
            22.               585.827           582.293
            23.               584.787           581.069
            24.               583.739           579.836
            25.               582.685           578.596
            26.               581.625           577.349
            27.               580.56            576.098
            28.               579.492           574.843
            29.               578.421           573.585
            30.               577.348           572.325
            31.               576.273           571.063
            32.               575.197           569.801
            33.               574.12            568.538
            34.               573.043           567.275
            35.               571.966           566.013
            36.               570.89            564.752
```

Source – Original

4.2.3 Dynamic generation of simulation jobs

Abaqus uses input files to describe the simulation job. This file is generated using the Abaqus user interface, however its format is text-based and can, therefore, be overwritten. For example, the following code defines a convection interaction in the Abaqus input file. The first line describes the interaction name. The following line defines which type of interaction occurs, in this case a convection interaction. The last line defines the name of the surface where the interaction occurs, followed by the film type label (F is used for a uniform film coefficient and FNU for a non-uniform film coefficient), the sink temperature, and the convection coefficient. Altogether, the code defines a uniform convection interaction named “Int-Cond-lower_Mold“ in the surface “Surf-Conv-lower_CoolPlate“ with sink temperature of 20℃ and convection coefficient of $239\frac{\mathrm{W}}{\mathrm{m}^2\mathrm{K}}$.

Algorithm 1: Definition of a convection interaction in Abaqus input file

```
** Interaction: Int-Cond-lower_Mold
*Sfilm
Surf-Conv-lower_CoolPlate, F, 20., 239
```

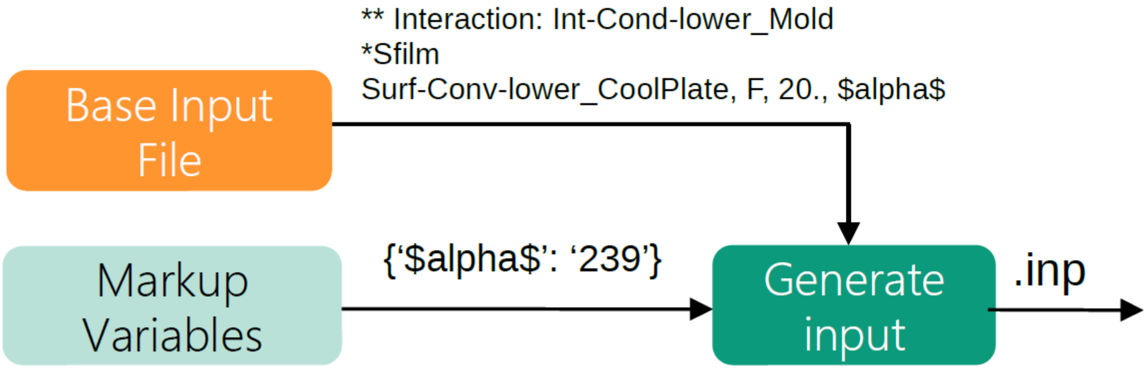
Therefore, the approach to generate the simulation files dynamically is to create a base input file with the description of the model and simulation parameters, which are presented as markup variables. Taking the code above as an example, if the convection coefficient ought to be changed dynamically, the base input file would look like this:

Algorithm 2: Definition of a convection interaction in Abaqus input file with markup variable for convection coefficient

```
** Interaction: Int-Cond-lower_Mold
*Sfilm
Surf-Conv-lower_CoolPlate, F, 20., $alpha$
```

In order to run a simulation with a certain convection coefficient, the markup variable can be replaced by the value and a valid input file is generated. This approach can be extended for multiple markup variables. The function that generates the input receives a base file and a dictionary of markup variables with the markup as key and the value to be replaced as its value. The markup variables are replaced by the value contained in the dictionary. The input file generation workflow is depicted in Figure 28.

Figure 28 – Workflow for the input file generation.



```
** Interaction: Int-Cond-lower_Mold
*Sfilm
Surf-Conv-lower_CoolPlate, F, 20., $alpha$
```

Source – Original

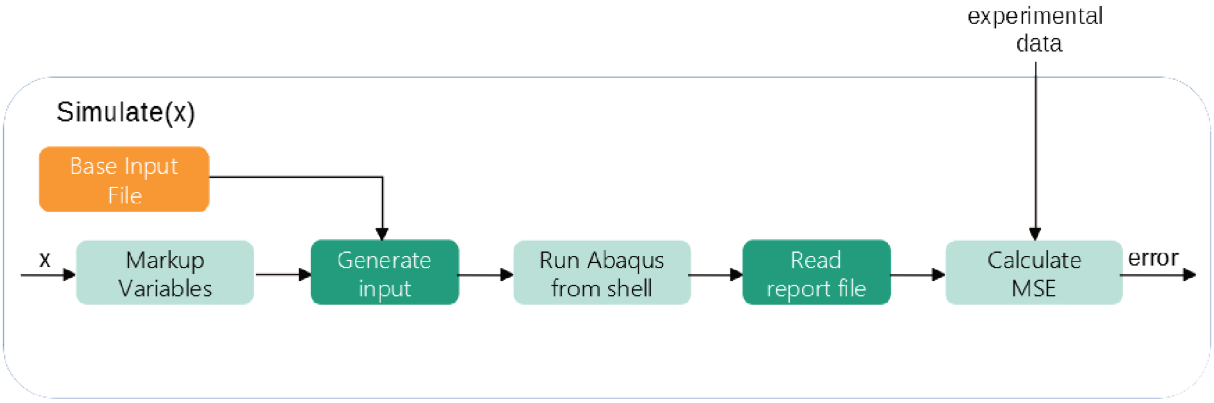
4.2.4 Calculate the error between experimental and simulation curves

Since both experimental and simulation data are stored in the same data structure, calculating the error between curves consists of applying a metric algorithm to calculate the error between curves. There are various metrics to measure the similarity between curves, such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and R-squared (MISHRA, 2019). MSE was chosen due to the fact that it is simple to calculate and it over-penalizes the model for big errors, since it squares the error.

4.2.5 Interface between Abaqus and optimization algorithms

Most implementations of optimization algorithms require a function that receives an array with the candidate solution and returns the cost function. To comply with this pattern, all of the latter features are sequentially integrated to generate the *simulate* function. The workflow of the *simulate* function can be seen in Figure 29.

Figure 29 – Workflow of *simulate* function.



Source – Original

For the optimization, derivative-free algorithms are used (as discussed in section 2.6). Nevergrad, a Python package created by Facebook, was chosen because it is specially focused on derivative-free optimization algorithms (FACEBOOK, 2019). In addition, the implementation of the algorithms already use a predefined set of parameters, thus avoiding the effort of testing multiple parameter setups. Since the simulations take long time, having to benchmark multiple algorithms with multiple parameter setups is a time-consuming task. This design choice is a key-point to the future analysis of the algorithms. Since there is a trade-off between parameter tuning and time, and the time factor was considered more relevant, it is not possible to claim a certain method to be inefficient based on the benchmark results.

The approach to choose the algorithm to be used was to perform a benchmark with a predefined set of parameters. Based on the results, the best algorithm was used for the rest of the experiments. Section 2.6 presented a set of derivative-free optimization algorithms, which were tested in the algorithm benchmark.

4.2.6 Helper functions

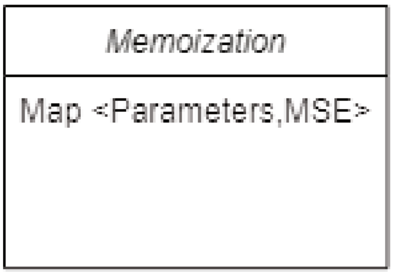
In addition to the features described before, a couple of helper functions were developed to facilitate the automation of the simulation. These functions are necessary for the identification process. However their implementations did not require any software modeling, therefore being called helper functions.

The first developed helper function was responsible for cropping an interval of the experimental data. This is required because the identification is not done throughout the whole experiment, but only during an specific step (e.g. in cooling experiments, the cooling step is the only interval that is useful for the identification).

Another helper function was built to fix any missing values in the database. In case there is any missing temperature measurement in the database, a linear interpolation is applied in order to fill the missing data.

Also, memoization was developed in order to prevent the optimization algorithm from simulating the same simulation more than once. Memoization is a process of storing the results of expensive function calls and returning the stored result if the function is called with the same input. Every time a simulation is done with a certain set of candidate parameters, the corresponding error result is saved into a JSON file. The structure of the JSON file is depicted in Figure 30.

Figure 30 – Memoization data structure.



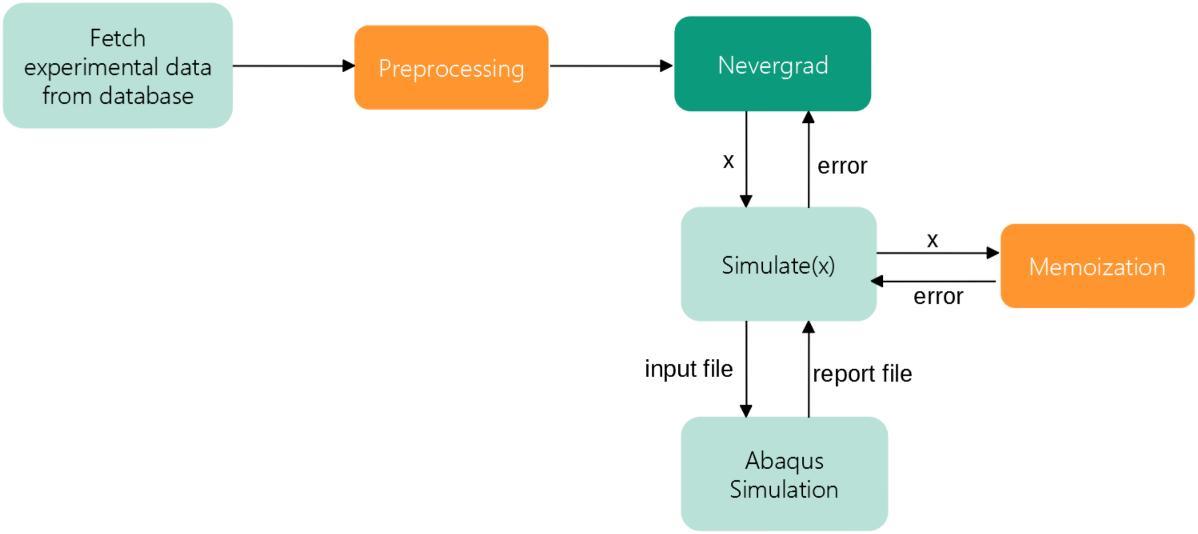
Source – Original

When there is a simulation request, the script will open the memoization file and check if those parameters were already tested. If so, the result will be fetched from the file. Otherwise, the simulation will be completed.

4.2.7 Complete Optimization Architecture

Integrating the previous features, the complete software architecture for identification can be set. Figure 31 presents the complete software architecture. A defined experiment is fetched from the database, followed by the preprocessing of the data (described in 4.2.6). After the preprocessing, the control of the application is transferred to Nevergrad, which calls the simulate function and receives the error. In case the error value exists in the memoization layer, no simulation is required. If not, the simulation is ran and the error is calculated from the Abaqus report file.

Figure 31 – Complete optimization architecture.



Source – Original

This section presented the main steps in the modelling of the cooling experiments, as well as the software development for integrating the simulation software with the optimization module. The complete architecture will be used in chapter 5 to identify the parameters for each type of experiments presented in section 3.3.2.

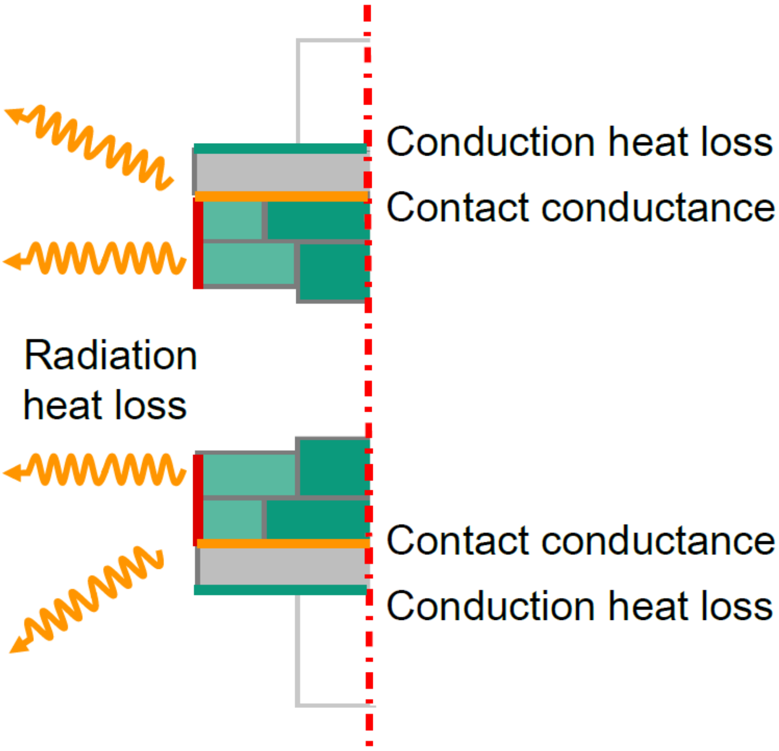
5 RESULTS AND ANALYSIS

The present section presents the algorithm benchmark proposed in section 4.2.5, the identification results and analysis for vacuum cooling experiments, in section 5.1, and for nitrogen cooling experiments, in section 5.2.

5.1 VACUUM COOLING EXPERIMENTS

This experiment consists in heating the machine (without glass preform) up to a high temperature. Afterwards, the machine is cooled down by vacuum. The main goal of this experiment is the identification of the system heat losses model. Within this experiment, the following thermal phenomena are involved: conduction heat loss, radiation heat loss, and contact conductance. The model for the vacuum cooling experiments can be seen in Figure 32.

Figure 32 – Thermal phenomena in vacuum cooling experiments.

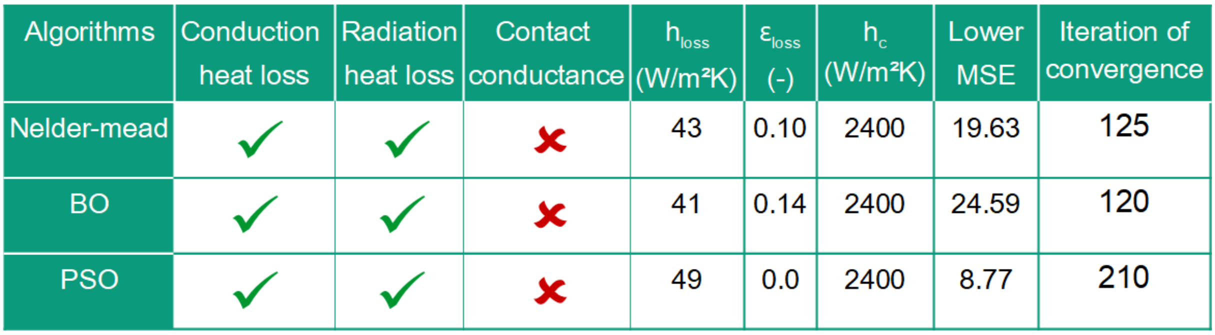


Source – Original

From these phenomena, the following thermal parameters can be identified: upper and lower conduction coefficients of heat loss (h_{loss}), emissivity of radiation heat loss (ϵ_{loss}) and contact conductance (h_c).

As explained in section 4.2.5, a benchmark was done in order to decide which algorithm would be used in the identification process. The benchmark consists in the optimization of the thermal parameters for the first model hypothesis in the vacuum cooling experiment. This hypothesis assumes that conduction coefficient (h_{loss}) and emissivity (ϵ_{loss}) are learned and the contact conductance remains at a constant value of 2400 $\frac{\mathrm{W}}{\mathrm{m^2K}}$. This value is the one defined in the current simulation and was identified in Vu et al. (2019). In order to reduce the complexity of the benchmark, the optimization was carried only using the lower mold temperature data. The results are presented in Figure 33 and show that the best MSE was reached by PSO. For this reason, it was decided that PSO would be used in the further experiments.

Figure 33 – Algorithm benchmark results.

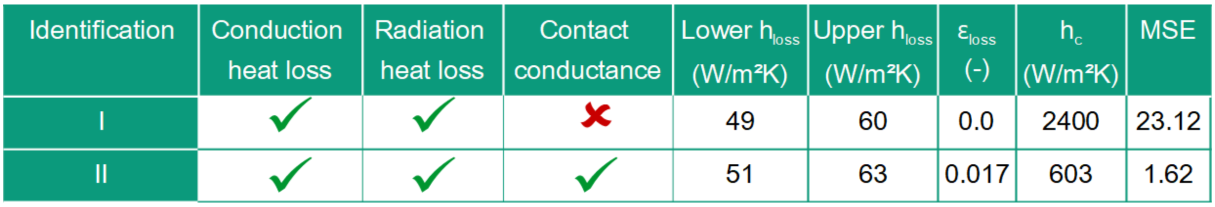


Algorithms	Conduction heat loss	Radiation heat loss	Contact conductance	h_{loss} (W/m²K)	ε_{loss} (-)	h_c (W/m²K)	Lower MSE	Iteration of convergence
Nelder-mead	✓	✓	✗	43	0.10	2400	19.63	125
BO	✓	✓	✗	41	0.14	2400	24.59	120
PSO	✓	✓	✗	49	0.0	2400	8.77	210

Source – Original

Even though PSO presented the best result, the MSE found in the first hypothesis was too high, meaning that the model does not represent the experimental data well. The second hypothesis consists in learning the contact conductance value as well. The comparison between hypotheses can be seen in Figure 34.

Figure 34 – Comparison between multiple vacuum cooling model hypotheses.



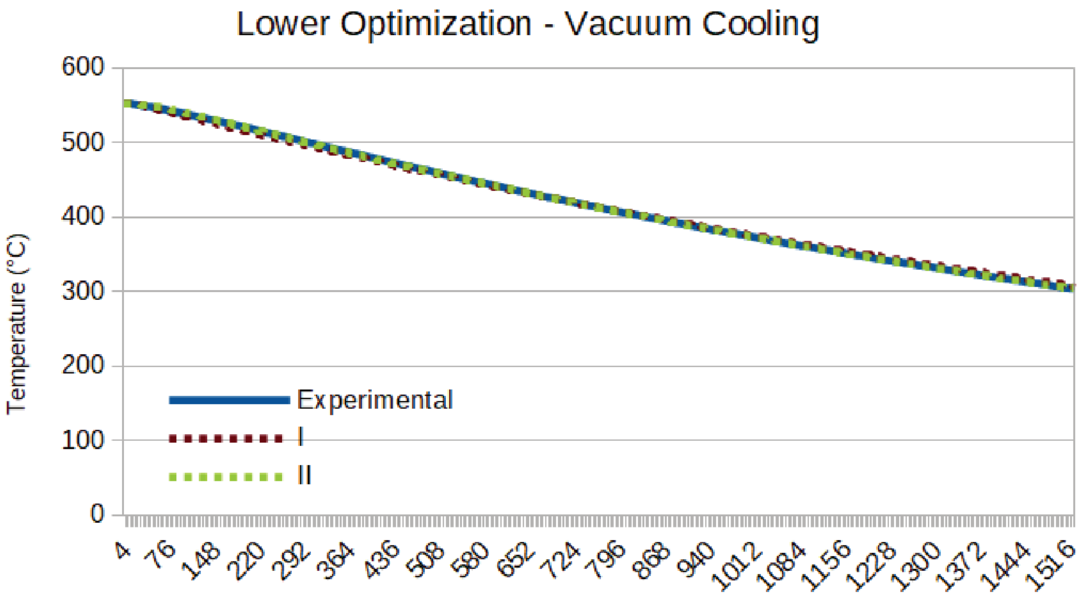
Identification	Conduction heat loss	Radiation heat loss	Contact conductance	Lower h_{loss} (W/m²K)	Upper h_{loss} (W/m²K)	ε_{loss} (-)	h_c (W/m²K)	MSE
I	✓	✓	✗	49	60	0.0	2400	23.12
II	✓	✓	✓	51	63	0.017	603	1.62

Source – Original

Hypothesis 2 yields a result that is able to describe the experimental curve. A comparison between the lower mold temperature curves can be seen in Figure 35 and the upper mold temperature curves can be seen in Figure 36.

The reason for identifying such a small contact conductance is due to the fact that there is vacuum between the surfaces, which reduces significantly the area of contact. In the measurements done in Vu et al. (2019), the mold chamber was not under vacuum, meaning that there was gas filling the gap between the holder and cooling plate. The presence of gas at the interface increases the contact area, therefore

Figure 35 – Comparison between lower mold experimental temperature curve and both hypotheses.



Source – Original

resulting in a bigger contact conductance. The identified value for gap conductance lies in the same range of the data presented in Ostrouchov et al. (2011), where a mixed numerical-experimental approach was used to identify the gap conductance between mold and glass under vacuum. The small emissivity value can be explained by the thermal isolation of the machine, which reflects most radiation back to the machine.

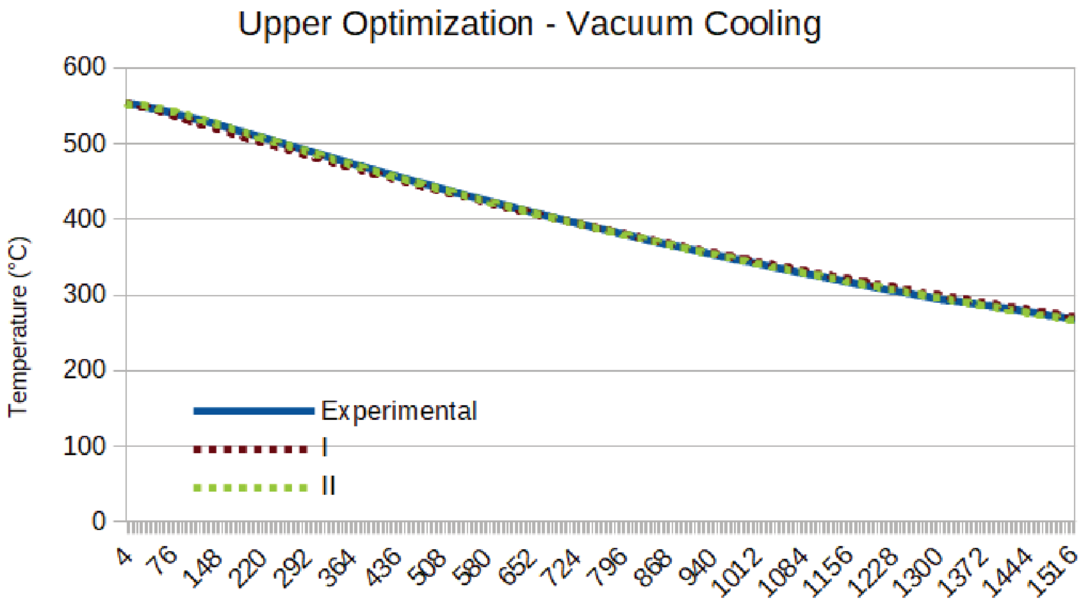
5.2 NITROGEN COOLING EXPERIMENTS

This experiment consists in heating the machine (without glass preform) up to a high temperature. Afterwards, the machine is cooled down by nitrogen (different nitrogen flow setups will be used). The main goal of this experiment is the identification of the convection heat losses, as well as their relation with the nitrogen flow operating point.

Within this experiment, the following thermal phenomena are involved: conduction heat loss, nitrogen convection heat loss, radiation heat loss, and contact conductance. The model for nitrogen cooling experiments can be seen in Figure 37.

From these phenomena, the following thermal parameters can be identified: upper and lower conduction/convection coefficients of heat loss (h_{loss}) (since both conduction and convection have linear boundary conditions and occur in the same surface, it is assumed that conduction and convection can be modeled together), emissivity of radiation heat loss (ϵ_{loss}), contact conductance (h_c), and natural convection (h_{nat}).

Figure 36 – Comparison between upper mold experimental temperature curve and both hypotheses.



Source – Original

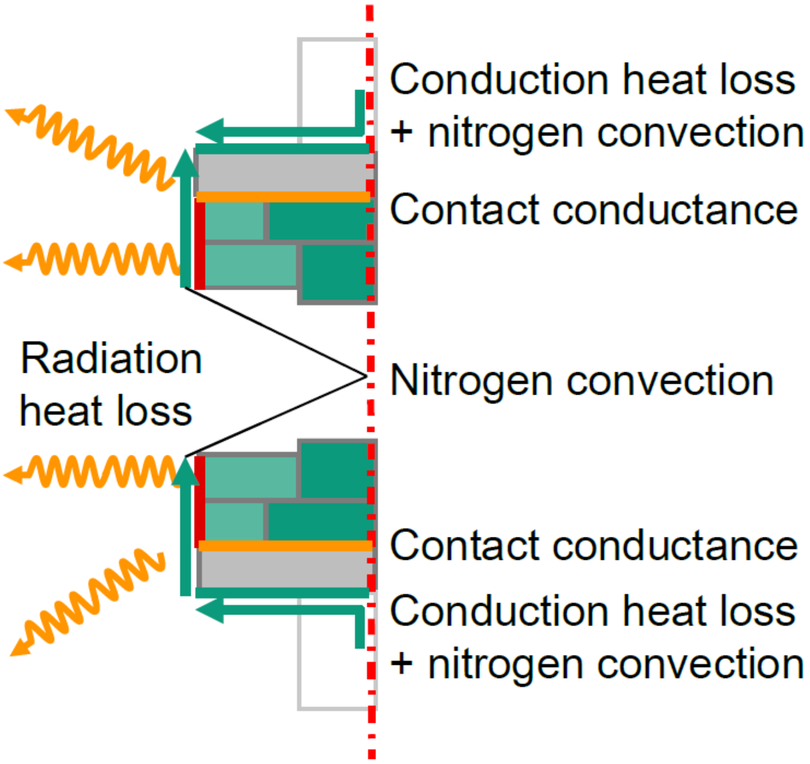
The first hypothesis consists in assuming that the parameters remain the same ones identified in the vacuum cooling model, except for the convection coefficient, which is expected to increase. As a second hypothesis, it is assumed that there is also natural convection occurring in the side of the mold assembly. Hypothesis 3 assumes that there is no natural convection and the contact conductance changes. At last, hypothesis 4 assumes that there is natural convection and contact conductance changes. The comparison between the hypotheses can be seen in Figure 38 for an experiment with a nitrogen flow of 60 l/min.

Hypothesis 4 yielded a good result that is able to represent the experimental curve. A comparison between the lower mold temperature curves can be seen in Figure 39 and the upper mold temperature curves can be seen in Figure 40.

Between the vacuum cooling model and the nitrogen cooling model, an increase in the convection/conduction coefficient was expected due to the fact that there is nitrogen gas flowing in the machine. The approach also was able to identify the presence of considerable natural convection in the side of the mold assembly.

Also, a great increase in the gap conductance value was found. This can be explained by the fact that the nitrogen gas fills the gap between the surfaces, therefore, increasing the area of heat transfer. Since the parameters in nitrogen cooling are operating-point dependent, other operating points were optimized. Figure 41 depicts the identification results for each nitrogen flow value.

Figure 37 – Thermal phenomena in nitrogen cooling experiments.



Source – Original

Figure 38 – Comparison between multiple cooling model hypotheses for a nitrogen flow of 60 l/min.

Identification	Conduction heat loss	Radiation heat loss	Contact conductance	Natural convection	Lower h_{loss} (W/m^2K)	Upper h_{loss} (W/m^2K)	ε_{loss} (-)	h_c (W/m^2K)	h_{nat} (W/m^2K)	MSE
I	✓	✗	✗	✗	895	1029	0.017	603	0	269.9
II	✓	✗	✗	✓	668	782	0.017	603	30	121.6
III	✓	✗	✓	✗	383	427	0.017	50,000	0	3.47
IV	✓	✗	✓	✓	350	393	0.017	34,856	15	1.62

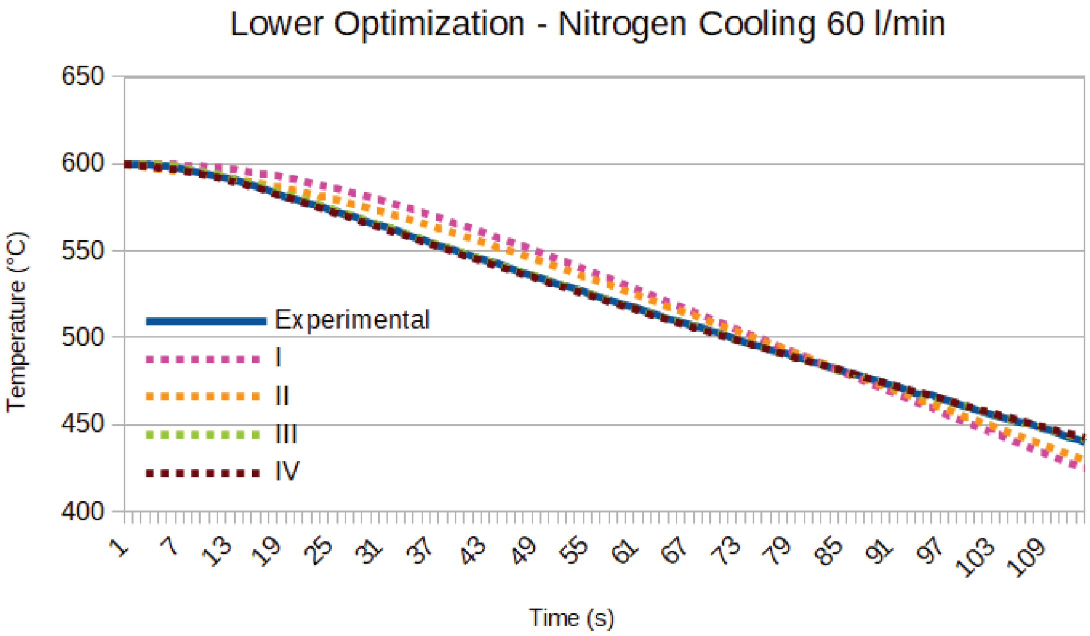
Source – Original

It is possible to calculate the equivalent thickness of a thin nitrogen film using the identified gap conductance using the equation

$$U = \frac{k}{x}, \tag{13}$$

where U is the conductance, k is the conductivity, and x is the thickness. Figure 42 presents the equivalent thickness for each operating-point presented in Figure 41. The values range from 0.73 μm to 3.49 μm. The surface flatness provided by the manufacturing tolerance is close to 1.8 μm. The explanation for the variance of x could be the

Figure 39 – Lower mold temperature comparison between multiple cooling model hypotheses for a nitrogen flow of 60 l/min.



Source – Original

change of thermal conductivity of nitrogen in each operating-point, due to the pressure increase. Also, the pressure increase can result in a decrease of the gap in the interface between the cooling plate and holder.

Figure 43 and Figure 44 depict the lower and upper convection coefficient as a function of the nitrogen flow, respectively. Both graphs have a positive derivative, which means that increasing the nitrogen flow rate causes an increase in the convection coefficient. These results agree with the theory, since a greater flow of nitrogen is able to remove more heat from the surface. Graphically, the relationship between nitrogen flow and convection coefficient appears linear between 10 l/min and 60 l/min. A regression analysis was done using the identified data, reaching a coefficient of determination of 0.96 for the lower convection coefficient using equation

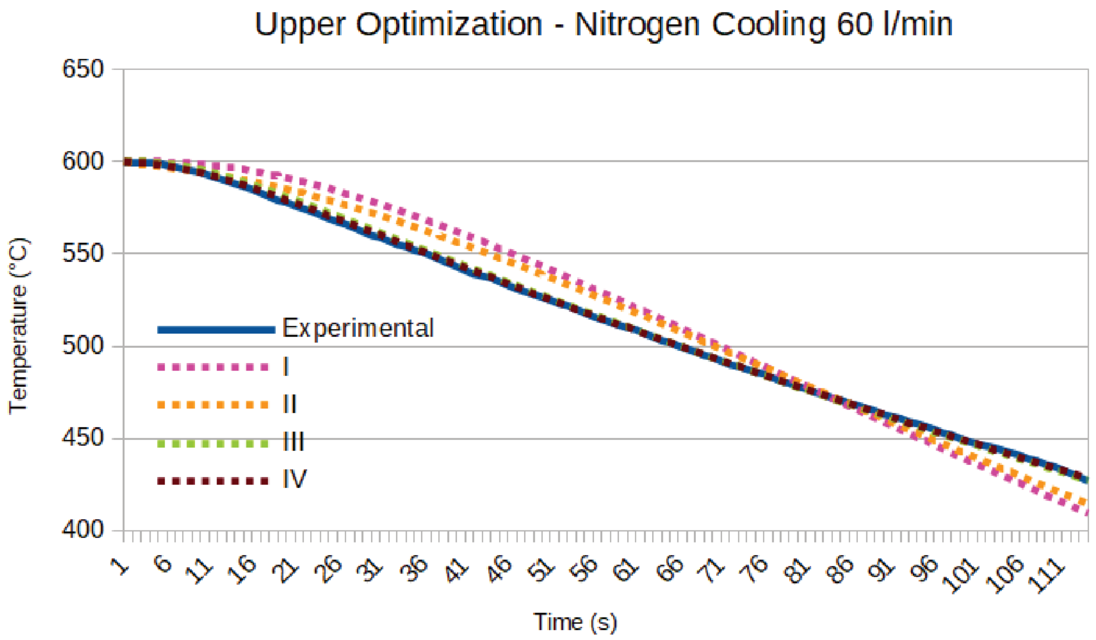
$$lowerh_{loss}(x) = 4.22x + 81.25, \tag{14}$$

meaning that 96% of the variance in the convection coefficient can be explained by the nitrogen flow values. For the upper convection coefficient regression, a coefficient of determination of 0.99 was found using equation

$$upperh_{loss}(x) = 5.02x + 82.64. \tag{15}$$

Figure 45 depicts the lateral convection coefficient as a function of the nitrogen flow. The graph also presents a positive derivative (excluding the point between 10 l/min

Figure 40 – Upper mold temperature comparison between multiple cooling model hypotheses for a nitrogen flow of 60 l/min.



Source – Original

Figure 41 – Nitrogen cooling identification results for each operating-point.

Operating-Point (l/min)	Lower h_{loss} (W/m^{2}K)	Upper h_{loss} (W/m^{2}K)	h_{nat} (W/m^{2}K)	h_c (W/m^{2}K)	Total MSE
60	350	393	15	34856	1.62
40	224	269	10	15993	2.16
20	164	180	5	8566	1.45
10	135	141	5	7294	1.65

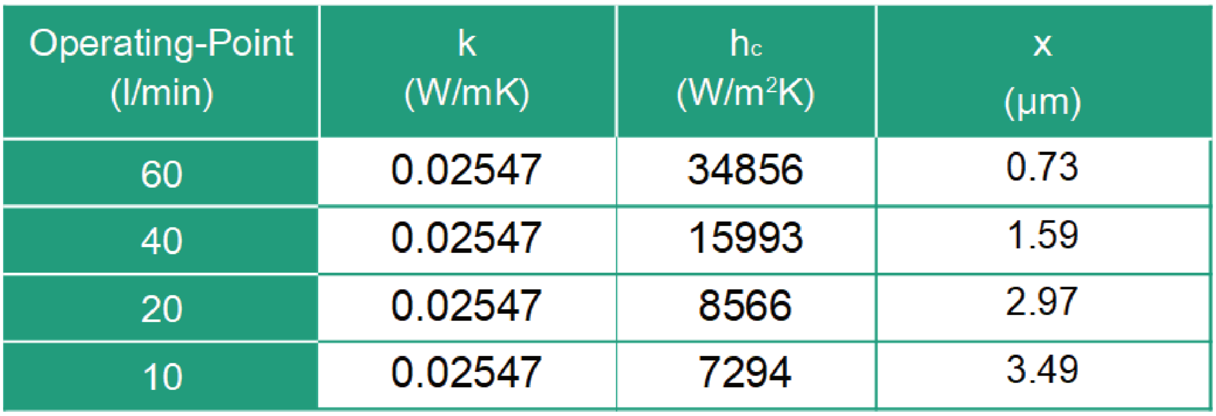
Source – Original

and 20 l/min, meaning that increasing the nitrogen flow causes an increase in the lateral heat extraction. The points corresponding to the nitrogen flow rate of 10 l/min and 20 l/min present the same lateral convection coefficient. This might be a numerical error and further experiments with these operation points are necessary to investigate these results. A regression analysis was also done, reaching a coefficient of determination of 0.96 using equation

$$h_{nat}(x) = 0.21x + 1.86. \tag{16}$$

Figure 46 depicts the gap conductance as a function of the nitrogen flow. The graph also presents a positive derivative, meaning that increasing the nitrogen flow causes an increase in the gap conductance. This means that the heat transfer in the

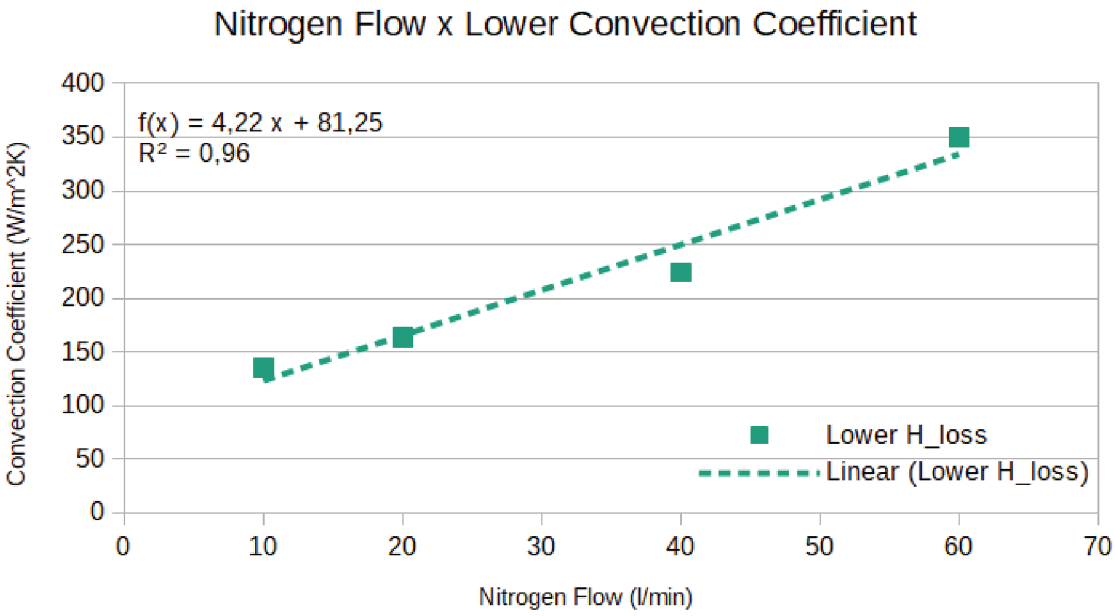
Figure 42 – Equivalent thickness of a thin nitrogen film for each nitrogen flow value.



Operating-Point (l/min)	k (W/mK)	h_c (W/m^{2}K)	x (µm)
60	0.02547	34856	0.73
40	0.02547	15993	1.59
20	0.02547	8566	2.97
10	0.02547	7294	3.49

Source – Original

Figure 43 – Lower convection coefficient in function of nitrogen flow.



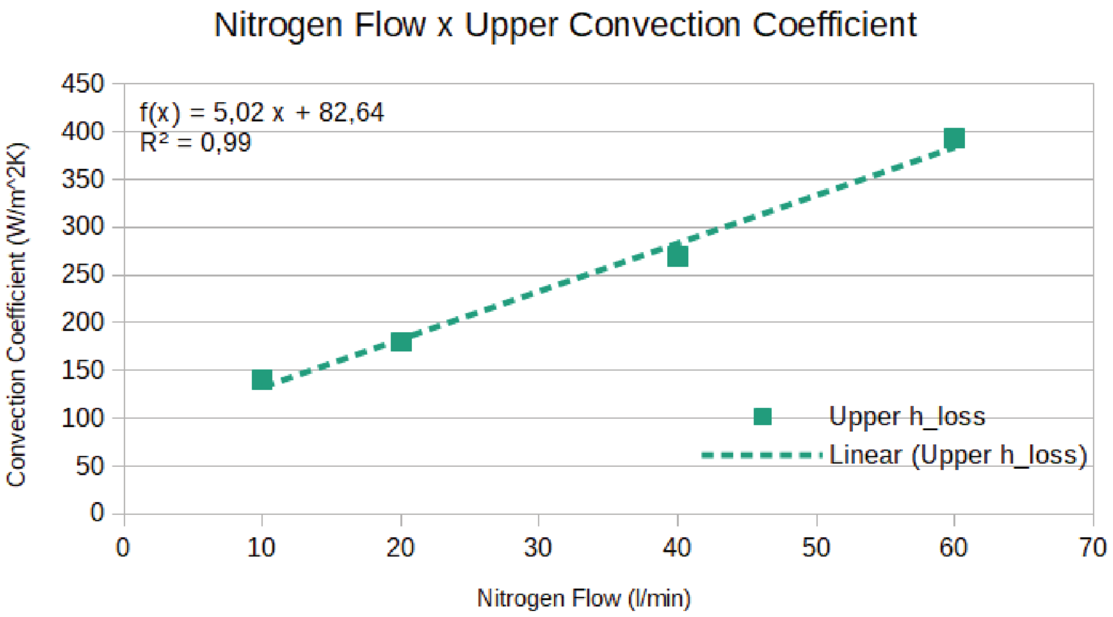
Source – Original

interface increases by increasing the flow of nitrogen in the machine. The relationship between gap conductance and nitrogen flow rate does not appear linear. Thus, a second-order polynomial was used. A regression analysis was done and the coefficient of determination found was 1.0 using equation

$$h_c(x) = 12.53x^2 - 334.13x + 9679.20. \qquad (17)$$

It is important to point that the identified parameters only represent numerical optima and that further experiments in the same operating-points need to be carried out and compared to the identified parameters. This method may determine whether

Figure 44 – Upper convection coefficient in function of nitrogen flow.



Source – Original

the current model is physically reliable and is able to describe the experimental curves. Also it would be interesting to analyze more operating-points to have a better regression analysis.

5.3 EXPERIMENTAL EVALUATION

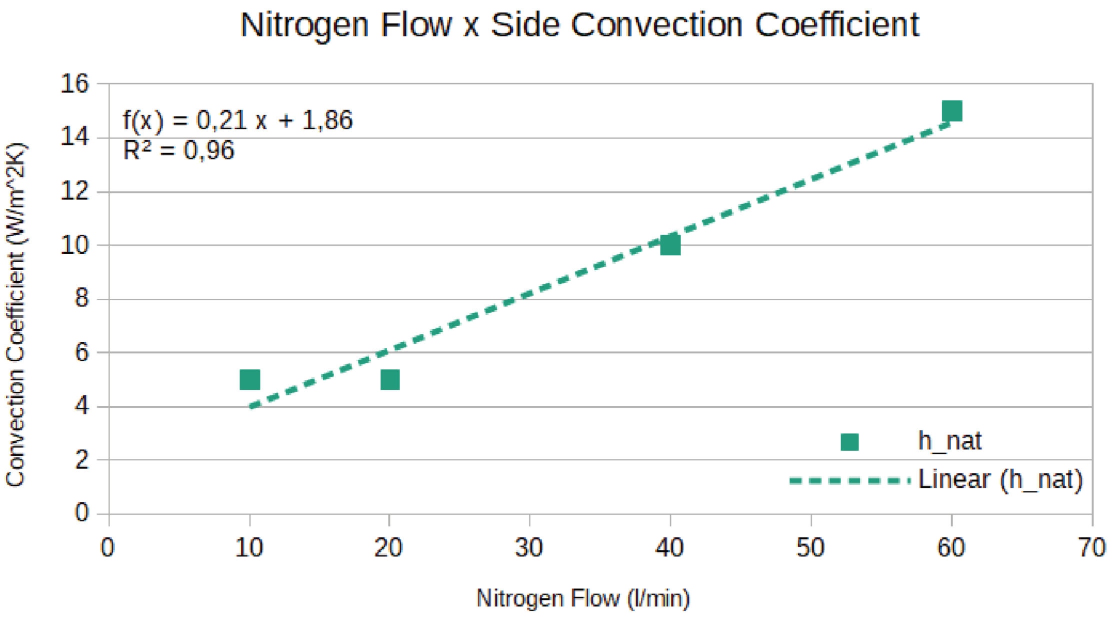
This section presents the experimental evaluation of the models identified in section 5.2. The approaches for the evaluation comprise evaluating known operating-points, in section 5.3.1, and unknown operating-points, in section 5.3.2.

5.3.1 Known operating-point analysis

In section 5.2, a set of parameters were identified for the nitrogen cooling experiments in each operating-point. The best MSE for these experiments was found at 20 l/min. For this reason, the experiment was repeated in this operating-point and the approach was to evaluate if the parameters identified previously would be able to describe the new experiment. Figure 41 shows that the identified values in the experiment were 164 $\frac{\text{W}}{\text{m}^2\text{K}}$ for the lower convection coefficient, 180 $\frac{\text{W}}{\text{m}^2\text{K}}$ for the upper convection coefficient, 5 $\frac{\text{W}}{\text{m}^2\text{K}}$ for the natural convection, and 8566 $\frac{\text{W}}{\text{m}^2\text{K}}$ for the gap conductance. Thus, these parameters will be used to compare the simulation data with the new experimental data.

The MSE found for the new experiment was 4.13, while 1.62 was found for the

Figure 45 – Lateral convection coefficient in function of nitrogen flow.



Source – Original

first experiment. This difference might be due to small changes in the assembly of the mold, in the temperature of the nitrogen gas or even in the ambient temperature. Figure 47 and Figure 48 show the comparison between the lower and upper mold temperature of the simulation and the respective experimental mold temperature.

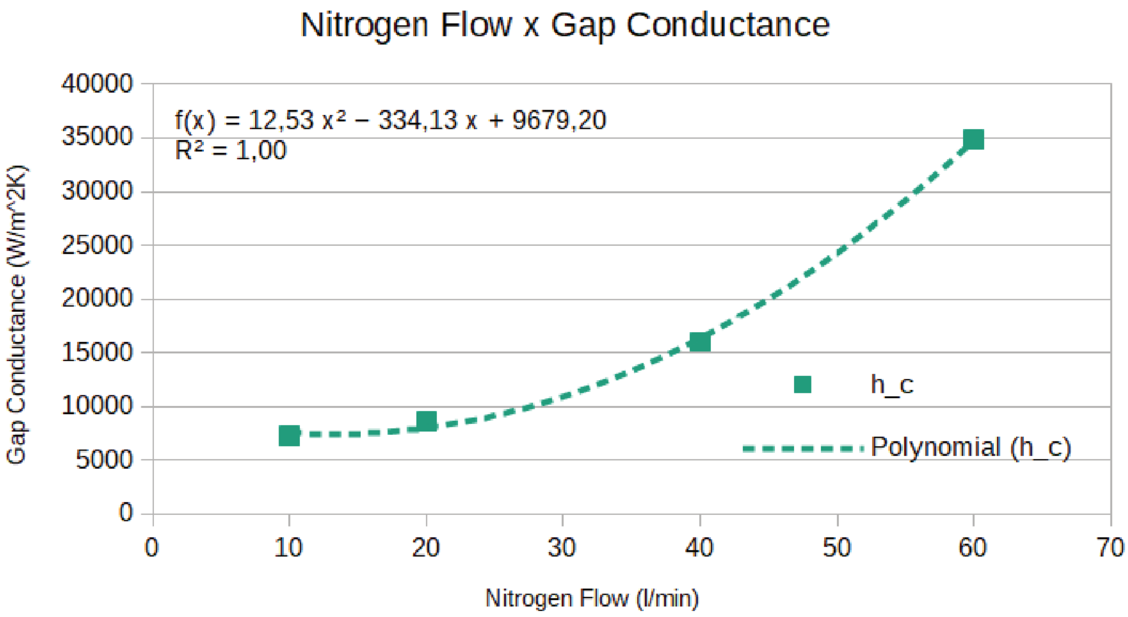
Finally, these results prove that the identified parameters are able to represent the experimental curves and that the model is physically reliable in the evaluated operating-point. Further experiments in the remaining known operating-points are necessary to evaluate if the parameters can also describe the respective experimental data.

5.3.2 Unknown operating-point analysis

In section 5.2, many operating-points had its corresponding thermal parameters identified and a regression analysis was done. In order to analyze if the fitted curves are able to describe how the thermal parameters vary with respect to the operating-point, the proposed method was to run an experiment in an unknown operating-point analysis. The selected operating-point to do this analysis was 30 l/min, due to the fact that it is located in the mid-range of the previously identified operating-points. Using the equations from section 5.2, the parameters to be tested were calculated. These are shown in Table 3.

With the fitted parameters, the found MSE was 50.26. This value indicates that the fitted parameters are not able to describe the experimental curves well. Figure 49

Figure 46 – Gap conductance in function of nitrogen flow.



Source – Original

Table 3 – Fitted thermal parameters for a nitrogen flow of 30 l/min.

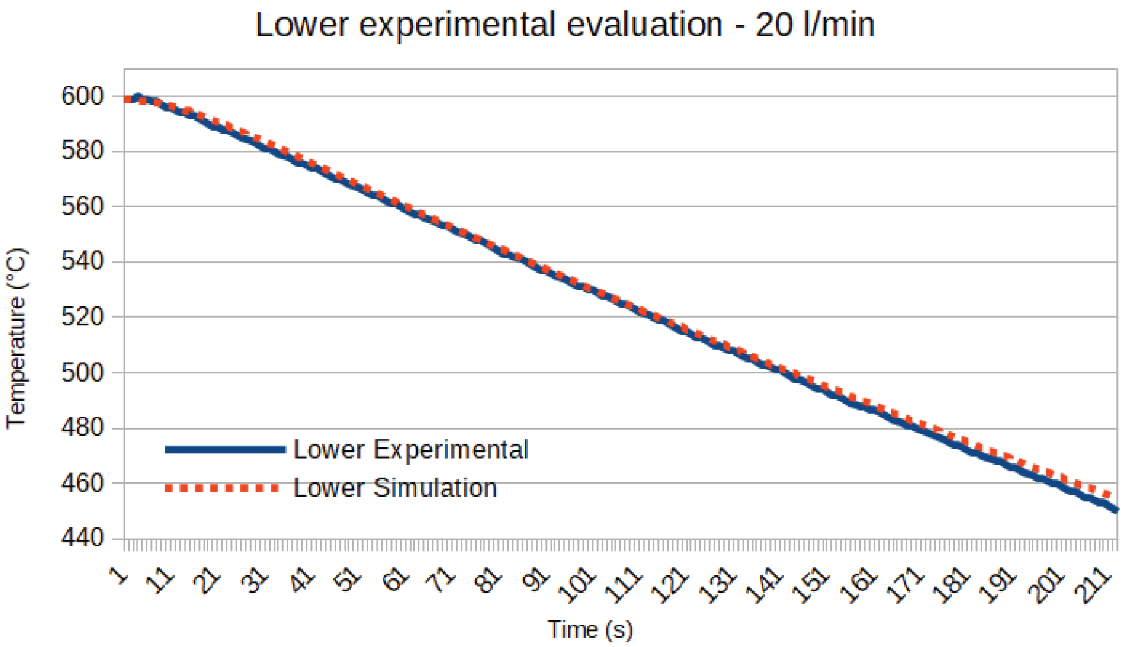
Parameter	Value ($\frac{W}{m^2K}$)
Lower h_{loss}	207.85
Upper h_{loss}	233.24
h_{nat}	8.16
h_c	10932.3

Source – Original

shows the comparison between experimental data and simulation data in the upper and lower mold. The greatest error is found in the upper mold temperature, meaning that the regression model for the upper mold is worse fitted than for the lower mold.

However, this result does not completely exclude the regression approach. Since the regression was done using few observations, the fitted curves are largely affected by noise and might not be statistically significant. In order to evaluate this, an optimization job was done close to the point suggested by the regression analysis. This approach was able to find a set of parameters that yielded a MSE of 1.03. This means that the regression was suggesting a good region, however due to noise and few observations, the suggested point was yielding a poor MSE. A comparison between the fitted variables and the later optimized variables can be seen in Table 4. It is possible to see that the difference between most parameters is within 10%. Therefore the hypothesis that the regression was pointing to the right region is validated. The only big difference between the values occurs with the natural convection value, meaning that the relation might

Figure 47 – Lower experimental evaluation at 20 l/min.



Source – Original

not be linear (specially when analyzing the fact that the natural convection value for 10 l/min and 20 l/min was 5 $\frac{\text{W}}{\text{m}^2\text{K}}$). Figure 50 and Figure 51 show respectively the lower and upper temperature comparisons between experimental and simulation data using the optimized parameters.

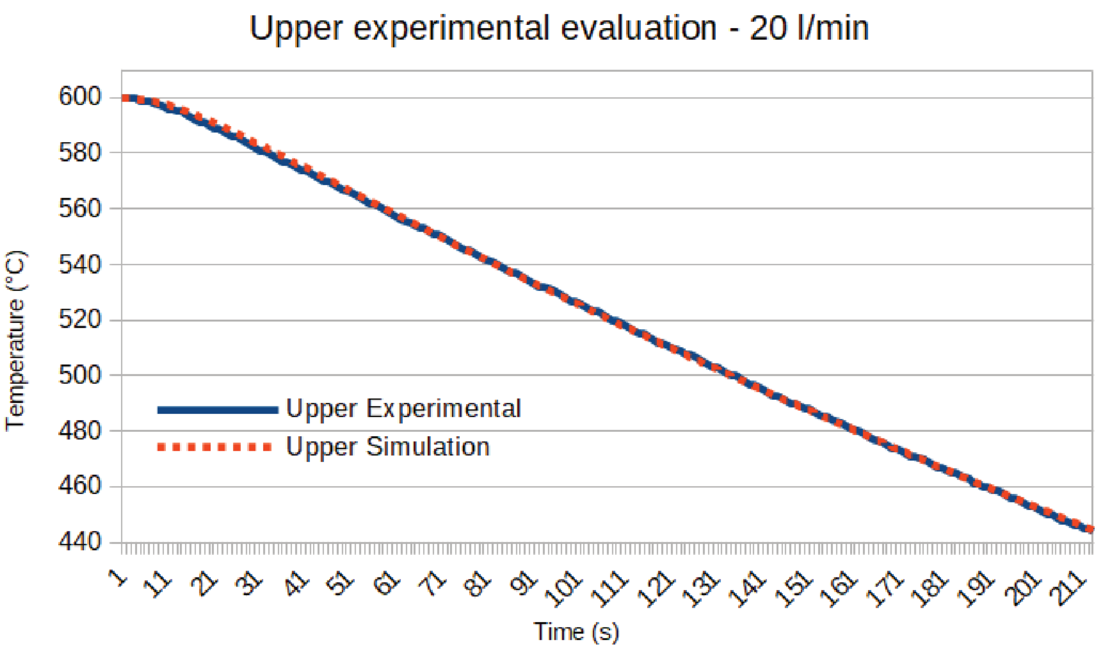
Table 4 – Comparison between fitted and optimized thermal parameters for a nitrogen flow of 30 l/min.

Parameter	Fitted ($\frac{\text{W}}{\text{m}^2\text{K}}$)	Optimized ($\frac{\text{W}}{\text{m}^2\text{K}}$)
Lower h_{loss}	207.85	205
Upper h_{loss}	233.24	219
h_{nat}	8.16	6
h_c	10932.3	9800

Source – Original

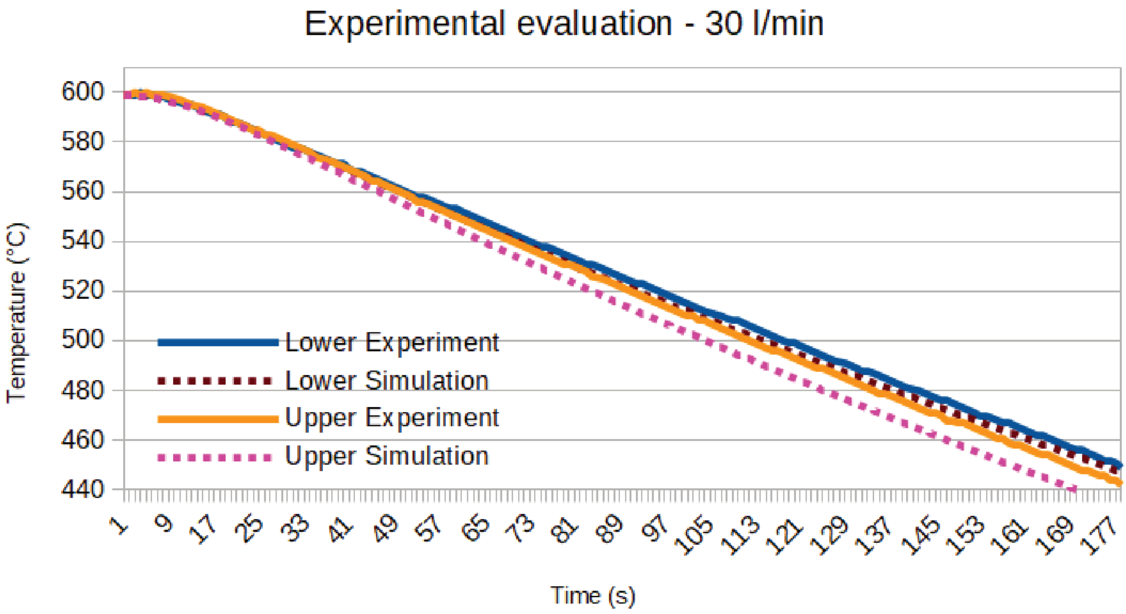
Finally, the presented results show that the regression analysis indicates the right region for the analyzed operating-point. However, due to few observations, the model is susceptible to noise and suggests an incorrect point. After running an optimization job in the region suggested by the regression model, a set of identified parameters were found. These were able to well describe the experimental data. Thus, further operating-points need to have its parameters identified in order to increase the number of observations in the regression analysis.

Figure 48 – Upper experimental evaluation at 20 l/min.



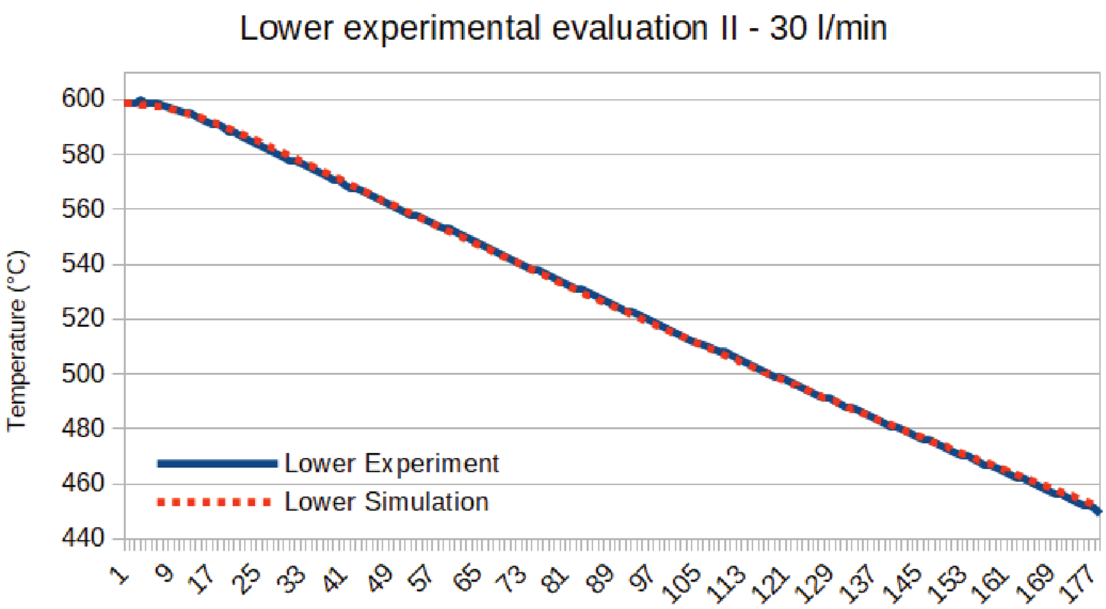
Source – Original

Figure 49 – Experimental evaluation at 30 l/min with fitted parameters.



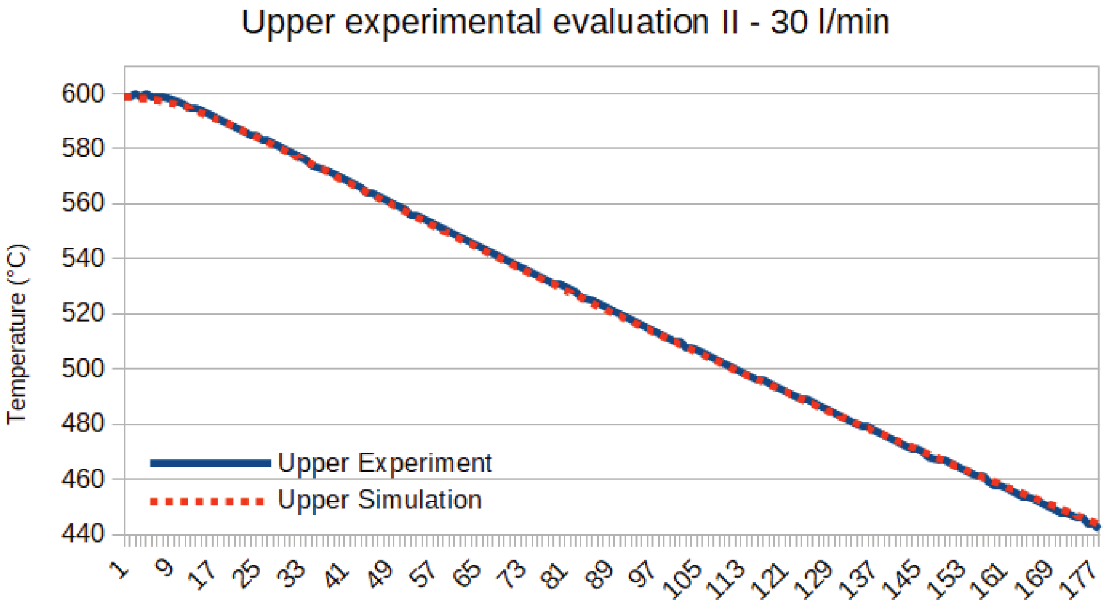
Source – Original

Figure 50 – Lower experimental evaluation at 30 l/min with optimized parameters.



Source – Original

Figure 51 – Upper experimental evaluation at 30 l/min with optimized parameters.



Source – Original

6 CONCLUSION AND FURTHER WORK

This report presented the development of a simulation-based optimization strategy to identify unknown parameters for a precision glass molding thermal model. The project consisted of an extensive analysis of the current thermal simulation, a software implementation for the identification of parameters and a further analysis of the results.

The suggested approach proved to be feasible and presented good results for vacuum cooling and nitrogen cooling experiments. The results agree with theory and experimental data. The identified gap conductance showed an agreement with papers addressing the same topic, however using different identification approaches. Also, an experimental evaluation of the nitrogen cooling models was made for known and unknown operating-points.

Due to the limited amount of time, as well as restricted access to the laboratory at the time of this work's development (due to the COVID-19 outbreak), some of the experimental part of this project was delayed. Heating experiments were not addressed since the identification of cooling experiments were simpler and would allow the experimental evaluation of the proposed method. Also, heating experiments would require a slightly different approach for the identification. Since the estimation of the lamp temperature consists of a dynamic identification (meaning that the temperature of lamps change with time), the proposed approach would be to assume the system as a *n*th order system and the goal would be to identify its coefficients, which are static. In case the lamp system is non-linear, these coefficients will change based on the operating-point and the same approach used in the nitrogen cooling experiments can be used. Within the constraints encountered during the time of this work, the main goals were achieved.

Further work consists of evaluating the identified models with experimental data in the remaining operating-points. Also further operating-points for nitrogen cooling must have its parameters identified in order to increase the number of observations for the regression analysis. After validating the regression approach in multiple unknown operating-points, heating experiments can be addressed.

The development of the present project was a very enriching experience, since it allowed the author to deal with both practical and theoretical aspects related to the Control and Automation Engineering Course in an international research institute. The basis offered by the course, in addition to the guidance from both company supervisor and university supervisor, were essential for the completion of the project.

REFERENCES

24CHEMICALRESEARCH. **Precision Glass Molding Market Insights, Forecast to 2025**. North Main Road Koregaon Park, Pune, India - 411001.: 24chemicalresearch, 2019. Available from: `https://www.24chemicalresearch.com/reports/33837/precision-glass-molding-2025-611`.

ASTER, R.C.; BORCHERS, B.; THURBER, C.H. **Parameter Estimation and Inverse Problems**. [S.l.]: Elsevier Science, 2018. ISBN 9780128134238. Available from: `https://books.google.de/books?id=VuRyDwAAQBAJ`.

BERGMAN, T.L. et al. **Introduction to Heat Transfer**. [S.l.]: Wiley, 2011. ISBN 9780470501962. Available from: `https://books.google.de/books?id=YBaNaLurTD4C`.

DENG, Geng. **Simulation-based optimization**. 2007. PhD thesis – University of Wisconsin–Madison.

FACEBOOK, AI Research. **Nevergrad - A gradient-free optimization platform**. [S.l.: s.n.], 2019. Accessed: 2020-07-01. Available from: `https://facebookresearch.github.io/nevergrad/`.

FARGHALY, Hala Ahmed Salama Ashry. **EXPERIMENTAL INVESTIGATION AND MODELING OF FRACTURE FOR MOLDED INFRARED GLASS OPTICS**. 2019. MA thesis – Clausthal University of Technology.

FELIPPA, Carlos A. Introduction to finite element methods. **University of Colorado**, v. 885, 2004.

FRAZIER, Peter I. **A Tutorial on Bayesian Optimization**. [S.l.: s.n.], 2018. arXiv: `1807.02811 [stat.ML]`.

FUJILLOY. **Fujilloy Lineup**. [S.l.: s.n.], 2008. Accessed: 2020-07-01. Available from: `http://www.fujilloy.co.th/wp-content/uploads/PDF/Material%20&%20Manufacturing%20Technology%20Eng.pdf`.

GANJI, Davood Domairry; SABZEHMEIDANI, Yaser; SEDIGHIAMIRI, Amin. Chapter 3 - Radiation Heat Transfer. In: GANJI, Davood Domairry; SABZEHMEIDANI, Yaser; SEDIGHIAMIRI, Amin (Eds.). **Nonlinear Systems in Heat Transfer**. [S.l.]: Elsevier, 2018. P. 105–151. ISBN 978-0-12-812024-8. DOI: `https://doi.org/10.1016/B978-0-12-812024-8.00003-5`.

HUTTON, D.V. **Fundamentals of Finite Element Analysis**. [S.l.]: McGraw-Hill, 2003. (Engineering Series). ISBN 9780072922363. Available from: `https://books.google.de/books?id=CV2HLsuNMtsC`.

IPT, Fraunhofer. **Facts and Figures Fraunhofer IPT**. Germany: Fraunhofer IPT, 2020. `https://www.fraunhofer.de/en/about-fraunhofer/profile-structure/facts-and-figures.html`. Accessed: 2020-07-01.

IQBAL, Waqas. **IDENTIFYING THE OPTIMUM PROCESS PARAMETERS OF PRECISION GLASS MOLDING FOR ASPHERICAL LENSES**. 2009. MA thesis – Clemson University. Available from: `https://tigerprints.clemson.edu/all_theses/687`.

KANNAN, Saravanan. **MODELING THE COOLING PHASE OF THE LENS MOLDING PROCESS**. 2009. MA thesis – Clemson University. Available from: `https://tigerprints.clemson.edu/all_theses/705`.

KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: PROCEEDINGS of ICNN'95 - International Conference on Neural Networks. [S.l.: s.n.], 1995. 1942–1948 vol.4.

KLEINERMANN, Jean-Pascal; PONTHOT, Jean-Philippe. Parameter identification and shape/process optimization in metal forming simulation. **Journal of Materials Processing Technology**, v. 139, n. 1, p. 521–526, 2003. IMCC2000 Vol. 2 S.I. ISSN 0924-0136. DOI: `https://doi.org/10.1016/S0924-0136(03)00530-2`.

KLOCKE, Fritz; STAASMEYER, Jan-Helge. Lecture for Precision Glass Molding. Unpublished Presentation. [S.l.], 2015.

LECOMPTE, David et al. Mixed numerical–experimental technique for orthotropic parameter identification using biaxial tensile tests on cruciform specimens. **International Journal of Solids and Structures**, v. 44, n. 5, p. 1643–1656, 2007. ISSN 0020-7683. DOI: `https://doi.org/10.1016/j.ijsolstr.2006.06.050`.

LEVENSPIEL, O. **Engineering Flow and Heat Exchange**. [S.l.]: Springer US, 2014. ISBN 9781489974549. Available from: `https://books.google.de/books?id=f%5C_CbBQAAQBAJ`.

LI, Ziyuan. **Experimental and numerical studies of the J-integral in the temperature and strain-rate dependent fracture of glass material**. 2016. MA thesis – RWTH Aachen.

LIU, G. **Modeling Fracture Behavior in Precision Glass Molding**. [S.l.]: Apprimus Wissenschaftsverlag, 2018. ISBN 9783863596422. Available from: `https://books.google.de/books?id=JV19DwAAQBAJ`.

MISHRA, Divyanshu. **Regression: An Explanation of Regression Metrics And What Can Go Wrong**. [S.l.: s.n.], 2019. Available from: `https://towardsdatascience.com/regression-an-explanation-of-regression-metrics-and-what-can-go-wrong-a39a9793d914`.

MOELLER, Dietmar P. F. Parameter Identification of Dynamic Systems. In: MATHEMATICAL and Computational Modeling and Simulation: Fundamentals and Case Studies. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004. P. 257–310. ISBN 978-3-642-18709-4. DOI: 10.1007/978-3-642-18709-4_5.

NGUYEN, Anh Tuan; REITER, Sigrid; RIGO, Philippe. A review on simulation-based optimization methods applied to building performance analysis. **Applied Energy**, v. 113, Jan. 2014. DOI: 10.1016/j.apenergy.2013.08.061.

OSTROUCHOV, Christopher et al. A combined numerical and experimental approach to measuring gap conductance for precision glass molding. **Materials Science and Technology Conference and Exhibition 2011, MS and T'11**, v. 2, p. 1729–1736, Jan. 2011.

POLI, Riccardo; KENNEDY, James; BLACKWELL, Tim. Particle swarm optimization. **Swarm intelligence**, Springer, v. 1, n. 1, p. 33–57, 2007.

SHI, Y.; EBERHART, R. C. Empirical study of particle swarm optimization. In: PROCEEDINGS of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406). [S.l.: s.n.], 1999. 1945–1950 vol. 3.

SINGER, S.; NELDER, J. Nelder-Mead algorithm. **Scholarpedia**, v. 4, n. 7, p. 2928, 2009. revision #91557. DOI: 10.4249/scholarpedia.2928.

SONG, S; YOVANOVICH, MM; NHO, K. Thermal gap conductance-Effects of gas pressure and mechanical load. **Journal of thermophysics and heat transfer**, v. 6, n. 1, p. 62–68, 1992.

VU, Anh Tuan et al. Experimental investigation of contact heat transfer coefficients in nonisothermal glass molding by infrared thermography. **Journal of the American Ceramic Society**, v. 102, n. 4, p. 2116–2134, 2019. DOI: 10.1111/jace.16029.

WIKIPEDIA. **Fraunhofer-Institut für Produktionstechnologie**. [S.l.: s.n.], 2010. Accessed: 2020-01-07.

WINDHOEK. **A Fascinating Historical Time Line on Lenses**. [S.l.: s.n.], 2015. Accessed: 2020-07-01. Available from: https://www.windhoek-optics.com/index.php/2015-09-03-10-59-08/historical-timeline-on-lenses.

YAN, Jiwang et al. Modeling high-temperature glass molding process by coupling heat transfer and viscous deformation analysis. **Precision Engineering**, v. 33, n. 2, p. 150–159, 2009. ISSN 0141-6359. DOI: https://doi.org/10.1016/j.precisioneng.2008.05.005.