Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

Trym Arve Lund Gabrielsen

Extrusion Cycle Optimization

Application of Direct Collocation to the Extrusion Process

Master's thesis in Cybernetics and Robotics
Supervisor: Lars Struen Imsland
June 2023

NTNU
Norwegian University of
Science and Technology

Trym Arve Lund Gabrielsen

Extrusion Cycle Optimization

Application of Direct Collocation to the Extrusion Process

Master's thesis in Cybernetics and Robotics
Supervisor: Lars Struen Imsland
June 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

Abstract

Through developing precise extrusion process control technology, the *High Precision Extrusion Temperature Control through Digital Technology (ExtruTeC)* research project sets out to meet increasing demands of aluminium profiles by allowing extrusion of higher quality product and increasing production.

As part of ExtruTeC, the work in this thesis is focused on conceptualizing a practical overall extrusion cycle optimization layer, as part of an extrusion process control hierarchy. Furthermore, a partial implementation of the optimization layer is covered, as means of testing the method and implementation, and thus proving the effectiveness of the proposed extrusion cycle optimization layer.

There are several works focused on optimizing the extrusion process, many of which concentrate on optimizing the ram speed curve over the extrusion phase. The idea of the extrusion cycle optimization scheme is to take the entire extrusion process into account; the heating-, transit-, and extrusion phase, and optimize the initial extrusion phase billet temperature, and subsequently the heating phase reference point, in order to optimize the extrusion period and increase production rate.

The extrusion cycle optimization problem is simplified by separating the problem into several smaller problems, representing the different phases of the extrusion process, and arguing for its equivalence with respect to optimality. The optimization of the extrusion phase is implemented as a nonlinear program (NLP), with the use of direct collocation. The implementation is made possible because of a progressor transformation of the model dynamics, from time to extrusion length. This transformation is shown to yield several benefits, such as effortless implementation of time varying and discontinuous extrusion model equations, and a great reduction of the number of decision variables of the respective NLP.

Upon solving the extrusion phase optimization problem in the form of an NLP, for several configurations, it is found that by optimizing the initial taper, as well as the ram speed and coolant valve opening, the proposed method may potentially reduce the extrusion time by up to 50% from a non-optimized process. Additionally, it is found that introducing initial taper optimization may yield a reduction of 26% from the case of only ram speed and coolant flow optimization, and that introducing coolant flow optimization from a case of only ram speed optimization may reduce the extrusion time by up to 23%. The results, however, are not verified experimentally, and are contingent on the applicability of the implemented extrusion model. The model used in this thesis is assumed reasonable based on crude comparisons to the reference model, upon which this model is based.

Sammendrag

Ved å utvikle presis kontollteknologi for ekstruderingsprosesser, skal forskningprosjektet *High Precision Extrusion Temperature Control through Digital Technology (ExtruTeC)* møte den økende etterspørselen av høykvalitetsprofiler ved å øke produktkvaliteten og produksjonsraten.

Som en del av ExtruTeC, tar arbeidet i denne oppgaven for seg et forslag til hvordan man kan bygge opp et overordnet optimaliseringlag av ekstrudeingssyklusen, som den del av at et kontrollhierarki for ekstruderingprosessen. I tillegg, er en sentral del av denne løsningen implementert for å både bekrefte at implementeringen av denne metoden er mulig og praktisk, og for å produsere resultater som understreker verdien av å optimalisere hele ekstruderingssyklusen.

Det er flere studier som fokuserer på å optimalisere ekstruderingprosessen, hvorav mange tar for seg optimalisering av ramhastiheten som funksjon av tid, over ekstruderingsfasen. Idéen til et optimaliseringslag av ekstruderingsskylusen, er å tilrettelegge for kortest mulig ekstruderingstid ved å optimalisere den initielle temperaturprofilen i billeten i starten av ekstruderingsfasen, ved å ta hensyn til hele syklusen, som innebærer varmefasen, transportfasen og ekstruderingsfasen.

Det overordnede syklus-optimaliseringsproblemet er forenklet ved å dele problemet inn i flere, mindre problem, som representerer de enkelte fasene av ekstruderingsprosessen, for så og løse dem bakover. Det vil si at ekstruderingsfasen optimaliseres først, så transportfasen, og til slutt varmefasen. Under noen antagelser agrumenteres det for at denne forenklingen er ekvivalent til det samlede problemet. Optimalisering av ekstruderingsfasen er implementert som en ulineært program (NLP), ved hjelp av *direkte kollokasjon*. Denne implmenteringen er gjort mulig ved hjelp av en progressortransformasjon av modellen fra tid til ektruderingslengde. Denne transformasjonen er vist at yter flere fordeler, som å gjøre implementasjon av tidsvarierende og diskontinuerlige modelligninger enkelt og effektivt, samt at det vil være en betydelig reduksjon av optimaliseringsvariabler i det resulterende ulineære programmet.

Etter å ha løst optimaliseringsproblemet av ekstruderingsfasen for flere konfigurasjoner, er det funnet at ved å optimalisere den initielle temperaturprofilen til billeten ved starten av ekstruderingsfasen, kan man redusere ektrsuderingstiden med opp til 50% fra en ikke-optimalisert prosess. I tillegg er det funnet at ved å optimalisere initiell temperaturprofil samt ramhastighet og kjøleprofil, versus å kun optimalisere ramhastihet og kjølingsprofil, kan ekstruderingtiden reduseres med 26%. Det er også funnet at optimalisering av både kjøling og ramhastighet, versus bare optimalisering av ramhastighet, kan redusere ektruderingstiden med 23%. Resultatene er ikke validert med experimentell data.

Table of Contents

List of Figures vi

List of Tables ix

1 Introduction 1

1.1 Motivation . . . 2

1.2 Thesis Objective . . . 3

1.3 Literature Review . . . 3

1.4 Contribution . . . 5

1.5 Thesis Outline . . . 6

1.6 Notation . . . 7

2 Theory 8

2.1 Metal Extrusion . . . 8

2.1.1 Extrusion Basics . . . 8

2.1.2 Types of Extrusion . . . 10

2.2 Differential equations and Difference Schemes . . . 12

2.2.1 Partial Differential Equations . . . 12

2.2.2 Finite Difference Schemes . . . 13

2.3 Nonlinear Programming . . . 14

2.3.1 Transcription Method . . . 15

2.3.2 Direct Collocation . . . 16

3 Extrusion Process 19

3.1 Heating phase . . . 20

3.2 Transit Phase . . . 21

3.3 Extrusion Phase . . . 21

3.3.1 Temperatures . . . 22

4 Extrusion Model 24

4.1 Continuous Model . . . 26

4.1.1 Viscous Dissipation . . . 30

4.1.2 Viscous Dissipation for Peak Section . . . 31

4.1.3 Reduction Work . . . 31

4.1.4 Reduction Work for Peak Section . . . 32

4.1.5 Average Section Speeds . . . 32

4.2 Model Discretization . . . 33
4.2.1 Discretized Dynamic Equations . . . 36
4.2.2 Discretization Under 'Diminishing Billet' Conditions . . . 37
4.2.3 Coolant Model . . . 38

5 Optimization **39**
5.1 Optimal Control Problem . . . 39
5.1.1 Separation of Problem Formulation . . . 40
5.2 Nonlinear Programming . . . 42
5.2.1 Progressor Transformation of the Extrusion Model . . . 43
5.2.2 Various NLP Considerations . . . 45
5.2.3 Final NLP Formulation . . . 53
5.3 Analysis on Differentiability . . . 54
5.4 Implementation . . . 55
5.4.1 MATLAB Implementation . . . 56
5.4.2 NLP Horizon . . . 59
5.4.3 Notable IPOPT options . . . 59
5.4.4 Scaling of variables . . . 60
5.4.5 Sparsity Patterns . . . 60

6 Results **63**
6.1 Model Tuning and Validation . . . 63
6.2 Simulation Results . . . 63
6.3 Optimal Ram Speed . . . 66
6.4 Optimization with Coolant . . . 68
6.5 Optimizing Initial Taper . . . 70

7 Discussion **76**
7.1 Advantages of the Proposed Method . . . 76
7.1.1 Advantages of the Extrusion Cycle OCP Separation . . . 76
7.1.2 Problem Reduction by Progressor Transformation . . . 78
7.2 Applicability in a Real Industrial Setting . . . 79
7.3 Model Discrepancies . . . 80
7.4 Transit IVP and Feasible Region of the Initial Taper . . . 81
7.5 Initial Guess . . . 81
7.6 Regularization . . . 82

7.7 Further Work 83

8 Conclusion **84**

Bibliography **85**

Appendix **87**

A Listings **87**

List of Figures

1 Illustrations of a heat exchanger and the extrusion profile used as a case study in the ExtruTeC project. These illustrations are sketches, and are not true depictions of the heat exchangers and extrusion profile in question. 1

2 An illustration showing how the webs of an MPE tube may rupture under pressure. This type of failure is referred to as “web tearing”. 2

3 The intended control hierarchy for the extrusion process, consisting of an MPC solution to control the extrusion phase (right blue), a controller for the heating phase (left blue), and an overall extrusion cycle optimization layer (yellow). 3

4 A sketch of the four main components of an extrusion press; the billet, container, die, and ram, in a ’direct’ extrusion configuration. 8

5 A rudimentary sketch of an extrusion press, illustrating its main components and the peak temperature of the aluminium. 10

6 Illustration of *Direct* and *Indirect* extrusion. Fig. 6a illustrates direct extrusion, and Fig. 6b illustrates indirect extrusion. Parts that are fixed in space are shown in red, and moving parts are shown in green, where the green arrows indicate the direction of movement. The aluminium is shown in blue. 10

7 The typical shape of the pressure curves for direct and indirect extrusion, shown along the extrusion length. The curves are typically divided into three zones; I) Rapid Rising, II) Steady State and III) Sharp Rise. The figure is a recreation of [Fig. 2] by Saha (2000). 11

8 Illustration of the extrusion cycle of a billet. This is the cycle subject to optimization by the ECOL. 19

9 Phases: -Heater, -Transit, -Extrusion, -Reload Time 19

10 Illustration of the induction heater, showing an aluminium billet and four surrounding induction coils, each with their own controlled AC voltage. By controlling the power of each coil individually, one may produce a temperature taper in the billet. 20

11 A cut view of the rudimentary recreation of the die assembly considered in this thesis at the end of a container (yellow), consisting of 6 dies (blue), a die holder (green), and a die ring (orange). Fig. (b) provides a closer look at the crudely reconstructed die. 21

12 Depiction of the various model sections, as defined by the model. Steel sections are shown in brown, and aluminium sections are shown in gray. The red arrow indicates the direction of flow, or ‘extrusion direction’. The vertical dashed gray line separates the sections that are common for the whole system, and the sections that represent a specific cavity. The yellow and red dashed lines represent the main and secondary center axes respectively. 26

13 The cylindrical coordinate frame used to describe points within the billet and various other components of the extrusion press. 27

14 An illustration of the way the billet is divided into control volumes, referred to as ‘cells’. The image shows a cut view of a subset of the billet cells, whereas the true discretizations fill the full volume of the different sections. 34

15 Depiction of discretization of sections into ‘cells’. The coordinates represent the numbering of the cells, where n_r and n_x represent the number of cells in the radial and axial dimensions respectively in the relevant section section. 34

16 Illustration of three ways to adjust the spatial discretization of the billet upon it diminishing during extrusion. The figure shows four instances of the billet and its discretization, the top being the initial/non-shifted version, and the three bottom ones being the three options after a shift of ΔL. 37

17 Illustration of the backmost cells disappearing, or 'dying', during the extrusion process. 38

18 Phases: -Heater, -Transit, -Extrusion, -Reload Time. Illustration of the extrusion cycles with small extrusion times. The figure showcases how the heating time must be very large in comparison to the extrusion time before it affects the extrusion period. 41

19 Illustrating the separation of the OCP into two parts, connected via the transit phase. The second part (B) encompasses the extrusion phase and is solved first, and the first part (A) encompasses the heating phase and is solved second, based on the solution of OCP B and the dynamics of the transit phase (the IVP). 41

20 A conceptual sketch of the relationship between progress in time and progress in extrusion length, as given by the ram position curve (shown in orange). The figure illustrates how the times at which the discontinuities of the model occur are dependent on the ram velocity curve, and therefore unknown prior to solving for the optimal ram speed. 44

21 An illustration of a discretization grid in extrusion length, that aligns with all model discontinuities of the model. These discontinuities occur at the axial billet cell intersections, where the green arrows point. The blue arrows indicate additional discretization points. The red arrow indicates the direction of extrusion. 45

22 The figure depicts a piecewise constant ram speed curve over both extrusion length. L, and time, t. The piecewise constant speeds are of uniform length in extrusion length, both not in time, making a time-center-to-center time derivative approximation (violet) nonlinear in ram speed. 46

23 An illustration of how the piecewise constant ram speed reference will be tracked in practice by the hydraulic ram system. The blue graph represents the piecewise constant ram speed reference, and the purple graph represents the hydraulic ram system tracking the reference by availing maximum acceleration near the steps in the reference, denoted $t_{ac.,k}$and $t_{ac.,k+1}$. 47

24 Flow chart depicting the process of defining cycle specific properties, and making and solving the corresponding NLP. 55

25 Structure of the BuildNLP routine of the *ExtrusionProcess*-class. The related Listings are given in square brackets. 56

26 Sparsity pattern of the constraints Jacobian of the implementation of (174) with parameters as specified in Table 17. Blue dots indicate all elements with potentially non-zero values upon evaluation. Density: 0.0058023. 61

27 The sparsity of the respective objective Jacobian to the NLP defined in Table 17. The Jacobian consists only of a single row. 62

28 The fixed initial taper used for all instances where a fixed taper is used in this thesis. The temperature of a cell is displayed inside that cell in $[^oC]$. The cell-color is a visual aid, where the color represents the relative temperature compared to the rest of the billet, and highlight the temperature distribution inside the billet. The discrete radial and axial coordinates are displayed along the sides of the billet. The figure corresponds to the structure presented in Fig. 12, such that the extrusion happens from right to left. 64

29 The simulation results of the tests defined in Table 18. Temperatures: ■-peak, ■-port, ■-feeder, ■-billet. Within a color-scheme, darker colors represent cells at higher radial coordinates. The shortest billet graphs represent the rightmost cells, that die first, and the longest graphs represent the leftmost cells, that die last. 65

30 The ram speed trajectory results of the tests defined in Table 19. 66

31 The temperature trajectory results of the tests defined in Table 19. Temperatures: ■-peak, ■-port, ■-feeder, ■-billet. 67

32 The optimal control trajectories of the NLPs defined by Table 20. 68

33 The temperature trajectory results of the tests defined in Table 20. RC → Ram Speed and Coolant NLP. Temperatures: ■-peak, ■-port, ■-feeder, ■-billet. 69

34 The valve opening trajectory results of the tests defined in Table 21. FF → Fine Full NLP. 71

35 The temperature trajectory results of the tests defined in Table 21. FF → Fine Full NLP. Temperatures: ■-peak, ■-port, ■-feeder, ■-billet. 71

36 The valve opening trajectory results of the tests defined in Table 22. CF → Coarse Full NLP. 72

37 The temperature trajectory results of the tests defined in Table 22. CF → Coarse Full NLP. Temperatures: ■-peak, ■-port, ■-feeder, ■-billet. 73

38 The optimal initial tapers with respect to the NLPs defined in Table 21. FF → Fine Full NLP. The color scheme is fixed with respect to the temperature values in all results of the set of Full NLPs. 74

39 The optimal initial tapers with respect to the NLPs defined in Table 22. CF → Coarse Full NLP. The color scheme is fixed with respect to the temperature values in all results of the set of Full NLPs. 75

40 The graphs represent the reduction factor arising from the separation of the extrusion cycle OCP, under the assumptions given in Section 7.1, for various values of $n_{w,h}$, $n_{w,t}$, and $n_{w,e}$. For simplicity of notation, (b) uses $n_{w,h} = 100n_h$, and $n_{w,t} = 100n_t$. 77

41 The figure illustrates how the proper allocation of variables for the billet cells throughout the NLP horizon contributes to reducing the necessary number of NLP variables. The dark cells indicate 'active' cells, and light cells indicate 'dead' cells. The red line illustrates the how the number of active cells are nearly half the total number of cells. 78

List of Tables

1 Time related parameters that describe the extrusion cycle. 20

2 Constraints of the PID controlled hydraulic ram speed, and the coolant flow valve opening. 22

3 The assumed initial temperatures of the extrusion phase in various "sections" of the extrusion press. The sections are defined in Section 4. 23

4 Constraints on temperatures during extrusion, where T is used to denote any temperature, T_{peak} denotes the temperature at the exit of the die, and $\delta t \in \langle 0, t_f \rangle$ denotes some time into the extrusion phase. 23

5 Model parameters. 25

6 Model section descriptions. 26

7 Billet section boundary conditions. 28

8 Container section boundary conditions. 28

9 Feeder section boundary conditions. 28

10 Port section boundary conditions. 29

11 Peak section boundary conditions. 29

12 Plate section boundary conditions. 29

13 Die section boundary conditions. 29

14 Resolution of section discretizations. ND reads "Not Determined", and means that the parameter is tuned/chosen for each dynamic optimization problem (DOP). . . . 35

15 IPOPT options. 59

16 Variable Magnitudes. 60

17 Particular configuration for the NLP of which the constraint and objective Jacobians are inspected in Fig. 26 and Fig. 27. 60

18 Simulation results, where Billet, Container, and Die are the number of cells in the respective sections, d is the number of collocation points, n_L is the number of discretization points per axial billet cell, thus $N_L = n_{x,billet} \cdot n_L$ is the number of discretization points along the progressor, n_w is the total number of decision variables, n_g is the total number of constraints, n_{eq} is the number of equality constraints, n_{in} is the number of inequality constraints, v_{ram} is given in m/s, t_f is the extrusion time in seconds, t_{sol} is the solve time in seconds, and N_{iter} is the number of iterations before convergence. Entries in black are defining properties, entries in green (■) are recessive properties, and entries in red (■) are the results. 64

19 Results of the Ram Speed Optimization NLPs. Symbols and colors have the same meaning as in Table 18. 66

20 Results of the Ram Speed Optimization NLPs. Symbols and colors have the same meaning as in Table 18. 68

21 Results of the fine full NLPs, (174), for various parameters. FF $\rightarrow$ Fine Full NLP. Symbols and colors have the same meaning as in Table 18. 70

22 Results of the coarse full NLPs, (174), for various parameters. CF $\rightarrow$ Coarse Full NLP. Symbols and colors have the same meaning as in Table 18. 73

Listings

1 Imposing initial conditions, where $\Delta T_{init}^{-} \leq h_{billet} \leq \Delta T_{init}^{+}$ 57
2 Defining billet dynamics. 57
3 Impose collocation and continuation constraints. 58
4 Computing cost. 59
5 Code that instantiates an *ExtrusionProcess()*, builds the corresponding NLP, and solves it. 87
6 Defining SX variables. 87
7 Defining initial optimization variables. 87
8 Defining collocation variables. 87
9 Defining algebraic and parameter equations. 88
10 Solving for parameters. 88
11 Defining next state variables. 88
12 Imposing soft constraints. 88
13 Solve for algebraic states at collocation points. 88
14 Defining next optimization variables. 88
15 Impose constraints on steps in ram speed. 88
16 Solve for the algebraic states at the discretization points. 89
17 Constructing *Input_Cell*{}. 89

Nomenclature

Extrusion Process

Billet	Cylinder of the material that is meant to be extruded
Cell	'Cell' is the name given to the part of a control volume that is occupied by a material that is modelled. Most cells are equal to the control volume, except from the the back-most cells of the billet are sub-volumes of the control volumes as the billet is being translated towards the die.
Container	The steel housing that encases the billet. A part of the extrusion press.
Die	A steel component of the extrusion press, through which the billet is being forced by the ram, imposing a geometric cross section of the extrudate. It is also the name of a group of control volumes/cells within the physical die.
Discard (or Butt)	The backmost part of the billet, which is not extruded, but rather sheared off before a new billet is inserted
Exit	The name of a group of control volumes/cells at the end of the aluminium flow, where it has exited the die.
Extrudate	Material that has been extruded through a die.
Extrusion cycle	The periodic chain of events, following a billet from the start of its heating phase (phase 1) to the end of its extrusion phase (phase 3).
Extrusion length	The length that the ram has travelled from its starting position, which is equal to the length of billet that has been extruded at any given time during the extrusion phase.
Extrusion phase	The third and last phase (phase 3) of the extrusion cycle, where a billet is being extruded.
Extrusion press	A machine that houses a billet and pushes it with a hydraulic piston (ram) through a die, to produce the exiting extrudate.
Extrusion pressure	The pressure exerted by the ram on the billet during extrusion.
Extrusion radius	The distance from the main center axis to the secondary center axes, where the main axis is the center axis of the billet, and the secondary axes are those of each cavity.
Extrusion ratio	The ratio between the speed of the ram, and the outgoing extrudate in an extrusion process.
ExtruTeC	"High Precision Extrusion Temperature Control through Digital Technology", The research project of which the work in this thesis is a part of.
Feeder	A section of the extrusion press where the aluminium flow is guided into the port. It is also the name given to the control volumes/cells that represents the aluminium in the feeder region.
Heating phase	The first phase (phase 1) of the extrusion cycle, where a billet is preheated in an induction heater before being transported (phase 2) to the extrusion phase (phase 3).
Initial taper	The billet taper at the start of the extrusion phase.

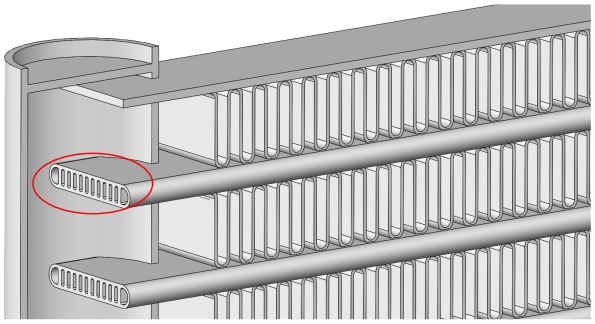
Isothermal extrusion	Extrusion where the critical temperature og the extrudate at the exit of the die is maintained at a constant value, to ensure consistent product quality.
MPE	Multi Port Extrusion, a type of extruded aluminium tubing used in heat exchangers, and the main extrusion profile in ExtruTeC's case study.
Peak	A section and control volume/cell of the die where aluminium flows and is assumed to have the highest temperature during the extrusion phase.
Plate	A steel component of the extrusion press, that has channels for the flow of liquid nitrogen.
Port	A section of the die where the aluminium flow is guided to into the "peak" region of the die.
Ram	A hydraulic piston, responsible for forcing the billet through the die.
Saturation time	A period of time during the heating phase, typically towards the end of the phase, where the induction coils are idle, to allow the radial temperature gradients in the billet to even out.
Taper	The temperature distribution within a billet, typically 'tapering' off from the extrusion end to the other.
Transit phase	The second phase (phase 2) of the extrusion cycle, where the billet is being transported from the induction heater to the extrusion press.
Miscellaneous	
IRK	Implicit Runge-Kutta scheme; an implicit numerical integration scheme.
IVP	Initial Value Problem; the problem of finding the trajectory of a dynamic system, that contains a given value (the 'initial value').
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
Optimization	
(N)MPC	(Nonlinear) Model Predictive Control; a model-based controller that predicts the state and input trajectories a given distance into the future, called the "horizon".
Collocation point	A point representing a given value of the progressor of a dynamical system between two of its discretization points, at which a collocation polynomial is designed to match the system dynamics.
Decision variable	A variable in the set of variables that an optimization algorithm searches through to find the optimal value of the respective objective function
Direct collocation	A simultaneous dynamic optimization technique that uses collocation polynomials in the constraints of an NLP to integrate the system dynamics.
Discretization interval	The interval of the progressor of a dynamical system between two discretization points. The discretization interval is considered to correspond to the one of the discretization points that define the interval that appears 'earlier' in the progression of the system.
Discretization point	A point representing a given value of the progressor of a dynamical system, at which the system state is described as part of discretization of the system trajectory along that progressor.
NLP	Nonlinear Program; the name given to the formulation of an optimization problem with either a nonlinear objective function or constraints.

1 Introduction

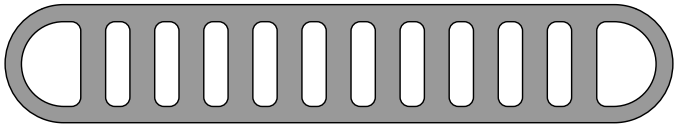
The use of aluminium is vast and important, and in particular, the automotive industry relies on aluminium to construct light weight vehicles. The automotive industry continuously seeks to increase vehicle efficiency, thus demanding lighter and lighter components for their vehicles. This puts pressure on the aluminium extrusion industry, since extruded aluminium profiles are frequently used for various components, such as heat exchangers, in vehicles. The aluminium profiles need to be thinner and lighter while maintaining their strength and quality, to decrease weight without sacrificing functionality nor safety margins. The extrusion industry is already achieving incredibly small thicknesses, and further reduction requires sophisticated and precise temperature control to ensure high quality extrudate. With the introduction of a precise control system for temperature control and advanced optimization techniques, the industry is also setting out to push the limits of productivity.

The work in this thesis is part of the *High Precision Extrusion Temperature Control through Digital Technology (ExtruTeC)* research project, whose aim is to develop precise temperature control for the extrusion process, to obtain isothermal extrusion, and to improve production rate by means of optimization techniques. The project is a collaboration between Hydro Extruded Solutions - Precision Tubing, Cybernetica AS, SINTEF Industry and Norwegian University of Science and Technology (NTNU). A part of the project is the development of accurate models of the extrusion process, upon which a hierarchical Model Predictive Control (MPC) control scheme will be based. Computationally heavy and comprehensive simulations are run to gain understanding of the intricacies of the process, and dynamic estimation techniques are pinpointing model parameters. In the center of the project is the case study of an extrusion press in Tønder owned by Hydro Extruded Solutions.

Specifically, the extrusion profiles that are focused on in the ExtruTeC project are that of Multi Port Extrusions (MPE) for heat exchangers, as illustrated in Fig. 1. A normal view of the MPE profile is shown in Fig. 1b, where one can see a stadium shape with internal walls referred to as “webs”. These webs are the main subjects of thickness reduction while developing the precise temperature control technology.



(a) A ‘cut-view’ illustration of a heat exchanger for vehicles. Circled in red is the MPE profile used as a case study in the ExtruTec project.



(b) The MPE profile, appearing in heat exchangers for vehicles, as seen in Fig. 1a, which is the main extrusion profile of ExtruTeC’s case study. The internal walls are referred to as “webs” and are the subjects of thickness reduction in the ExtruTeC project.

Figure 1: Illustrations of a heat exchanger and the extrusion profile used as a case study in the ExtruTeC project. These illustrations are sketches, and are not true depictions of the heat exchangers and extrusion profile in question.

1.1 Motivation

During operation, an MPE tube is pressurized, thus the strength of the inner walls are paramount. If, for some reason, the web strength of a tube is not according to the requirements, there is a risk of "web tearing". Web tearing is when the pressure inside the tube is high enough to rupture the webs. Figure 2 shows an illustration of an MPE profile that has experienced web tearing, causing the profile to bulge. This type of failure is critical and unacceptable during operation, and must therefore be avoided. There are two main mechanisms that cause weakening of the webs that result in web tearing, and are both related to the extrusion phase;

i) Local melting of the webs during extrusion of the MPE profiles, due to local temperature peaks rising above the melting point of aluminium,

ii) and flow imbalance of the inner webs and outer walls, as the extrudate exits the die.

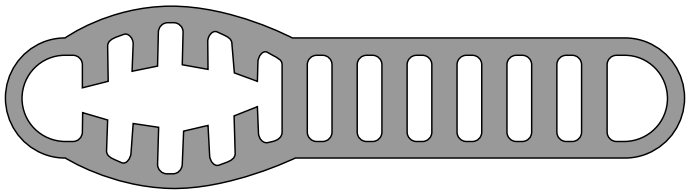


Figure 2: An illustration showing how the webs of an MPE tube may rupture under pressure. This type of failure is referred to as "web tearing".

To combat the mechanisms that weaken the webs during extrusion, precise temperature control is required. Both melting and flow rate are directly related to the temperature of the aluminium, and it is clear that gaining understanding of the complex temperature picture of the aluminium undergoing complex geometrical changes when extruded can help develop methods of combating these weakening mechanisms. Of course, upon gaining deep understanding of the extrusion process and developing sophisticated models to describe their behavior, the door opens to the world of optimization. Because of the growing use of aluminium, and in particular, extruded aluminium profiles, the optimization of production is desired to facilitate the current and potential future increase in demand. Resultingly, the main angle of this thesis, is on the optimization of production, by minimizing the extrusion cycle period.

In the bigger picture, an overarching motivation behind the thesis, in cooperation with Cybernetica AS, is to gain more knowledge about- and understanding of how one can use nonlinear optimization approaches to optimize complex industrial processes. Industrial processes are typically highly complicated and hard to model, often resulting in greatly simplified or highly nonlinear and even non-smooth or discontinuous models. Optimization of such systems is therefore nontrivial, and the various typical approaches has advantages and disadvantages. Nonlinear programs (NLPs) are strong contenders when choosing an optimization approach, since they lend themselves nicely to being solved numerically by computers. NLPs are especially useful due to the immense increase in computing power we have seen the last few decades, allowing computers to efficiently handle increasingly large NLPs. Sequential methods is a category of simple ways to implement an optimal control problem (OCP) as an NLP, though these methods, as most methods, have drawbacks in addition to their strengths. It is therefore useful to have experience with another category of NLPs, such as simultaneous methods, that can yield advantages outside what one can expect from sequential methods. Therefore, the motivation behind the work presented in this thesis is partially to gain understanding of how, and in what circumstances, simultaneous methods are applicable to complex industrial processes, and how to implement them in practice. In particular, the focus is on *Direct Collocation*, a powerful type of simultaneous method.

1.2 Thesis Objective

The thesis objective is to conceptualize a practical solution to the extrusion cycle optimization problem, and perform a partial implementation as proof of concept. The thesis covers the use of optimization techniques, in particular; *Direct Collocation* to optimize the overall extrusion cycle with respect to minimum cycle period, thus increasing production rate. The proposed control hierarchy is shown in Fig. 3, where the efforts made in this thesis are to develop the optimization layer for the extrusion cycle. In short terms, the extrusion process can be divided into a heating phase, a transit phase and an extrusion phase. Other parties within the ExtruTeC research project are developing an MPC layer for the extrusion phase, to ensure precise and accurate temperatures, and fast extrusion. Gabrielsen (2022) investigated the use of optimization techniques to control the heating phase through simulations with promising results. The extrusion cycle optimization layer (ECOL) computes an open-loop solution of the extrusion phase, optimizing over the initial temperature distribution (initial taper) of the billet as it enters the extrusion phase. The ECOL also provides a reference taper and open loop trajectories for the heating phase, and accounts for the temperature diffusion happening during the transit phase.

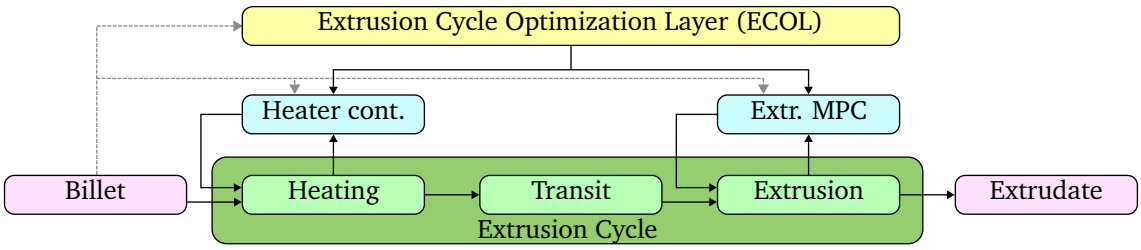


Figure 3: The intended control hierarchy for the extrusion process, consisting of an MPC solution to control the extrusion phase (right blue), a controller for the heating phase (left blue), and an overall extrusion cycle optimization layer (yellow).

The extrusion cycle optimization layer consists of a dynamic optimization problem in the form of a nonlinear program (NLP) that optimizes the open-loop trajectories of the extrusion phase with respect to extrusion time, where the control variables are the initial taper, a set point for the ram speed, and a set point for a coolant flow valve. It must also simulate the taper backwards through the transit phase, as an initial value problem (IVP), to provide a reference taper for the heating phase. By accounting for the heater control problem and transit dynamics, constraints can be imposed on the initial taper to ensure feasibility with respect to the heating phase. The open loop trajectory solutions of the extrusion phase found by the ECOL are then made available for reference to the extrusion phase MPC layer, opening up possibilities for the MPC control structure, such as reference tracking. Whenever a new billet will start its cycle, the extrusion cycle optimization layer is provided with information about the billet, such as its alloy composition and properties thereof, its geometry, its temperature, the ambient temperature, etc., and computes a unique solution for the individual billet. In this thesis, we do not consider varying billet properties between cycles, such that consecutive cycles are assumed identical and one can draw conclusions on the overall productivity.

1.3 Literature Review

Several works have studied modeling, control and subsequently optimization of the extrusion process, mainly in order to achieve better product quality through isothermal extrusion, and higher productivity through optimizing the ram velocity profile. This section provides a review of various publications that present different approaches to modelling, control and optimization the extrusion process.

Tibbetts and Wen (1995) expands on a new modelling technique described by Han et al. (1986), and decomposes the extrusion process into a hierarchical architecture to improve the tractability of

the control problem. They formulate two extrusion process control problems to optimize the ram position profile; one for obtaining isothermal extrusion, and one for minimizing extrusion time. The respective objectives they present are

$$\min_{L(t)} \int_0^{t_f} \left(T(L(t),t) - T_d\right)^2 dt \tag{1}$$

$$\min_{L(t)} \int_0^{t_f} \frac{1}{\dot{L}(t)} dL \tag{2}$$

with any relevant process constraints, where

L : ram position as function of time
t_f : final time of extrusion phase
T : exit/peak/extrusion temperature
T_d : desired exit temperature.

They continue to show that optimizing the extrusion process can yield significantly shorter extrusion times.

Cuéllar Matamoros (1999) presents a *"semi-analytical"* approach to the modelling of isothermal extrusion processes, where by assuming that the aluminium behaves like a highly viscous fluid. They write;

> "By treating the metal as a highly viscous fluid and making appropriate assumptions regarding its flow, the velocities, pressure, strain, and strain rates distributions needed for the modeling of the process are solved analytically leaving therefore only the temperature distributions to be solved numerically. This semi-analytical approach allows for a considerable reduction in computation times as compared to the usual Finite Elements Method for the modeling of extrusion processes."

The approach taken to model the extrusion process presented in this thesis is based on work by Halås (unpublished) from an internal memo, and is reminiscent of the idea presented by Cuéllar Matamoros (1999). Matamoros proceeds to formulate and solve open-loop optimization problems that optimize the ram speed both as a constant speed and as a varying speed over time, to minimize to extrusion time. It is however not clear form the their PhD thesis exactly how they implemented the optimization problem, nor what approach was used. Furthermore, they expand the optimization framework to include an NMPC control scheme for online feedback control. They also recognize the importance of the initial billet temperature profile, the 'initial taper', although did not implement such an optimization scheme.

Chanda et al. (2001) also studies the effect of ram speed on isothermal extrusion, but does so using 3D finite element method (FEM) simulations. Their focus is on constant ram speed and step-wise varying ram speed, and their effect on the exit/peak temperature of the aluminium, and on the extrusion pressure. The simulations are done only for early stages of the extrusion phase, due to large computation times. They find that by using an initially high ram speed, and step wise decreasing it over time, they are able to extrude at the same isothermal temperature as for constant ram speeds, but with a higher average speed, thus increasing production. In addition, they see reduced tensile stresses and effective strain rates in critical regions, which makes the extrudate less prone to tearing, and reduced pressure requirements for the press. However, the instantaneous ram speed changes are also found to cause sudden changes in pressure.

Zhou et al. (2004) expanded on the study by Chanda et al. (2001) in the presence of increased computing power. They simulate an extrusion process over the entire extrusion stroke with a gradually varying ram speed. By using a continuously decaying ram speed, they are able to maintain the temperature within a range of 10^oC over the entire extrusion stroke. The results were verified by experimental results.

Farjad Bastani et al. (2010) studies the effects of container cooling, ram speed, front temperature of billet, and initial billet temperature on isothermal extrusion. The extrusion phase is divided into

three sub-phases; start, middle and end of the ram stroke. By using 3D finite element software, they simulate over 20 press cycles, and show that container cooling allows for an increase in ram speed. In addition, they find that a flatter temperature taper of allows them to maintain isothermal extrusion more consistently through the three phases of the press stroke.

Hölker et al. (2013a) uses an additive manufacturing technique called 'selective laser melting' (SLM) to produce dies with cooling channels, and performs trial runs to investigate the effects of cooling strategies with compressed air. They show that cooling the die during extrusion can lower the extrudate temperature without significant rise in extrusion pressure, thus allowing faster extrusion with higher surface quality of the extrudate.

Finally, **Holven (2020)** develops and implements an NMPC control scheme for the extrusion phase using Cyberneticas' tools for NMPC control. The NMPC control objective is to maintain isothermal extrusion while minimizing extrusion time, by using the ram speed and a coolant valve opening for die cooling, as control variables. Similarly, **Kulås (2022)** uses Cybernetica's tools to implement open loop dynamic optimization and parameter estimation for a combined model; connecting the heating, transit and extrusion phase models. This includes the temperature set points for the heating phase and the corresponding initial taper.

As seen from the aforementioned publications, there are several works related to the study of ram speed curves, and methods to decrease the extrusion time. Usually, they hint at the fact that the initial billet temperature taper is a key factor in dictating the minimal extrusion time, although, little to no research is done on its optimization except for Kulås (2022), who implements a version of initial taper optimization. In their Master's thesis, however, they state that the *"the optimizations of both ram speed and heating coil set points struggled with instability and were difficult to tune"*. On top of this, the optimization algorithm took *"around 5 minutes"* to converge, which is rather slow if one plans on implementing such an algorithm in an on-line adaptive extrusion cycle optimization scheme, since this is in the same range as the duration of the extrusion phase for some products. Kulås (2022) attributes these issues to the use of *single shooting* as the dynamic optimization approach, and they point out that single shooting is not suited for highly nonlinear dynamical systems over long prediction horizons, which in many cases is true. Especially when using an explicit integration scheme, as was done by Kulås (2022), there are in general no guarantees that the integration is successful in providing meaningful state trajectories, which may result in failure to converge to an optimal solution. As a solution, they recommend either of two other approaches; *multiple shooting* or *orthogonal collocation (direct collocation)*. Another recommendation is to introduce the coolant flow as a control variable, to obtain even shorter extrusion times.

1.4 Contribution

This thesis presents an extrusion cycle optimization control scheme, in which the problem is separated into smaller optimization problems that represent different phases of the extrusion cycle; the heating phase, the transit phase, and the extrusion phase. Under certain assumptions, some of which are verifiable upon attaining a solution, optimality is preserved in spite of the simplification by separation. The focus of the thesis is on dynamic open loop optimization of the extrusion phase, and how to connect it to the heating phase. To attack the problem of optimizing the initial taper is in itself a recent and interesting development, and this thesis presents the novel use of direct collocation for this purpose, as suggested by Kulås (2022). In addition to optimizing the initial billet taper, the ram speed and coolant flow are optimized over the entire horizon of the extrusion phase. This is achieved reliably due to a novel approach of process discretization within the field of extrusion optimization, where the system state and control trajectories are discretized in extrusion length rather than time. The new approach allows efficient implementation of direct collocation, which in turn allows optimization over long time-horizons while maintaining accurate predictions. As a bonus, the new method of discretization greatly reduces the problem size, in terms of the number of variables in the resulting NLP.

To summarize, this thesis presents

1. a proposed control structure for an open loop extrusion cycle optimal control scheme, in

which the problem formulation is separated into smaller problems, while maintaining optimality,

2. the use of direct collocation to provide reliable optimal trajectory solutions, even over long time horizons, within a feasible time frame,
3. a novel approach to the discretization of the extrusion process state and control trajectories, that allow simple and efficient implementations of direct collocation and contributes to decreasing the problem size significantly,
4. and the use of initial taper, ram speed and coolant flow optimization combined.

1.5 Thesis Outline

The thesis starts by providing an overview of previous work that has been done on the optimization of extrusion processes in Section 1.3, and summarizes the contributions this thesis makes to the field of extrusion and extrusion cycle optimization. Continuing, a brief summary of the most fundamental concepts that underlie the techniques used and presented in this thesis is given in Section 2. The extrusion process is described in Section 3, which covers the overall process and its decomposition into three main phases; the heating phase, the transit phase, and the extrusion phase. An overview of the underlying continuous model that form the basis of the model implementation presented in this thesis, is given in Section 4.1, followed by the description of its discretization into implementable form, presented in Section 4.2.

The main contributions of the thesis are presented in Section 5. The Section starts by formulating an optimal control problem (OCP), that represents the objective of minimizing the extrusion cycle time, and upholds process constrains such as the extrusion press and heater limitations, and requires isothermal extrusion at the desired peak temperature. Section 5.1.1 separates the optimal control problem into smaller OCPs that represent the heating (OCP A) and extrusion phase (OCP B), where optimality with the respect to the original OCP is preserved under some assumptions. The conversion of OCP B into a nonlinear program (NLP) is covered in Section 5.2, where Section 5.2.1 presents the progressor transformation of the extrusion model, from time to extrusion length. Before formulating the final NLP, various considerations regarding the formulation are covered in Section 5.2.2, and the final NLP formulation is presented in Section 5.2.3. The implementation of the extrusion phase NLP into MATLAB, and properties thereof, are presented in Section 5.4.

Section 6 presents the results of- and comments on the extrusion phase NLP testing, where solutions are found for versions of the NLP that simulates the process without optimized control, optimizes only the ram speed, optimizes the ram speed and coolant valve opening, and optimizes the complete extrusion phase NLP. The discussion, in Section 7, covers

- model discrepancies between the model as implemented for the work in this thesis and the model upon which it is based (Halås unpublished),
- the applicability of the proposed extrusion cycle optimization structure and the implementation presented in this thesis, with respect to a real industrial setting,
- various advantages of the proposed method,
- a more detailed discussion on the topic of understanding the feasible region of the initial taper and on the initial guess,
- and recommendations for further work on the method.

Lastly, the conclusion is presented in Section 8.

1.6 Notation

This thesis makes use of the following notation:

- $\mathbb{Z}_{a,b}$ - Short-hand notation for the set $\{x \in \mathbb{Z} | a \leq x \leq b\}$.
- $\mathcal{F}$ - Used to denote a feasible set.
- **Bold Variables** denote array valued variables.
- $\boldsymbol{f} : \mathbb{R}^n \times [\cdot] \mapsto \mathbb{R}^n$ - Used to denote a mathematical model on the form: $\dot{\boldsymbol{q}} = \boldsymbol{f}(\boldsymbol{q}; \cdot)$.
- T - Used for temperature related variables.
- t - Used to denote various time related variables.
- L - Used to denote lengths. When a subscript is used, it denotes the length of the object denoted by the subscript, otherwise it denotes the extrusion length as a progressor for the extrusion press model.
- μ - Used to denote dynamic viscosity when relevant to the viscous dissipation terms of the extrusion press model. When the subscript "red" is used, it denoted a reduction factor.

2 Theory

2.1 Metal Extrusion

2.1.1 Extrusion Basics

Metal extrusion is the process of forcing metal through an opening, producing a prismatic extrudate on the other side. Saha (2000) defines extrusion as

> "Extrusion is a plastic deformation process in which a block of metal (billet) is forced to flow by compression through the die opening of a smaller cross-sectional area than that of the original billet",

and Terry Sheppard (2013) writes

> "The [extrusion]process converts a cast billet of solid metal into a continuous length of generally uniform cross-section by forcing it to flow through a die which is shaped to produce the required form of product."

Compared to methods such as *rolling* and *forging*, extrusion is a modern method of shaping metals. The first credited use of extrusion as a method of forming metals was when Joseph Bramah described his method of extruding molten lead, in his patent which was granted in 1797 (Terry Sheppard 2013). The manufacturing of lead pipes then became viable in 1820, when Thomas Burr invented the hydraulic extrusion press. However, it was in 1886, following the new-found method of extracting aluminium from bauxite using electrolysis, that the extrusion method took off. The 200 year old method is to this day unrivaled in its ability to produce complex and thin-walled aluminium profiles and at the rate at which it does so (Terry Sheppard 2013).

The basic principle of extrusion is that a metal cylinder, called a 'billet', is placed into a container and pressed through an opening in a component called the 'die', which is illustrated in Fig. 4. The four quintessential parts of aluminium extrusion can be summarized as;

- the *Billet*; an aluminium cylinder that is the source material for the aluminium profiles,
- the *Container*; a hollow, steel cylinder that encases the billet during extrusion,
- the *Die*; a steel component at the end of the container, through which the billet is forced, shaping the aluminium into its desired shape,
- and the *Ram*; a hydraulic piston that pushes on the back end of the billet, forcing it through the die.

The *billet*, shown as the light grey section of Fig. 4, is the source metal for the extrudate. It can be of varying alloy composition, have varying geometry, though is usually a solid right cylinder, and is usually heated to specific temperatures. All these factors have to be considered for each individual extrusion process to obtain various desired process and product properties. Even the grain and compositional structure of the billet can affect the structure and properties of the outgoing product, making homogenization of the billets a field of study, see Yucel Birol (2014), Yücel Birol (2004) and Murai et al. (2003). During extrusion, the billet behaves like a fluid, and has intricate flow patterns that are hard to analyse and account for.

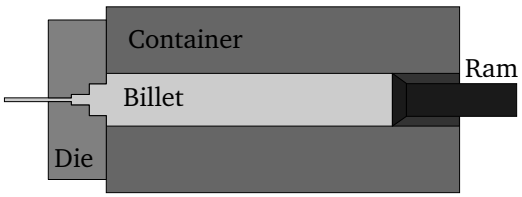


Figure 4: A sketch of the four main components of an extrusion press; the billet, container, die, and ram, in a 'direct' extrusion configuration.

Several studies have been carried out with the aim of gaining understanding of the material flow during extrusion, such as Valberg (1992) and Mahmoodkhani et al. (2014), which often focus on the flow through the die. The material flow can be complicated to understand even all the way back in the billet, before the metal reaches the complex die geometry. For instance, Kim and Ikeda (2000) have explored the flow of material in the billet during extrusion, where they find a clear connection between the friction between the billet and container, and the billet surface layer flow.

The *container* is a rigid component that encapsulates the billet during extrusion. The container is responsible for preventing the billet from expanding outwards, thus having to withstand the high extrusion pressures, which can range from tens to several hundred megapascals. The container is therefore typically made of a steel alloy, for example "5%Cr hot-work steels", as described in Terry Sheppard (2013). The container affects the extrusion process in two main ways; through heat conduction and friction between the billet and the container. The container can be heated to closely match the temperature of the billet, to reduce the heat flow between them.

The *die* is the component through which the billet is forced, giving shape to the outgoing aluminium profile. When applying force to the billet, it deforms as it passes through the die, and takes its intended shape. This part of the extrusion process is very complex, due to the complex die geometry. The dies are designed individually for each new profile, with varying complexity. Nevertheless, even the simplest dies create intricate aluminium flow patterns. Two important fields of study are;

- the optimal die geometry with respect to the required extrusion pressure, product quality and die life,
- and the understanding and modelling of the temperature generation due to the deformation and friction work in the die.

Examples of such studies, in addition to the aforementioned articles, are Qamar (2010), Li et al. (2003), and Hölker et al. (2013b), where in the latter, they investigate the effect of cooling channels in the die. The heat generation in the die is heavily influenced by the extrusion speed, extrusion pressure, die geometry and die cooling. It is within the die that the temperature reaches its highest value, making the understanding, design and modelling of the die paramount within the topic of temperature control for aluminium extrusion.

The *ram* is the hydraulically actuated piston that applies force to the back of the billet, forcing it through the die. The ram force is typically controlled by a PID controller, such that the ram speed tracks a reference speed, which is usually constant or a predetermined velocity profile. The ram is responsible for providing the extrusion pressure, and the maximum pressure capacity of the hydraulic system limits the possible extrusion pressure in the extrusion press. This is an important consideration when designing an extrusion process, as the required extrusion pressure is highly dependant on the billet alloy, the billet-container friction, deformation work and cooling effect in the die, and the extrusion ratio, which is the ratio between the speed of the ram and the speed of the outgoing product.

In Fig. 5 you can see a summarizing sketch of the extrusion press, and its main components, similarly to Fig. 4. The figure also illustrates the typical axial symmetry that is common in the extrusion industry, and coarsely depicts some of the important and complex die design parameters. That is, parameters such as feeder and port dimensions, and bearing length, which are shown as the various incremental reductions in cross-section towards the extrusion exit. The peak temperature, as the rest of the temperature in the die, is hard to estimate in terms of both location and temperature, but is generally assumed to be located at the smallest cross-section, which is right at the extrusion exit. The figure also shows how several 'cavities', or holes, are included in a die to produce several profiles at once.

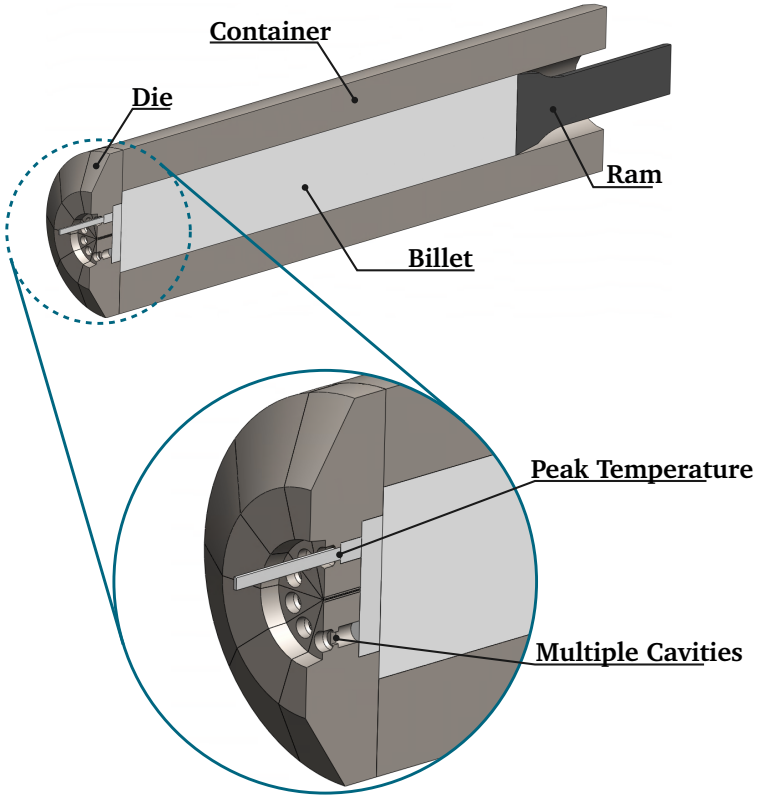
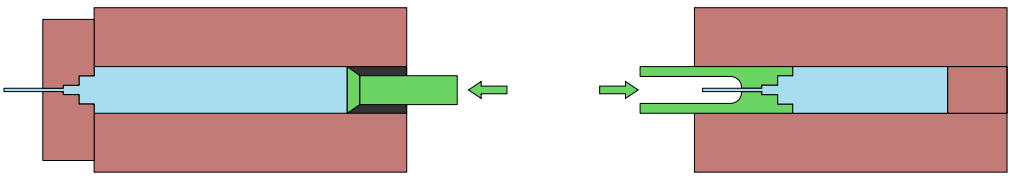


Figure 5: A rudimentary sketch of an extrusion press, illustrating its main components and the peak temperature of the aluminium.

2.1.2 Types of Extrusion

Direct and Indirect Extrusion

Extrusion can be divided into two categories; *Direct* and *Indirect* extrusion. In direct extrusion, the container and die are fixed in space, and a billet is placed into the container-die and compressed by a hydraulic ram that extends into the container. This is the most common form of aluminium extrusion. Indirect extrusion aims at reducing the billet-container friction, to drastically reduce the extrusion pressure, by fixing the container and ram, and have the die be extended into the container instead. However, according to Terry Sheppard (2013), it is generally more convenient when using indirect extrusion to fix the die in space, and move the container-ram instead. These to concepts are illustrated in Fig. 6, where the fixed entities are shown in red, and the moving entities are shown in green.



(a) Direct Extrusion. (b) Indirect Extrusion.

Figure 6: Illustration of *Direct* and *Indirect* extrusion. Fig. 6a illustrates direct extrusion, and Fig. 6b illustrates indirect extrusion. Parts that are fixed in space are shown in red, and moving parts are shown in green, where the green arrows indicate the direction of movement. The aluminium is shown in blue.

The most prominent difference between the two methods is the container-billet friction. Since

the ram must exert additional force to overcome the friction, the extrusion pressure increases significantly. Fig. 7 shows the typical pressure curves for direct and indirect extrusion, over the extrusion length, that is, the ram/die progress/extension into the container. The pressure curves are divided into three distinct zones Saha (2000);

I) a zone of rapid rise in pressure at the start of the extrusion process,

II) a steady state zone, where the pressure steadily descends or is maintained,

III) and a zone where the pressure sharply increases as the “discard” is compressed.

As is clear from the figure, the pressure steadily declines during zone II in direct extrusion, this reflects the decrease in friction force as there is less and less aluminium remaining in the container. For the indirect method, the pressure is constant during zone II, which reflects the lack of friction force overall. Note that these curves are typically seen for constant ram speeds, and without die cooling, as the pressure is heavily influenced by the extrusion speed and aluminium temperature. Also note that Terry Sheppard (2013) illustrates the pressure curves with less pronounced features in their [Fig. 1.5].

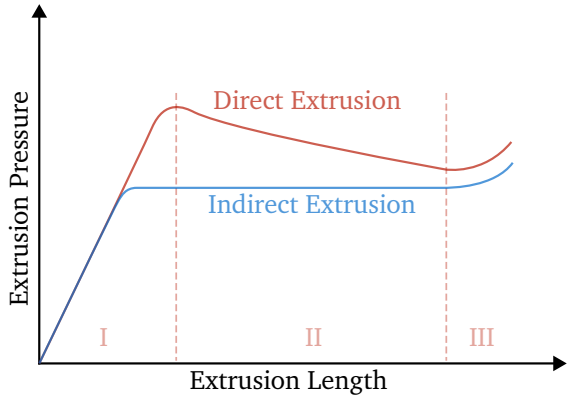


Figure 7: The typical shape of the pressure curves for direct and indirect extrusion, shown along the extrusion length. The curves are typically divided into three zones; I) Rapid Rising, II) Steady State and III) Sharp Rise. The figure is a recreation of [Fig. 2] by Saha (2000).

The lack of friction in indirect extrusion comes with several advantages, as listed by Terry Sheppard (2013);

- lower extrusion pressure,
- more even extrusion pressure over time, as the billet-container surface decreases,
- more uniform material flow due to the lack of relative motion,
- no heat generation at billet-container interface.

Nevertheless, direct extrusion is more common due to the simplicity of the extrusion press design, and that direct extrusion allows for a larger circumscribing circle of the extruded profile.

Hot and Cold Extrusion

Extrusion can be divided into hot and cold extrusion (Saha 2000), where hot extrusion is the most common and the type considered in this thesis. In hot extrusion, the billet is typically preheated in a designated induction heater before being placed into the extrusion press. This step ensures that the billet is malleable, and can be extruded with much less pressure than with cold extrusion. The exact temperature of the aluminium in a hot extrusion set-up has great impact on the quality of the

product. However, it is also possible to extrude cold aluminium. An advantage of extruding cold is that the product has different structural properties, due to the lack of effective heat treatment.

This initial heating process of hot extrusion is responsible for producing the desired temperature distribution in the billet prior to the extrusion. Particularly, the temperature distribution along the length of the billet is of importance. The distribution of temperature in the the billet is referred to as the *taper*, as it typically 'tapers off' towards the back end, and the initial distribution before the press stroke begins is referred to as the *initial taper*.

2.2 Differential equations and Difference Schemes

This section is mainly based on theory by Evans (2022), Wazwaz (2002) and Strikwerda (2004).

2.2.1 Partial Differential Equations

Partial differential equations (PDEs) is an immensely broad area of mathematics, that reaches far beyond the scope of what is needed in this thesis. Nevertheless, this section attempts to provide a brief introduction, such that the reader may form a basic understanding of what they are, before proceeding to delve into the main body of the thesis.

A partial differential equation is defined as an expression on the form (Evans 2022)

$$F\left(\frac{\partial^k u(\boldsymbol{x})}{\partial \boldsymbol{x}^k}, \frac{\partial^{k-1} u(\boldsymbol{x})}{\partial \boldsymbol{x}^{k-1}}, \ldots, \frac{\partial u(\boldsymbol{x})}{\partial \boldsymbol{x}}, \boldsymbol{x}\right) = 0, \tag{3}$$

where $k \geq 1$, $F : \mathbb{R}^{n^k} \times \mathbb{R}^{n^{k-1}} \times \cdots \times \mathbb{R}^n \times \mathbb{R} \times \mathcal{D} \mapsto \mathbb{R}$ and $\boldsymbol{x} \in \mathcal{D} \subseteq \mathbb{R}^n$ for some $n \in \mathbb{N}$. The unknown variable is then $u(\boldsymbol{x}) : \mathcal{D} \mapsto \mathbb{R}$, which is called the *dependent* variable. In contrast, $\boldsymbol{x}$ is called the *independent* variable. We call this a k^{th}-order PDE. A subset of PDEs is that of ordinary differential equations (ODEs), which have the same form as (3), but is only differential in one dimension. That is;

$$F\left(\frac{\partial^k u(y, \boldsymbol{x})}{\partial y^k}, \frac{\partial^{k-1} u(y, \boldsymbol{x})}{\partial y^{k-1}}, \ldots, \frac{\partial u(y, \boldsymbol{x})}{\partial y}, y, \boldsymbol{x}\right) = 0, \tag{4}$$

with $y \in \mathbb{R}$.

PDEs turn out to be incredibly useful and versatile, and have a wide variety of applications. They turn up in countless of real world scenarios, and are predominant in describing many real world phenomena. Wazwaz (2002) writes

> "It is well known that most of the phenomena that arise in mathematical physics and engineering fields can be described by partial differential equations (PDE). In physics for example, the heat flow and the wave propagation phenomena are well described by partial differential equations."

In fact, in this thesis we encounter both heat flow and wave propagation when describing the temperature evolution and movement of the billet during the extrusion process.

Equation (3) is said to be *linear* if it takes the form

$$\sum_{i=1}^{k} a_i(\boldsymbol{x}) \frac{\partial^i u(\boldsymbol{x})}{\partial \boldsymbol{x}^i} = f(\boldsymbol{x}), \tag{5}$$

and is referred to as *homogeneous* if $f(\boldsymbol{x}) \equiv 0$. Two quintessential PDEs are

$$\frac{\partial u}{\partial t} = \alpha\left(\frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} + \frac{\partial^2 u}{\partial x_3^2}\right) \tag{6}$$

$$\frac{\partial u}{\partial t} = -v\frac{\partial u}{\partial x_1}, \tag{7}$$

where $u : \mathbb{R}^3 \times \mathbb{R} \ni (\boldsymbol{x}, t) \mapsto \mathbb{R}$, and $\boldsymbol{x} = [x_1, x_2, x_3]^\top$. Both (6) and (7) are on the form of (5) with $f(\boldsymbol{x}) \equiv 0$, and are therefore linear homogeneous PDEs. Equation (6) is a standard heat equation in Euclidean 3-space, and (7) is a 1-dimensional one-way wave equation (an advection equation). Both of these equations are relevant to the extrusion process, although, it will become apparent that it is useful to us to rewrite (6) in terms of reduced cylindrical coordinates and combine them into one equation;

$$\frac{\partial u}{\partial t} = \alpha\left(\frac{\partial^2 u}{\partial x^2} + \frac{1}{r}\frac{\partial u}{\partial r} + \frac{\partial^2 u}{\partial r^2}\right) - v\frac{\partial u}{\partial x}, \tag{8}$$

where $u : \mathbb{R}^2 \times \mathbb{R} \ni (\boldsymbol{\xi}, t) \mapsto \mathbb{R}$, and $\boldsymbol{\xi} = [r, x]^\top$. This is referred to as a convection-diffusion equation, since it captures both heat diffusion and advection.

Typically when dealing with PDEs on a domain, such as $\boldsymbol{\xi} \in \mathcal{W} \subset \mathbb{R}^2$, where $\mathcal{W}$ for example represents a billet, one also imposes 'boundary conditions';

$$\left.\frac{\partial^i u}{\partial \boldsymbol{\xi}^i}\right|_{\boldsymbol{\xi}_j} = \boldsymbol{b}^\top(u(\boldsymbol{\xi})), \tag{9}$$

for some $i \in \mathbb{N} : i \leq k$, where $\boldsymbol{\xi}_j$ is on the boundary of $\mathcal{W}$, and $\boldsymbol{b}(\cdot) \in \mathbb{R}^2$ is some boundary value. For the boundaries of an object that do not dissipate energy to their surroundings at that boundary, one assigns the boundary condition

$$\left.\frac{\partial u}{\partial \boldsymbol{\xi}}\right|_{\boldsymbol{\xi}_j} = \mathbf{0}^\top. \tag{10}$$

Initial Value Problem
The reader should also be familiar with the notion of an initial value problem (IVP). In this thesis, we use IVP to mean a system of ODEs accompanied by the complete description of the dependent variable, $u(\cdot)$, for a single value of the independent variable, t, for which to solve for u over the entire domain $t \in \mathcal{D}$. That is;

$$\frac{d\boldsymbol{u}(t)}{dt} = \boldsymbol{f}(\boldsymbol{u}(t), t), \quad \left.\frac{d^i \boldsymbol{u}(t)}{dt^i}\right|_{t_0} = \boldsymbol{b}_i \ \ \forall i \in \mathbb{Z}_{0,k}, \tag{11}$$

with $u : \mathbb{R} \supseteq \mathcal{D} \mapsto \mathbb{R}^{n_u}$, where $t_0 \in \mathcal{D}$.

2.2.2 Finite Difference Schemes

Finite difference schemes are a family of schemes that attempt to solve PDEs numerically by approximating their derivatives using finite differences. The most basic examples of such are

$$\text{Forward scheme:} \quad \left.\frac{\partial u}{\partial x}\right|_{x_0} = \frac{u(x_0 + \Delta x) - u(x_0)}{\Delta x} \tag{12}$$

$$\text{Backward scheme:} \quad \left.\frac{\partial u}{\partial x}\right|_{x_0} = \frac{u(x_0) - u(x_0 - \Delta x)}{\Delta x} \tag{13}$$

$$\text{Central scheme:} \quad \left.\frac{\partial u}{\partial x}\right|_{x_0} = \frac{u(x_0 + \Delta x) - u(x_0 - \Delta x)}{2\Delta x}, \tag{14}$$

$$\tag{15}$$

which intuitively approximate the derivative due to their similarity to the definition of a derivative

$$\frac{du(x_0)}{dx} = \lim_{\Delta x \to 0} \frac{u(x_0 + \Delta x) - u(x_0)}{\Delta x}. \tag{16}$$

Similarly, the second order derivative can be approximated by

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_{x_0} = \frac{u(x_0 + \Delta x) - 2u(x_0) + u(x_0 - \Delta x)}{\Delta x^2}. \tag{17}$$

By combining different finite differences, one can discretize a PDE into a set of ODEs. Take for instance the convection-diffusion equation (8). We start by defining a grid of evenly spaced points; $\{r_1, r_2, \ldots, r_a\} \times \{x_1, x_2, \ldots, x_b\}$ with spacing Δr and Δx, at which we evaluate the dependent variable, which we denote; $u_{i,j}(t) \triangleq u(r_i, x_j; t)$. Then, (8) can be approximated by

$$\begin{aligned}\frac{du_{i,j}(t)}{dt} = \alpha\Bigg(&\frac{u_{i,j+1}(t) - 2u_{i,j}(t) + u_{i,j-1}(t)}{\Delta x^2} \\ &+ \frac{u_{i+1,j}(t) - 2u_{i,j}(t) + u_{i-1,j}(t)}{\Delta r^2} \\ &+ \frac{u_{i+1,j}(t) - u_{i-1,j}(t)}{r\Delta r}\Bigg) \\ &- v\frac{u_{i,j+1}(t) - u_{i,j}(t)}{\Delta x},\end{aligned} \tag{18}$$

which forms a system of ODEs. By providing an initial condition, $u_{i,j}(t_0) = u^0_{i,j} \;\; \forall i, j$, the PDE has been reduced to an IVP. A caveat to remark, is that (18) breaks down at the boundaries of the grid, as there are no points beyond the grid to evaluate the dependent variable at. In place of (18), one may simply use another set of schemes at these boundary points. Note that (18) may be discretized further; in time t, however, in this thesis, we do not use finite differences to approximate time evolution. Instead, *collocation* is used for this purpose, which is further discussed in Section 2.3.2.

Differential Algebraic Equations
Lastly, we briefly mention the concept of a differential algebraic equation (DAE), which is a system of equations containing both differential equations and algebraic equations;

$$\boldsymbol{F}\left(\frac{\partial u(x,y)}{\partial x}, u, x, y\right) = \mathbf{0}, \tag{19}$$

where y is an independent variable. When working with DAEs, we call the variables that appear in the differentials; *differential variables*, and the ones that only appear algebraically; *algebraic variables*. A system of ODEs can become a system of DEAs when for example assuming a steady state condition on one or more points in the grid of points, as is the case in this thesis. By assuming that $\frac{\partial u_{i,j}(t)}{\partial t} \equiv 0$ for some i, j, one obtains the following algebraic equation

$$\begin{aligned}0 = \alpha\Bigg(&\frac{u_{i,j+1}(t) - 2u_{i,j}(t) + u_{i,j-1}(t)}{\Delta x^2} \\ &+ \frac{u_{i+1,j}(t) - 2u_{i,j}(t) + u_{i-1,j}(t)}{\Delta r^2} \\ &+ \frac{u_{i+1,j}(t) - u_{i-1,j}(t)}{r\Delta r}\Bigg) \\ &- v\frac{u_{i,j+1}(t) - u_{i,j}(t)}{\Delta x},\end{aligned} \tag{20}$$

where $u_{i,j}(t)$ has become an algebraic variable.

2.3 Nonlinear Programming

This section is based on theory by Betts (2010), Biegler (2010), Biegler (2007), Rawlings et al. (2017) and Diehl and Gros (2011).

Nonlinear programming (NLP) is a type of optimization, where the output of an objective function, or cost function, $L : \mathbb{R}^n \mapsto \mathbb{R}$ is minimized by an algorithm that searches its domain. An NLP is

typically formulated as

$$\min_{\boldsymbol{w} \in \mathbb{R}^n} \quad L(\boldsymbol{w}) \tag{21a}$$

$$\text{s.t.}$$

$$\boldsymbol{c}_{\mathcal{E}}(\boldsymbol{w}) = \boldsymbol{0}, \tag{21b}$$

$$\boldsymbol{c}_{\mathcal{I}}(\boldsymbol{w}) \geq \boldsymbol{0} \tag{21c}$$

where the elements of $\boldsymbol{w}$ are considered the *decision variables*, and $\boldsymbol{c}_{\mathcal{E}}(\boldsymbol{w}) = \boldsymbol{0}$ and $\boldsymbol{c}_{\mathcal{I}}(\boldsymbol{w}) \geq \boldsymbol{0}$ are equations and inequalities that the solution, often denoted $\boldsymbol{w}^*$, must satisfy. There are several algorithms that can solve such NLPs, many of which have certain advantages over others, and some are better suited for certain classes of problems than others. In this report, the *Interior Point Optimizer (Ipopt)* algorithm (Wächter and Biegler 2006) is used, which is intended for solving large, sparse problems. The effects of utilizing other algorithms are not discussed, although this a natural continuation of this work. Another important factor to note is that the 'best' algorithm for a particular NLP can be very dependent on the specific NLP-formulation, which is not necessarily unique for a given problem. This is, among other reasons, due to *sparsity patterns* in the Jacobian of the constraint vectors (Betts 2010), which can be exploited by the algorithm. Therefore, the effectiveness of the algorithm can vary between NLP-formulations. Another aspect to note is that most nonlinear optimization algorithms, that are useful for the type of problems presented in this thesis, are *local optimizers*, which means that there may exist better, global solutions than those found by these algorithms. Nevertheless, in this thesis, the solutions found by such an optimization algorithm is considered optimal.

2.3.1 Transcription Method

When formulating a dynamic optimization problem as an NLP, it is necessary to describe the solution in terms of a finite set of variables, and this usually includes describing the system state as a finite array of variables as well. Processes such as heat diffusion in metals are inherently continuous, and the temperature state of the system is typically described by $q(\boldsymbol{p}, t) \in \mathbb{R}$, where $\boldsymbol{p} \in \mathbb{R}^3$ describes the point in space, and t denotes time. *Transcription* is the conversion of a continuous variable into a set of evaluations of that variable at different points in its domain; as

$$\boldsymbol{q}(t) = \begin{bmatrix} q(\boldsymbol{p}_1, t) \\ q(\boldsymbol{p}_2, t) \\ \vdots \\ q(\boldsymbol{p}_{n_q}, t) \end{bmatrix} = \begin{bmatrix} q_1(t) \\ q_2(t) \\ \vdots \\ q_{n_q}(t) \end{bmatrix} \in \mathbb{R}^{n_q}. \tag{22}$$

This inherently implies a transformation of the system dynamics from a partial differential equation (PDE)

$$\dot{q}(\boldsymbol{p}, t) = f(q(\boldsymbol{p}, t), q_{\boldsymbol{p}}(\boldsymbol{p}, t), \boldsymbol{u}(t), t), \quad q_{\boldsymbol{p}}(\boldsymbol{p}, t) = \frac{\partial q(\boldsymbol{p}, t)}{\partial \boldsymbol{p}} \tag{23}$$

to a set of ordinary differential equations (ODEs), as described in Section 2.2;

$$\dot{\boldsymbol{q}}(t) = \boldsymbol{f}(\boldsymbol{q}(t), \boldsymbol{u}(t), t). \tag{24}$$

The state vector is still continuous in time, and can be further transcribed in time as well. To this end, one must select a final time, t_f, which will eventually be the horizon of a dynamic optimization problem. Then a set of points must be selected between $t_0 = 0$ and $t = t_f$;

$$0 < t_1 < t_2 < \cdots < t_f. \tag{25}$$

Typically, these time points will be uniformly distributed, with a constant time step Δt, though this is not necessary in general. Varying the time step allows one to increase the resolution of the discretization only at time intervals one suspects that the state will exert complex behavior, so as to increase the accuracy of the approximation. With a constant time step, the time points can be described by

$$t_k = t_0 + k\Delta t \quad \forall k \in \mathbb{Z}_{0,N}, \tag{26}$$

where N is the number of time points after the start time, t_0. The trajectories of the state vector, $\boldsymbol{q}(t) \in \mathbb{R}^{n_q}$, and the control vector, $\boldsymbol{u}(t) \in \mathbb{R}^{n_u}$, are then naturally discretized as

$$\boldsymbol{q}(t) \longrightarrow \{\boldsymbol{q}(t_0), \boldsymbol{q}(t_1), \ldots, \boldsymbol{q}(t_N)\} = \{\boldsymbol{q}_0, \boldsymbol{q}_1, \ldots, \boldsymbol{q}_N\} \tag{27}$$

$$\boldsymbol{u}(t) \longrightarrow \{\boldsymbol{u}(t_0), \boldsymbol{u}(t_1), \ldots, \boldsymbol{u}(t_{N-1})\} = \{\boldsymbol{u}_0, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_{N-1}\}. \tag{28}$$

Before one can formulate an NLP, on must also discretize the system model (24), in terms of the transcribed state and control trajectories;

$$\boldsymbol{f}_{model}(\boldsymbol{q}_k, \boldsymbol{q}_{k+1}, \boldsymbol{u}_k, k) = \boldsymbol{0}. \tag{29}$$

That is, the discretized trajectory (27) must approximate the true solution of (24), $(\boldsymbol{q}^*(t), \boldsymbol{u}^*(t))$;

$$\left.\begin{aligned} \boldsymbol{q}_k &\approx \boldsymbol{q}^*(t_k) \\ \boldsymbol{q}^*(t_k) &= \boldsymbol{q}(t_0) + \int_{t_0}^{t_k} \boldsymbol{f}(\boldsymbol{q}(t), \boldsymbol{u}^*(t), t)\, dt \end{aligned}\right\} \ \forall k \in \mathbb{Z}_{0,N}. \tag{30}$$

A dynamic optimization problem can then be written as;

$$\min_{\boldsymbol{w}} \quad L(\boldsymbol{w}) \tag{31a}$$

$$\text{s.t.}$$

$$\boldsymbol{f}_{model}(\boldsymbol{q}_k, \boldsymbol{q}_{k+1}, \boldsymbol{u}_k, k) = \boldsymbol{0} \quad \forall k \in \mathbb{Z}_{0,N-1}, \tag{31b}$$

$$\boldsymbol{c}_{\mathcal{E}}(\boldsymbol{w}) = \boldsymbol{0}, \tag{31c}$$

$$\boldsymbol{c}_{\mathcal{I}}(\boldsymbol{w}) \geq \boldsymbol{0}, \tag{31d}$$

where if $\boldsymbol{w}$ contains both the control variables and the state variables, we call this a simultaneous method. If the state variables are implicitly given, and $\boldsymbol{w}$ only contains the control variables, we call it a sequential method.

2.3.2 Direct Collocation

Direct collocation is a type of simultaneous method for dynamic optimization problems of type (31), where the state trajectories are approximated by a collocation based integration scheme (Biegler 2010) (Betts 2010) (Rawlings et al. 2017). The integration scheme is derived from collocation techniques, although, they are equivalent to implicit Runge-Kutta schemes(IRK). A major advantage of IRKs, over explicit integration schemes, is that they are stable numerical integrators, meaning that the integration error is bounded. One may therefore use these integrators to obtain a highly accurate numerical integration with relatively few time steps. The stable nature of the implicit integration schemes make them suited for stiff systems, as explicit integration schemes require very fine discretization to handle all modes of the stiff system.

The typical way of building collocation polynomial starts by defining d 'collocation points' on each time interval of the discretization of the state and control trajectories;

$$t_{k,i} \in [t_k, t_{k+1}] \ \ \forall i \in \mathbb{Z}_{1,d}, \ \ \forall k \in \mathbb{Z}_{0,N-1}, \tag{32}$$

and define

$$t_{k,0} = t_k \ \ \forall k \in \mathbb{Z}_{0,N-1}. \tag{33}$$

One then defines the Lagrange polynomials

$$l_{k,i}(t) = \prod_{j=0, j\neq i}^{d} \frac{t - t_{k,j}}{t_{k,i} - t_{k,j}} \ \ \forall i \in \mathbb{Z}_{0,d}, \ \ \forall k \in \mathbb{Z}_{0,N-1}, \tag{34}$$

which have the property

$$l_{k,i}(t_{k,j}) = \begin{cases} 1, i = j \\ 0, i \neq j \end{cases}. \tag{35}$$

By introducing the parameters $\boldsymbol{v}_{k,i} \in \mathbb{R}^{n_q} \ \ \forall i \in \mathbb{Z}_{0,d}$, one can obtain

$$\boldsymbol{v}_{k,i} \cdot l_{k,i}(t_{k,j}) = \begin{cases} \boldsymbol{v}_{k,i}, \, i = j \\ \boldsymbol{0}, \, i \neq j \end{cases} \in \mathbb{R}^{n_q}. \tag{36}$$

To obtain the 'collocation polynomial' for time interval k, one takes the sum of (36) for all times $t_{k,i} \ \ \forall i \in \mathbb{Z}_{0,d}$ to get a Lagrange polynomial;

$$\boldsymbol{p}_k(t, V_k) = \sum_{i=0}^{d} \boldsymbol{v}_{k,i} l_{k,i}(t), \tag{37}$$

where $V_k = \{\boldsymbol{v}_{k,0}, \boldsymbol{v}_{k,1}, \ldots, \boldsymbol{v}_{k,d}\}$. The polynomial then inherits the desired property

$$\boldsymbol{p}_k(t, V_k) = \begin{cases} \boldsymbol{v}_{k,i} \, , \, t = t_{k,i} \ \ \forall i \in \mathbb{Z}_{0,d} \\ \boldsymbol{p} \in \mathbb{R}^{n_q} \, , \, \text{otherwise} \end{cases}. \tag{38}$$

One can then set the initial value of the polynomial, $\boldsymbol{p}_k(t_{k,0}, V_k)$ to the state value at the start of that interval, $\boldsymbol{q}_k$, by simply setting $\boldsymbol{v}_{k,0} = \boldsymbol{q}_k$. By choosing V_k such that

$$\dot{\boldsymbol{p}}_k(t_{k,i}, V_k) = \boldsymbol{f}(\boldsymbol{p}_k(t_{k,i}, V_k), \boldsymbol{u}_k) \ \ \forall i \in \mathbb{Z}_{1,d}, \tag{39}$$

the collocation polynomial approximates the state trajectory on the interval $[t_k, t_{k+1}]$. Notice that the collocation polynomial is easily differentiable. Therefore the state trajectory can be approximated by finding the $V = \{V_1, V_2, \ldots, V_{N-1}\}$ that solves the system

$$\begin{bmatrix} \boldsymbol{q}_k - \boldsymbol{v}_{k,0} \\ \dot{\boldsymbol{p}}_k(t_{k,1}, V_k) - \boldsymbol{f}(\boldsymbol{p}_k(t_{k,1}, V_k), \boldsymbol{u}_k) \\ \dot{\boldsymbol{p}}_k(t_{k,2}, V_k) - \boldsymbol{f}(\boldsymbol{p}_k(t_{k,2}, V_k), \boldsymbol{u}_k) \\ \vdots \\ \dot{\boldsymbol{p}}_k(t_{k,d}, V_k) - \boldsymbol{f}(\boldsymbol{p}_k(t_{k,d}, V_k), \boldsymbol{u}_k) \end{bmatrix} = \boldsymbol{0} \ \ \forall k \in \mathbb{Z}_{0,N-1}, \tag{40}$$

which is referred to as the 'collocation equation', and evaluating the polynomial at (t_{k+1}, V_k) for all time intervals;

$$\boldsymbol{q}_{k+1} = \boldsymbol{p}(t_{k+1}, V_k) \ \ \forall k \in \mathbb{Z}_{0,N-1}. \tag{41}$$

One can rewrite the polynomials to be functions of a scaled variable τ, by mapping τ to t for each interval, using

$$t = (\tau \cdot \Delta t + t_k) \ \ \forall k \in \mathbb{Z}_{0,N-1}. \tag{42}$$

All polynomials $\boldsymbol{p}_k(t, V_k) \ \ \forall k \in \mathbb{Z}_{0,N-1}$ can then be written as a single polynomial $\boldsymbol{p}(\tau, V_k)$, which simplifies the implementation. Their derivatives are transformed by

$$\frac{d\boldsymbol{p}(\tau, V_k)}{d\tau} = \frac{d\boldsymbol{p}_k(t, V_k)}{dt} \cdot \frac{dt}{d\tau} \tag{43a}$$

$$\frac{dt}{d\tau} = \frac{d}{d\tau}\big(\tau \cdot \Delta t + t_k\big) = \Delta t \tag{43b}$$

$$\Rightarrow \quad \boldsymbol{p}'(\tau, V_k) = \dot{\boldsymbol{p}}_k(t, V_k) \cdot \Delta t. \tag{44}$$

Equation (40) can then be rewritten as

$$\begin{bmatrix} \boldsymbol{q}_k - \boldsymbol{v}_{k,0} \\ \boldsymbol{p}'(\tau_1, V_k) - \Delta t \cdot \boldsymbol{f}(\boldsymbol{p}(\tau_1, V_k), \boldsymbol{u}_k) \\ \boldsymbol{p}'(\tau_2, V_k) - \Delta t \cdot \boldsymbol{f}(\boldsymbol{p}(\tau_2, V_k), \boldsymbol{u}_k) \\ \vdots \\ \boldsymbol{p}'(\tau_d, V_k) - \Delta t \cdot \boldsymbol{f}(\boldsymbol{p}(\tau_d, V_k), \boldsymbol{u}_k) \end{bmatrix} = \boldsymbol{0} \ \ \forall k \in \mathbb{Z}_{0,N-1}. \tag{45}$$

Diehl and Gros (2011) states that collocation can result in very high orders of integration by choosing the collocation points carefully, which they explain by quadrature rules. This leads to many choices of collocation points that yield different properties.

The Direct Collocation NLP

The final NLP will optimize over all state variables at times $t_k \ \ \forall k \in \mathbb{Z}_{0,N}$ and control variables at times $t_k \ \ \forall k \in \mathbb{Z}_{0,N-1}$, and in addition, all the intermediate state variables $v_{k,i} \ \ \forall i \in \mathbb{Z}_{1,d}, \ \ \forall k \in \mathbb{Z}_{0,N-1}$. We define $\boldsymbol{w} = [\boldsymbol{u}_0^\top, \ldots, \boldsymbol{u}_{N-1}^\top, \boldsymbol{q}_0^\top, \ldots, \boldsymbol{q}_N^\top, \boldsymbol{v}_{0,0}^\top, \ldots, \boldsymbol{v}_{0,d}^\top, \ldots, \boldsymbol{v}_{N-1,0}^\top, \ldots, \boldsymbol{v}_{N-1,d}^\top]^\top$, and the complete NLP formulation becomes

$$\min_{\boldsymbol{w}} \quad L(\boldsymbol{w}) \tag{46a}$$

$$\text{s.t.}$$

$$\begin{bmatrix} \boldsymbol{q}_k - \boldsymbol{v}_{k,0} \\ \boldsymbol{p}'(\tau_1, V_k) - \Delta t \cdot \boldsymbol{f}(\boldsymbol{p}(\tau_1, V_k), \boldsymbol{u}_k) \\ \boldsymbol{p}'(\tau_2, V_k) - \Delta t \cdot \boldsymbol{f}(\boldsymbol{p}(\tau_2, V_k), \boldsymbol{u}_k) \\ \vdots \\ \boldsymbol{p}'(\tau_d, V_k) - \Delta t \cdot \boldsymbol{f}(\boldsymbol{p}(\tau_d, V_k), \boldsymbol{u}_k) \end{bmatrix} = \boldsymbol{0} \quad \forall k \in \mathbb{Z}_{0,N-1}, \tag{46b}$$

$$\boldsymbol{q}_{k+1} - \boldsymbol{p}(1, V_k) = \boldsymbol{0} \quad \forall k \in \mathbb{Z}_{0,N-1}, \tag{46c}$$

$$\boldsymbol{c}_{\mathcal{E}}(\boldsymbol{w}) = \boldsymbol{0}, \tag{46d}$$

$$\boldsymbol{c}_{\mathcal{I}}(\boldsymbol{w}) \geq \boldsymbol{0}, \tag{46e}$$

Note that adding $\boldsymbol{v}_{k,0}$ to the set of decision variables is redundant, since the state variable $\boldsymbol{q}_k$ can be used directly in $V_k = [\boldsymbol{q}_k, \boldsymbol{v}_{k,1}, \ldots, \boldsymbol{v}_{k,d}]$, thereby also removing the need for the first line in (46b). Also, to reflect an initial value problem, the state should be known at some point i; $\boldsymbol{q}_i = \boldsymbol{q}_{IV}$, thus negating the need to include this variable as a decision variable. Nevertheless, it is common for MPC purposes to include the initial value as a decision variable and instead add the inequality constraint $\boldsymbol{q}_{IV} \leq \boldsymbol{q}_i \leq \boldsymbol{q}_{IV}$, or the equality constraint $\boldsymbol{q}_i = \boldsymbol{q}_{IV}$.

3 Extrusion Process

This section breaks down the extrusion process, as seen from the complete extrusion cycle perspective. The extrusion process/cycle can be divided into three main phases;

1. the heating phase,
2. the transit phase,
3. and the extrusion phase.

The extrusion cycle is illustrated in Fig. 8, where a billet starts it cycle by being inserted in to a heater, then, after the heating phase is over, being transported during the transit phase, before finally arriving at the extrusion press, ready to begin its extrusion phase.

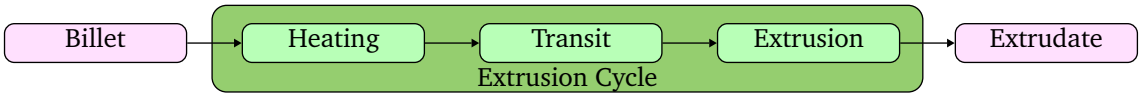


Figure 8: Illustration of the extrusion cycle of a billet. This is the cycle subject to optimization by the ECOL.

The heating phase requires an available heater. The extrusion plant of the case study has two heaters working in parallel, such that a billet may begin its cycle while another billet is still being heated. A heater model is described by Gabrielsen (2022), and it was shown that the use of optimal control techniques to regulate the heater to a reference taper is simple and effective. In this thesis, we make the assumption that the feasible reference tapers under this regulator scheme are or can easily be understood. The transit phase is assumed to be trivially modelled as an extension of the extrusion phase model from this thesis, or the already implemented heater model.

We define time frame of an extrusion cycle in the following way:

Phase:	**Heating**	**Transit**	**Reload**	**Extrusion**
Time Interval:	$[-t_h, 0] - t_t - t_{RL}$	$[-t_t, 0] - t_{RL}$	$[-t_{RL}, 0]$	$[0, t_f]$
Duration:	$t_h \geq 0$	$t_t \geq 0$	$t_{RL} \geq 0$	$t_f \geq 0$

where t_h is referred to as the 'heating time', t_t is the 'transit time', t_{RL} is a small 'reload time', and t_f is the 'final time' or the 'extrusion time'. The total cycle time is then

$$t_c = t_h + t_t + t_{RL} + t_f, \tag{47}$$

In practice the extrusion cycles will overlap, as illustrated in Fig. 9. Therefore the extrusion period, $\mathcal{T}_{ext}$, is not necessarily, and in fact usually not, the same as the total cycle time; $\mathcal{T}_{ext} \neq t_c$, and is closer to the extrusion time plus reload time; $\mathcal{T}_{ext} \approx t_f + t_{RL}$, unless an extrusion cycle is delayed due to the unavailability of heaters. The production rate is therefore, only dependent on the extrusion time, since the reload time is constant. This idea is explored further in Section 5.

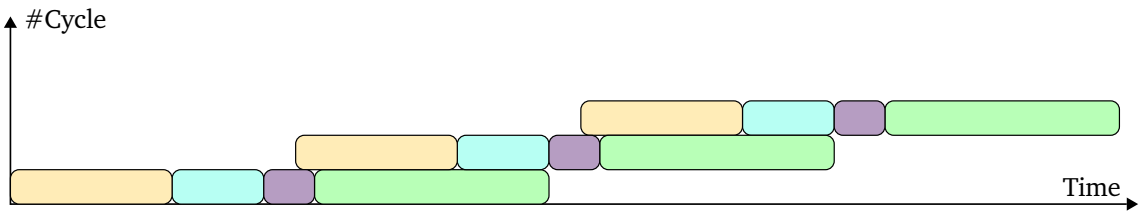


Figure 9: Phases: -Heater, -Transit, -Extrusion, -Reload Time

Mostly, in this thesis, we ignore the small reload phase, as there is only a trivial distinction between the reload phase and the transit phase, and the two may be seen as one larger phase. Both phases consist solely of 'waiting', from the perspective of the ECOL.

The following table summarizes the time related parameters describing the extrusion cycle:

Parameter	Unit	Description
t_h	s	Duration of the heater phase
t_t	s	Duration of the transit phase (constant)
t_f	s	Duration of the extrusion phase and end time of extrusion cycle
t_c	s	Duration of the one extrusion cycle
t_{RL}	s	Duration of the the reload phase (constant)
$t_0 = -t_h - t_t - t_{RL}$	s	Start time of the extrusion cycle
$\mathcal{T}_{ext}$	s	Extrusion period[a]

[a] The time between the start of two consecutive extrusion cycles.

Table 1: Time related parameters that describe the extrusion cycle.

The following subsections describe the different phases.

3.1 Heating phase

In the heating phase, a billet is placed into an induction heater as the first step in its extrusion cycle. The main components of the heater are a chamber that closely encapsulates the billet, and a set of inductors that uses electromagnetic induction to heat the billet at uniformly distributed 'zones' across its length. The zones are the intervals along the length of the billet that are directly affected by the induction coils, and nearly cover the entire surface of the billet. Fig. 10 shows a sketch of a typical induction coil arrangement. The number of coils in an induction heater may vary, but is typically greater than four, and the affected zones are usually close together, to avoid intermediate cold regions. Each coil has its own controlled voltage, allowing control of the power output from each individual coil, resulting in a number of controlled variables equal to the number of coils. It is this property that allows one to preheat the billet to a temperature distribution, called *temperature taper* or just *taper*, where the number of coils determine the potential complexity of the taper. Note that the radial temperature variations are also part of the temperature taper. In fact, the induction coils mainly affect the outer part of the billet, naturally inducing a radial taper in the billet. These radial variations can be some what accounted for by regarding the time radial heat diffusion get to occur. Typically, towards the end of the heating phase, the induction coils are idle for some time, to allow the radial gradients to even out before the billet is sent to the extrusion press. This time is referred to as the 'saturation time'. A short re-heating phase usually takes place at the very end of the heating process, to re-heat the billet, and counter the heat loss that occurs during transit.

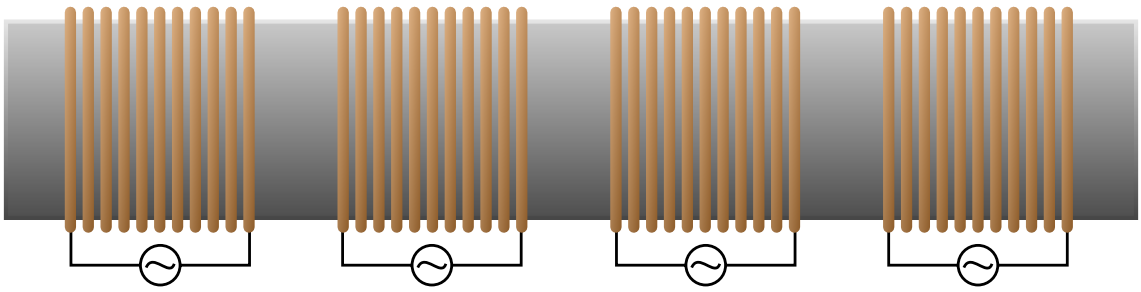


Figure 10: Illustration of the induction heater, showing an aluminium billet and four surrounding induction coils, each with their own controlled AC voltage. By controlling the power of each coil individually, one may produce a temperature taper in the billet.

3.2 Transit Phase

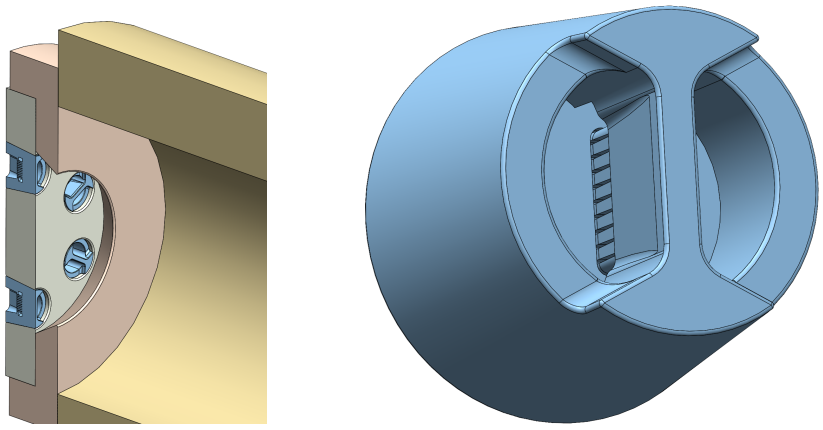
The transit phase only consists of the transportation of the billet from the induction heater to the extrusion press. This process is usually automated, and does not have many controllable variables. The most interesting part of the transit phase is how the temperature taper of the billet changes during the phase, due to internal heat conduction and dissipation of thermal energy to the surroundings. This is primarily affected by the duration of the transit phase, which is not considered a control variable in this thesis. The ambient temperature does also affect how the temperature taper changes during the transit phase and must be accounted for by the optimization layer. Lastly, the transit phase is assumed to be parallelizable. That is, several billet may be in their respective transit phases simultaneously, so as to not stall production.

3.3 Extrusion Phase

After the transit phase, the billet arrives at the extrusion press and is placed into the container, ready to begin extrusion. In practice, there is a short 'reload time', where the butt of the previous billet is sheered off, and the new billet is inserted into the container. The extrusion phase is by far the most complex phase, and is where the actual extrusion happens. The particular extrusion press considered in this thesis is a cylindrical press owned by Hydro Extruded Solution, and is located in Tønder, Denmark. The press uses the direct extrusion principle with hot extrusion, and, as is usual, consists of the four main parts; ram, container, billet and die. In addition, the die is an assembled component, consisting of several sub-components. In this thesis, three sub-components of the die assembly is considered, which are;

- the die ring,
- the die (as part of the 'die assembly', which has previously been referred to as just 'die'),
- and the die holder.

A cut view of a rudimentary recreation of the die assembly is presented in Fig. 11a. The figure shows a six fold symmetry, with six 'cavities', or 'dies', whereas the real die assembly uses ten cavities. The die is depicted in Fig. 11b, alluding to the complex nature of the geometry.



(a) The die assembly. (b) The die.

Figure 11: A cut view of the rudimentary recreation of the die assembly considered in this thesis at the end of a container (yellow), consisting of 6 dies (blue), a die holder (green), and a die ring (orange). Fig. (b) provides a closer look at the crudely reconstructed die.

The die also has cooling channels, with the possibility of using liquid nitrogen to cool the die during extrusion, to compensate the heat generation in the aluminium. Each die is connected to the same

cooling system, which means that there is only one valve to vary the coolant flow for all the dies. Thus the dies cannot be individually cooled during extrusion.

The hydraulic ram is controlled by a PID controller. The PID controlled hydraulic ram system is constrained in terms of both speed and acceleration. I addition, the ram speed cannot be negative, that is, the hydraulic ram cannot retract during the extrusion phase. This has to do with safety and product quality concerns. In this thesis, we introduce a safety margin, such that the velocity must be at least a specified value, $\underline{v}_{ram}$, above zero, except at the start of the extrusion phase, where the ram speed is zero.

The control signal to the cooling system is in the form of a valve opening set point as a decimal fraction, $z_{valve} \in [0, 1]$. The maximum allowed opening is 20%, or $z_{valve} = 0.2$.

The control constraints are given in Table 2.

Control constraints		
$v_{ram}(0)$	$= 0mm/s$	
$v_{ram}(t)$	$> 0mm/s$	$\forall t > 0$
$v_{ram}(t)$	$\geq \underline{v}_{ram} > 0mm/s$	$\forall t \geq \underline{t}_{ram} > 0$
$v_{ram}(t)$	$\leq 5mm/s$	$\forall t \geq 0$
$\lvert\dot{v}_{ram}(t)\rvert$	$\leq 0.4mm/s^2$	$\forall t \geq 0$
$z_{valve}(t)$	$\in [0, 0.2]$	$\forall t \geq 0$
$\lvert\dot{z}_{valve}(t)\rvert$	$\leq \overline{z}_{rate}$	$\forall t \geq 0$

Table 2: Constraints of the PID controlled hydraulic ram speed, and the coolant flow valve opening.

The complete control vector during the extrusion phase is

$$\boldsymbol{u}_{true}(t) = \begin{bmatrix} v_{ram,setpoint}(t) \\ z_{valve,setpoint}(t) \end{bmatrix} \in \mathbb{R}^2, \quad t \in [0, t_f], \tag{48}$$

where $v_{ram,setpoint}(t)$ is the set point for the ram speed, $z_{valve,setpoint}(t)$ is the set point for the valve opening, and t_f is the final time of the extrusion phase. The dynamics between the set points and actual values are not modelled, and for the remainder of the thesis, we assume that the true values are identically equal to the set points; $v_{ram} \equiv v_{ram,setpoint}(t)$ and $z_{valve} \equiv z_{valve,setpoint}$, and subsequently, that we can control the true values directly. Note that the initial billet temperature is also considered a control variable for the extrusion phase, though not *during* extrusion. For the purposes of this thesis, the state of the extrusion process during the extrusion phase is considered to be the temperature distribution of the billet, container, the downstream aluminium flow, and the die assembly.

The container has an embedded heating system, and for modelling purposes, we assume that the outer layer of the container has a fixed temperature of 390^oC, thus there is no loss from the container to the ambient air. Similarly, the die assembly is held in place by a component that, based on experimental data, is assumed to have a constant temperature of 250^oC. The heat flow from the die is then assumed to be proportional to

$$\frac{250 - T_{die}}{\tau_{die|dieholder}}, \tag{49}$$

where T_{die} is the relevant die temperature, and $\tau_{die|dieholer}$ is an experimentally found transfer speed parameter.

3.3.1 Temperatures

Data from within the research project is used to find some initial temperatures for a typical extrusion phase. Table 3 lists the initial temperatures at the start the extrusion phase for various

"sections" (defined in Section 4) within the extrusion press, as assumed in this thesis. The temperatures are assumed to be uniformly distributed within each section at the start of the extrusion phase.

Section	Initial Temperature
Container	447^oC
Feeder	455^oC
Port	480^oC
Peak	440^oC
Plate	480^oC
Die	487^oC

Table 3: The assumed initial temperatures of the extrusion phase in various "sections" of the extrusion press. The sections are defined in Section 4.

During the extrusion phase, all temperatures must be strictly below the melting point of aluminium; $T_{max} < T_{melt}$. This temperature varies with different alloys, though for simplicity, we assume that this is above $T_{melt} > 660^oC$, and impose the constraint $T \leq T_{max} = 660^oC$ for any system temperature T. Of course, no temperature can be below $0K/-273.15^oC$ neither. In addition, the temperature at the very end of the aluminium flow stream, as it exits the die (referred to as "peak temperature"), is critical to the product quality. In this thesis, we require that the peak temperature is in the interval $T_{peak} \in \mathcal{F}_{peak} = [600^oC, 610^oC]$. If this requirement is satisfied, the extrusion is considered isothermal at the desired temperature. Note that, since the initial temperature is outside this region, $T_{peak}(t = 0) \notin [600^oC, 610^oC]$, this requirement is not feasible, thus some deviation is acceptable in the beginning of the extrusion phase. The temperature constraints are summarized in Table 4

Temperature Constraints	
$T(t) \in \langle -273.15^oC, 660^oC]$	$\forall t \in [t_0, t_f]$
$T_{peak}(t) \in [600^oC, 610^oC]$	$\forall t \in [\delta t, t_f]$
$T_{peak}(t) \in [-273.15^oC, 610^oC]$	$\forall t \in [0, t_f]$

Table 4: Constraints on temperatures during extrusion, where T is used to denote any temperature, T_{peak} denotes the temperature at the exit of the die, and $\delta t \in \langle 0, t_f \rangle$ denotes some time into the extrusion phase.

4 Extrusion Model

The model used in this thesis is based on modelling work by Halås (unpublished). Various modifications to their work are made to adapt the model for use with direct collocation, and to suit the purposes of the thesis. The model is based on partial differential equations (PDEs) describing temperature evolution in metals, accompanied by boundary conditions. A more detailed look at how such equations are derived, and how they are adapted for a direct extrusion process is given in Özışık (1993), Aukrust and LaZghab (2000) and Cuéllar Matamoros (1999). Parts of the model are confidential, thus some details are left undocumented, and the model is described on a general level.

This section describes the extrusion phase model, as this is the model that will be the basis of the optimization problem presented later in the thesis. The heating phase model is described by Gabrielsen (2022), and is reminiscent of the extrusion model, without some of the challenging complexities of the extrusion phase. The transit phase remains unmodelled, but is an important step for future work, to use in verification of feasibility of the optimal solutions found in this thesis. This idea is explored further in Section 5 and 7.

The parameters used in the model are summarized in Table 5.

Parameter	Unit	Description
$h_{al\|steel}$	$W/(m^2K)$	Heat transfer coefficient between aluminium and steel
$h_{al\|a}$	$W/(m^2K)$	Heat transfer coefficient between aluminium and environment
α_{al}	m^2/s	Thermal diffusivity of aluminium
α_{steel}	m^2/s	Thermal diffusivity of steel
κ_{al}	$W/(mK)$	Thermal conductivity of aluminium
κ_{steel}	$W/(mK)$	Thermal conductivity of steel
$c_{p,al}$	$J/(kgK)$	Heat capacity of aluminium
$c_{p,steel}$	$J/(kgK)$	Heat capacity of steel
ρ_{al}	kg/m^3	Mass density of aluminium
ρ_{steel}	kg/m^3	Mass density of steel
ρ_{LN2}	kg/m^3	Mass density of liquid nitrogen
Q_{LN2}	W	Cooling capacity of liquid nitrogen
η_{LN2}	$-$	Efficiency of LN_2-cooling
$\dot{m}_{LN2}$	kg/s	Mass flow of liquid nitrogen
$\Delta h_{vap,LN_2}$	J/kg	Vaporization heat of liquid nitrogen
k_{max}	m^3/s	Maximum volumetric flow of liquid nitrogen in cooling channels
T_a	K	Ambient temperature
$T_{Ref,container}$	K	Reference temperature for container heating element
$A_{i\|j}$	m^2	Interfacing are between i and j
R_{billet}	m	Billet radius and inner radius of container
L_{billet}	m	Billet length[a]
$R_{container}$	m	Outer radius of container
$L_{container}$	m	Length of container
R_{feeder}	m	Radius of feeder section
L_{feeder}	m	Length of feeder section
R_{port}	m	Radius of port section
L_{port}	m	Length of port section
h_{peak}	m	Height of peak section
w_{peak}	m	Width of peak section
L_{peak}	m	Length of peak section
R_{plate}	m	Outer radius of plate section
L_{plate}	m	Length of plate section
R_{die}	m	Outer radius of die section
L_{die}	m	Length of die section
$R_{extrusion}$	m	Extrusion Radius
E_{ratio}	$-$	Extrusion Ratio
$N_{cavities}$	$-$	Number of cavities
R_f	$J/(molK)$	The gas constant[b]
$Q_f > 0$	kJ/mol	Activation energy[b]
$A_f > 0$	s^{-1}	A temperature independent parameter[b]
$n_f > 0$	$-$	A temperature independent parameter[b]
$\alpha_f > 0$	MPa^{-1}	A temperature independent parameter[b]

[a] This parameter is time dependent, and refers to the initial length when written without time as an argument.

[b] Related to flow of aluminium and resulting shear stress and strain (Sellars and Tegart (1972) and Sheppard and Wright (1979)), as cited by Aukrust and LaZghab (2000)

Table 5: Model parameters.

4.1 Continuous Model

The model divides the system into *'sections'* with their own governing PDEs and coordinate systems, and that interact via boundary conditions. A summary of these sections is given in Table 6, and Fig. 12 shows their arrangement in space. All sections are considered to be cylindrical about one of two center axes, depicted in red and yellow in Fig.12, with the exception of the peak- and exit-sections, which are assumed to be rectangular cuboids (their cross-sectional areas still assumed to be that of the MPE-profiles shown in Fig 1b).

Section name	Material	Variable	Description
"billet"	Aluminium	$T_{billet}(\cdot)$	The billet.
"container"	Steel	$T_{container}(\cdot)$	The container.
"feeder"	Aluminium	$T_{feeder}(\cdot)$	The part of the downstream aluminium flow that is inside the die ring.
"port"	Aluminium	$T_{port}(\cdot)$	The part of the downstream aluminium flow within the die itself.
"peak"	Aluminium	$T_{peak}(\cdot)$	The aluminium at the exit of the die, where the product profiles are imposed.
"plate"	Steel	$T_{plate}(\cdot)$	The part of the die that is closest to the peak-section. This section also contains cooling channels.
"die"	Steel	$T_{die}(\cdot)$	The remaining part of the die, surrounding the plate- and port-sections.
"Exit"	Aluminium		The aluminium extrudate. This section is NOT modelled in this thesis.

Table 6: Model section descriptions.

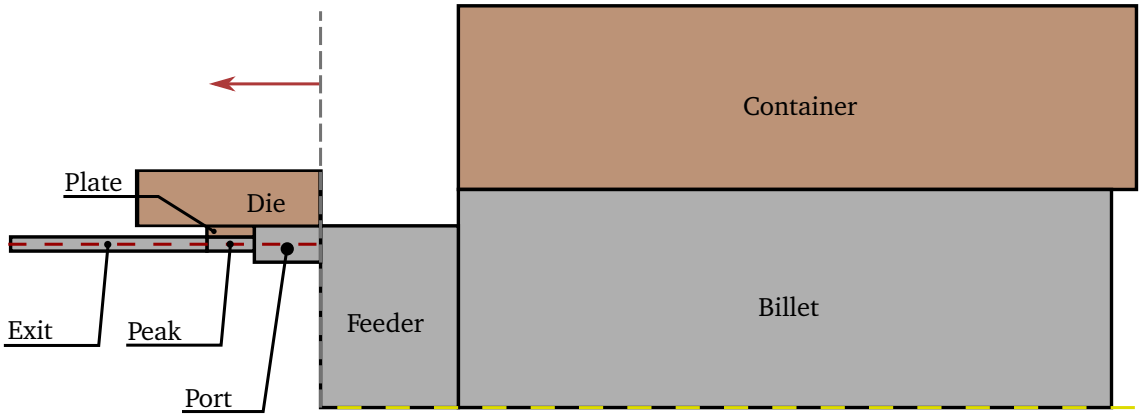


Figure 12: Depiction of the various model sections, as defined by the model. Steel sections are shown in brown, and aluminium sections are shown in gray. The red arrow indicates the direction of flow, or 'extrusion direction'. The vertical dashed gray line separates the sections that are common for the whole system, and the sections that represent a specific cavity. The yellow and red dashed lines represent the main and secondary center axes respectively.

The PDEs used to model the different sections are reduced cylindrical dynamic heat equations on the form

$$\frac{\partial T(\bar{\boldsymbol{\xi}};t)}{\partial t} = v(\bar{\boldsymbol{\xi}};t)\frac{\partial T(\bar{\boldsymbol{\xi}};t)}{\partial \bar{x}} + \alpha\left(\frac{1}{\bar{r}}\frac{\partial}{\partial \bar{r}}\left(\bar{r}\frac{\partial T(\bar{\boldsymbol{\xi}};t)}{\partial \bar{r}}\right) + \frac{\partial^2 T(\bar{\boldsymbol{\xi}};t)}{\partial \bar{x}^2}\right) + \tilde{\Phi}(\bar{\boldsymbol{\xi}};T;\bar{v}) + \tilde{\Psi}(\bar{\boldsymbol{\xi}};T;\bar{v}), \quad (50)$$

where $T(\bar{\boldsymbol{\xi}};t) \in \mathbb{R}$ and $v(\bar{\boldsymbol{\xi}};t) \in \mathbb{R}$ are the temperature and velocity of the material at point $\bar{\boldsymbol{\xi}} = [\bar{r},\bar{x}]^\top \in \mathbb{R}^2$ and time $t \in \mathbb{R}$, respectively, and α is the thermal diffusivity of the material. For more on how (50) is derived, we refer the reader to Cuéllar Matamoros (1999) and Özışık (1993).

The terms; $\tilde{\Phi}$ and $\tilde{\Psi}$, represent the heat generation from the viscous dissipation and reduction work respectively, which are dependent on the average velocity in the section, $\bar{v}$, and are such that

$$\bar{v} = 0 \Rightarrow \tilde{\Phi}, \tilde{\Psi} = 0. \tag{51}$$

The radial component of the velocity of the aluminium is assumed to be zero, such that flow only occurs in the axial direction, thus the velocity $v(\bar{\boldsymbol{\xi}}; t)$ is modelled as a scalar value, representing the axial speed in the extrusion direction. Notice that the extrusion direction is in the negative direction of the axial dimension, thus the sign of the translation/advection term is reversed. In addition, all steel components are assumed to be rigid, such that $\vec{v}_{steel}(t) \equiv \vec{0}$, which combined with (51) reduces the steel section models to

$$\frac{\partial T(\bar{\boldsymbol{\xi}}; t)}{\partial t} = \alpha_{steel} \left(\frac{1}{\bar{r}} \frac{\partial}{\partial \bar{r}} \left(\bar{r} \frac{\partial T(\bar{\boldsymbol{\xi}}; t)}{\partial \bar{r}} \right) + \frac{\partial^2 T(\bar{\boldsymbol{\xi}}; t)}{\partial \bar{x}^2} \right). \tag{52}$$

The dynamic heat equation, (50), uses reduced cylindrical coordinates, $\bar{\boldsymbol{\xi}} = [\bar{r}, \bar{x}]^\top$, where $\bar{r}$ is the radial dimension and $\bar{x}$ the axial dimension, as shown in Fig. 13, with $r = 0$ being at their respective center axes, shown in Fig. 12. The zero of the axial dimension is defined separately for each section, and is located at the leftmost point of the section, such that the section spans the interval $[0, L_{section}]$ in its axial dimension. This is a natural way of choosing the coordinate frames, since all modelled sections except for the peak-section are cylindrical. The interactions between the different sections are modelled as boundary conditions on the PDEs. These conditions are presented in Table 7, Table 8, Table 9, Table 10, Table 11, Table 12 and Table 13. For simplicity of notation, the coordinates and time of all temperatures $T_{section}(\bar{\boldsymbol{\xi}}; t)$ are omitted in the boundary condition tables, except for specific coordinate values. For example, we write $T_{billet,(\bar{r}=0)}$ to mean $T_{billet}(0, \bar{x}; t) \ \forall \bar{x}, t \in [0, L_{billet}] \times [0, t_f]$. Any omitted value and corresponding values in other sections are deducible from Fig 12. One shortcoming of the non-symmetric peak section is that some of its boundary conditions remain semi-defined in the continuous space model, although, this does not carry over to the discrete space model that we derive in Section 4.2. We do not model all intersections rigorously in continuous space, and $I_{A|B}$ is used in place of coordinates of boundary conditions where the coordinates are not defined in reduced cylindrical form.

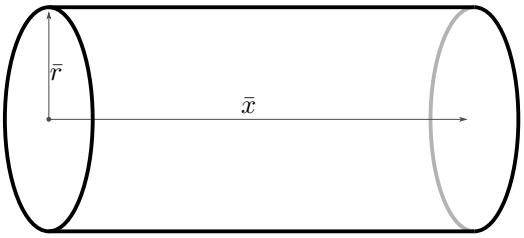


Figure 13: The cylindrical coordinate frame used to describe points within the billet and various other components of the extrusion press.

Billet	
$\frac{T_{billet}}{\partial \bar{r}}\Big\vert_{(\bar{r}=R_{billet})}$	$= \frac{h_{al\vert steel}}{\kappa_{al}}\left(T_{container,(\bar{r}=0)} - T_{billet}\right)$
$\frac{T_{billet}}{\partial \bar{x}}\Big\vert_{(\bar{x}=L_{billet}(t)),(\bar{x}=0\wedge\bar{r}>R_{feeder})}$	$= 0$
$\frac{T_{billet}}{\partial \bar{r}}\Big\vert_{(\bar{r}=0)}$	$= 0$
$\frac{T_{billet}}{\partial \bar{x}}\Big\vert_{(\bar{x}=0\wedge\bar{r}\leq R_{feeder})}$	$= \frac{T_{feeder}}{\partial \bar{x}}\Big\vert_{(\bar{x}=L_{feeder})}$

Table 7: Billet section boundary conditions.

Container	
$\frac{T_{container}}{\partial \bar{r}}\Big\vert_{\bar{r}=0}$	$= \frac{h_{al\vert steel}}{\kappa_{steel}}\left(T_{billet,(\bar{r}=R_{billet})} - T_{container}\right)$
$\frac{T_{container}}{\partial \bar{x}}\Big\vert_{(\bar{x}=L_{container}),(\bar{x}=0)}$	$= 0$
$T_{container,(\bar{r}=R_{container})}$	$= T_{Ref,container}$

Table 8: Container section boundary conditions.

Feeder	
$\frac{T_{feeder}}{\partial \bar{r}}\Big\vert_{(\bar{r}=R_{feeder}),(\bar{r}=0)}$	$= 0$
$\frac{T_{feeder}}{\partial \bar{x}}\Big\vert_{(\bar{x}=0\wedge\bar{r}\in I_{feeder\vert port})}$	$= \frac{T_{port}}{\partial \bar{x}}\Big\vert_{(\bar{x}=L_{port})}$

Table 9: Feeder section boundary conditions.

Port	
$\frac{T_{port}}{\partial \bar{r}}\Big\vert_{(\bar{r}=0)}$	$= 0$
$\frac{T_{port}}{\partial \bar{r}}\Big\vert_{(\bar{r}=R_{port})}$	$= \frac{h_{al\vert steel}}{\kappa_{al}}\left(T_{die,(\bar{r}=0)} - T_{port}\right)$
$\frac{T_{port}}{\partial \bar{x}}\Big\vert_{(\bar{x}=0 \wedge \bar{r} \in I_{port\vert plate})}$	$= \frac{h_{al\vert steel}}{\kappa_{al}}\left(T_{plate,(\bar{x}=L_{plate})} - T_{port}\right)$
$\frac{T_{port}}{\partial \bar{x}}\Big\vert_{(\bar{x}=0 \wedge \bar{r} \in I_{port\vert peak})}$	$= \frac{T_{peak}}{\partial \bar{x}}\Big\vert_{(\bar{x}=L_{peak} \wedge \bar{r} \in I_{peak\vert port})}$

Table 10: Port section boundary conditions.

Peak	
$\frac{T_{peak}}{\partial \bar{r}}\Big\vert_{(\bar{r}=0)}$	$= 0$
$\frac{T_{peak}}{\partial \bar{x}}\Big\vert_{(\bar{x}=0)}$	$= 0$
$\frac{T_{peak}}{\partial \bar{r}}\Big\vert_{(\bar{r}=I_{peak\vert plate})}$	$= \frac{h_{al\vert steel}}{\kappa_{al}}\left(T_{plate,(\bar{r}=I_{plate\vert port})} - T_{peak}\right)$

Table 11: Peak section boundary conditions.

Plate	
$\frac{T_{plate}}{\partial \bar{x}}\Big\vert_{(\bar{x}=0)}$	$= 0$
$\frac{T_{plate}}{\partial \bar{r}}\Big\vert_{(\bar{r}=I_{plate\vert die})}$	$= \frac{T_{die}}{\partial \bar{x}}\Big\vert_{(\bar{r} \in [L_{die}-L_{port}-L_{plate}, L_{die}-L_{port}])}$

Table 12: Plate section boundary conditions.

Die	
$\frac{T_{die}}{\partial \bar{x}}\Big\vert_{(\bar{x}=0),(\bar{x}=L_{die})}$	$= 0$
$\frac{T_{die}}{\partial \bar{r}}\Big\vert_{(\bar{r}=R_{die})}$	$= 0$

Table 13: Die section boundary conditions.

4.1.1 Viscous Dissipation

The term representing the heat generation caused by viscous dissipation, $\tilde{\Phi}(T(\bar{\boldsymbol{\xi}}); \bar{v})$, is based on work by Aukrust and LaZghab (2000). The term consists of multiple nested expressions, starting with the pressure gradient along the axial dimension;

$$\frac{\partial P(\bar{\boldsymbol{\xi}}, \bar{v})}{\partial \bar{x}} = \left(\frac{\bar{v}(n_f + 1)}{\sqrt{3} A_f R^{(n_f+1)} (1 - \frac{2}{n_f+3})} \right)^{\frac{1}{n_f}} \left(- \frac{2}{\sqrt{3}\alpha_f} \right) \left(e^{\frac{Q_f}{R_f T(\bar{\boldsymbol{\xi}})}} \right)^{\frac{1}{n_f}}, \tag{53}$$

which is then used for the calculation of the velocity profile;

$$v(\bar{\boldsymbol{\xi}}, \bar{v}) = e^{\frac{-Q_f}{R_f T(\bar{\boldsymbol{\xi}})}} \left(\frac{\sqrt{3}}{2} \alpha_f \right)^{n_f} \left(\frac{\sqrt{3} A_f R^{n_f+1}}{(n_f + 1)} \right) \left(\left| \frac{\partial P(\bar{\boldsymbol{\xi}}, \bar{v})}{\partial \bar{x}} \right| \right)^{n_f} \left(1 - \frac{\bar{r}^{n_f+1}}{R^{n_f+1}} \right). \tag{54}$$

Its derivative with respect to the radial dimension is then used to find the dynamic viscosity as

$$\mu(\bar{\boldsymbol{\xi}}, \bar{v}) = \frac{\sqrt{3}}{3\alpha_f \left| \frac{\partial v(\bar{\boldsymbol{\xi}}, \bar{v})}{\partial \bar{r}} \right|} \left(\frac{1}{\sqrt{3} A_f} \left| \frac{\partial v(\bar{\boldsymbol{\xi}}, \bar{v})}{\partial \bar{r}} \right| e^{\frac{Q_f}{R_f T(\bar{\boldsymbol{\xi}})}} \right)^{\frac{1}{n_f}}, \tag{55}$$

and ultimately the viscous dissipation, $\Phi[W/m^3]$;

$$\Phi(\bar{\boldsymbol{\xi}}, \bar{v}) = -\mu(\bar{\boldsymbol{\xi}}, \bar{v}) \left(\frac{\partial v(\bar{\boldsymbol{\xi}}, \bar{v})}{\partial \bar{r}} \right)^2. \tag{56}$$

The contribution to temperature change is then found via the heat capacity and density of the material, as

$$\tilde{\Phi}(\bar{\boldsymbol{\xi}}, \bar{v}) = -\frac{\Phi(\bar{\boldsymbol{\xi}}, \bar{v})}{\rho c_p}. \tag{57}$$

Equation (57) applies to all aluminium sections, except for the peak section, as the viscous dissipation contribution to the rate of change in temperature, as described in (50). Note that the section name subscripts are omitted from the variables in the equations in Section 4.1.1, as they apply to multiple sections and to simplify notation.

Since the objective of the work in this thesis is to implement the model equations as constraints to a simultaneous dynamic optimization problem, it is greatly beneficial to reformulate the equations such that all expressions are continuously differentiable with respect to the optimization variables, which include the temperature state of the model, $T(\bar{\boldsymbol{\xi}}) > 0$, and the ram speed, $v_{ram} > 0$. In this thesis, we therefore modify the equations such that they do not include the absolute value operator, and reduce them to simplify implementation. First off, notice that (53) is always negative because $v_{ram} > 0 \implies \bar{v} > 0$ and all flow related variables (subscripted "f") are positive real numbers, thus we have

$$\left| \frac{\partial P(\bar{\boldsymbol{\xi}}, \bar{v})}{\partial \bar{x}} \right| = - \frac{\partial P(\bar{\boldsymbol{\xi}}, \bar{v})}{\partial \bar{x}}. \tag{58}$$

Using (58), one can insert (53) into (54) as

$$\begin{aligned} v(\bar{\boldsymbol{\xi}}, \bar{v}) = e^{\frac{-Q_f}{R_f T(\bar{\boldsymbol{\xi}})}} & \left(\frac{\sqrt{3}}{2} \alpha_f \right)^{n_f} \left(\frac{\sqrt{3} A_f R^{n_f+1}}{(n_f + 1)} \right) \\ & \cdot \left(\left(\frac{\bar{v}(n_f + 1)}{\sqrt{3} A_f R^{(n_f+1)} (1 - \frac{2}{n_f+3})} \right)^{\frac{1}{n_f}} \left(\frac{2}{\sqrt{3}\alpha_f} \right) \left(e^{\frac{Q_f}{R_f T(\bar{\boldsymbol{\xi}})}} \right)^{\frac{1}{n_f}} \right)^{n_f} \\ & \cdot \left(1 - \frac{\bar{r}^{n_f+1}}{R^{n_f+1}} \right) \end{aligned}$$

which after cancellations yields

$$v(\bar{\boldsymbol{\xi}}, \bar{v}) = \bar{v} \frac{(n_f + 3)}{(n_f + 1)} \left(1 - \frac{\bar{r}^{n_f+1}}{R^{n_f+1}} \right). \tag{59}$$

The gradient of the velocity profile along the radial dimension is therefore

$$\frac{\partial v(\bar{\boldsymbol{\xi}},\bar{v})}{\partial \bar{r}} = \bar{v}\frac{(n_f+3)}{(n_f+1)}\frac{\partial}{\partial \bar{r}}\left(1-\frac{\bar{r}^{n_f+1}}{R^{n_f+1}}\right) = -\bar{v}\bar{r}^{n_f}\frac{(n_f+3)}{R^{n_f+1}}, \tag{60}$$

which , since $\bar{v} > 0$ and $\bar{r} \geq 0$, is always negative or zero, thus

$$\left|\frac{\partial v(\bar{\boldsymbol{\xi}},\bar{v})}{\partial \bar{r}}\right| = -\frac{\partial v(\bar{\boldsymbol{\xi}},\bar{v})}{\partial \bar{r}}. \tag{61}$$

Using (61) in (55) yields

$$\begin{aligned}\mu(\bar{\boldsymbol{\xi}},\bar{v}) &= \frac{\sqrt{3}}{3\alpha_f\left(-\frac{\partial v(\bar{\boldsymbol{\xi}},\bar{v})}{\partial \bar{r}}\right)}\left(\frac{1}{\sqrt{3}A_f}\left(-\frac{\partial v(\bar{\boldsymbol{\xi}},\bar{v})}{\partial \bar{r}}\right)e^{\frac{Q_f}{R_f T(\bar{\boldsymbol{\xi}})}}\right)^{\frac{1}{n_f}} \\ &= \frac{1}{\sqrt{3}\alpha_f}\left(\frac{1}{\sqrt{3}A_f}\right)^{\frac{1}{n_f}} e^{\frac{Q_f}{n_f R_f T(\bar{\boldsymbol{\xi}})}}\left(-\frac{\partial v(\bar{\boldsymbol{\xi}},\bar{v})}{\partial \bar{r}}\right)^{\frac{1-n_f}{n_f}},\end{aligned} \tag{62}$$

which can be used together with (56) and (60) to get

$$\begin{aligned}\Phi(\bar{\boldsymbol{\xi}},\bar{v}) &= -\frac{1}{\sqrt{3}\alpha_f}\left(\frac{1}{\sqrt{3}A_f}\right)^{\frac{1}{n_f}} e^{\frac{Q_f}{n_f R_f T(\bar{\boldsymbol{\xi}})}}\left(-\frac{\partial v(\bar{\boldsymbol{\xi}},\bar{v})}{\partial \bar{r}}\right)^{\frac{1-n_f}{n_f}}\left(-\frac{\partial v(\bar{\boldsymbol{\xi}},\bar{v})}{\partial \bar{r}}\right)^{2} \\ &= -\bar{v}^{\frac{1+n_f}{n_f}} e^{\frac{Q_f}{n_f R_f T(\bar{\boldsymbol{\xi}})}}\left(\alpha_f\sqrt{3}A_f^{\frac{1}{n_f}}\right)^{-1}\left(\frac{\bar{r}^{n_f}}{\sqrt{3}}\frac{(n_f+3)}{R^{n_f+1}}\right)^{\frac{1+n_f}{n_f}}\end{aligned} \tag{63}$$

4.1.2 Viscous Dissipation for Peak Section

When modelling viscous dissipation for the peak section, we assume high strain rates. This assumption results in slightly modified equations for $\tilde{\Phi}_{peak}$. First we introduce the 'characteristic length', λ, which is found by solving the implicit equation

$$\left(\frac{2\lambda}{h_{peak}}\right)^2\left(e^{\frac{h_{peak}}{2\lambda}}\left(\frac{h_{peak}}{2\lambda}-1\right)+1\right) - \frac{2^{n_f+1}\bar{v}_{peak}e^{\frac{Q_f}{R_f T_{peak}(\bar{\boldsymbol{\xi}})}}}{\sqrt{3}A_f h_{peak}} = 0, \tag{64}$$

which assumes $v_{peak}(\bar{\boldsymbol{\xi}}) = 0$ for $\bar{r} = \frac{1}{2}h_{peak}$, meaning that there is no slipping between the flowing aluminium and the steel wall. Note that the peak section is not cylindrical, $\bar{r}$ is therefore used to mean the height inside the peak section, as opposed to the width. The maximum radial coordinate is then $\bar{r} = \frac{1}{2}h_{peak}$, at which point the aluminium flow speed is assumed to be zero. The characteristic length is then used to find the velocity gradient in $\bar{r}$ and the dynamic viscosity as

$$\frac{\partial v_{peak}(\bar{\boldsymbol{\xi}};\bar{v}_{peak})}{\partial \bar{r}} = \frac{\sqrt{3}A_f}{2^{n_f}}e^{\left(\frac{\bar{r}}{\lambda(\bar{\boldsymbol{\xi}};\bar{v}_{peak})}-\frac{Q_f}{R_f T_{peak}(\bar{\boldsymbol{\xi}})}\right)} \tag{65}$$

$$\mu_{peak}(\bar{\boldsymbol{\xi}};\bar{v}_{peak}) = \frac{1}{\sqrt{3}\alpha_f\left|\frac{\partial v_{peak}(\bar{\boldsymbol{\xi}};\bar{v}_{peak})}{\partial \bar{r}}\right|}\log_e\left(\frac{2^{n_f}}{\sqrt{3}A_f}\left|\frac{\partial v_{peak}(\bar{\boldsymbol{\xi}};\bar{v}_{peak})}{\partial \bar{r}}\right|e^{\frac{Q_f}{R_f T_{peak}(\bar{\boldsymbol{\xi}})}}\right). \tag{66}$$

Equations (56) and (57) are then used to find Φ_{peak} and $\tilde{\Phi}_{peak}$, as for the other aluminium sections. We call attention to the fact that, similarly to the simplifications for the other sections, the equations for viscous dissipation in the peak section can also be reduced significantly, and result in

$$\Phi_{peak}(\bar{\boldsymbol{\xi}};\bar{v}_{peak}) = -\frac{A_f}{n_f 2^{n_f}\alpha_f}\left(\frac{\bar{r}}{\lambda(\bar{\boldsymbol{\xi}};\bar{v}_{peak})}\right)e^{\left(\frac{\bar{r}}{\lambda(\bar{\boldsymbol{\xi}};\bar{v}_{peak})}-\frac{Q_f}{R_f T_{peak}(\bar{\boldsymbol{\xi}})}\right)}. \tag{67}$$

4.1.3 Reduction Work

Similarly to viscous dissipation, a reduction work is also modelled, which contributes to heat generation in the flowing material. The equations governing the reduction work are based on an internal report by Hydro.

To compute the reduction work that is generated when aluminium is flowing from section A to section B, one first needs to compute a 'reduction ratio', $R_{A|B}$, and a 'reduction speed', $v_{red,A|B}(\bar{v}_A)$, as

$$v_{red,A|B}(\bar{v}_A) = 2\bar{v}_A \frac{R_A^2}{R_A^2 + R_B^2} \tag{68}$$

$$R_{A|B} = \sqrt{\frac{R_A^2 + R_B^2}{2}}. \tag{69}$$

Further, one computes a reduction force, $F_{red,A|B}$, using

$$\sigma_{A|B}(\bar{\boldsymbol{\xi}}, \bar{v}_A) = \frac{2}{3\alpha_f}\left(\frac{(n_f+3)v_{red,A|B}(\bar{v}_A)e^{\frac{Q_f}{R_f T(\bar{\boldsymbol{\xi}})}}}{\sqrt{3}A_f R_{A|B}}\right)^{\frac{1}{n_f}} \tag{70}$$

$$F_{red,A|B}(\bar{\boldsymbol{\xi}}, \bar{v}_A) = \pi R_A^2\left(K_{1,A|B} + K_{2,A|B}\sigma_{A|B}(\bar{\boldsymbol{\xi}}, \bar{v}_A)\log_e\left(\frac{R_A^2}{R_B^2}\right)\right), \tag{71}$$

to finally get the reduction work (or 'power'), $\Psi_{A|B}(\bar{\boldsymbol{\xi}}, \bar{v}_A)[W]$, as

$$\Psi_{A|B}(\bar{\boldsymbol{\xi}}, \bar{v}_A) = F_{red,A|B}(\bar{\boldsymbol{\xi}}, \bar{v}_A)v_{red,A|B}(\bar{v}_A). \tag{72}$$

The contributions to the rate of change of temperature is then found via

$$\tilde{\Psi}_{A|B}(\bar{\boldsymbol{\xi}}, \bar{v}_A) = \frac{\Psi_{A|B}(\bar{\boldsymbol{\xi}}, \bar{v}_A)}{c_{p,al}\rho_{al}V_B}, \tag{73}$$

where V_B is the volume of the section downstream of the aluminium flow.

4.1.4 Reduction Work for Peak Section

The reduction work equations are also affected by the assumption of high strain rates. To calculate the reduction speed from the port section to the peak section, we use the following equations;

$$A_{port|peak} = \frac{\pi R_{billet}^2}{E_{ratio}} \tag{74}$$

$$v_{red,port|peak}(\bar{v}_{port}) = 2\bar{v}_{port}\frac{\pi R_{port}^2}{\pi R_{port}^2 + A_{port|peak}}, \tag{75}$$

where $A_{port|peak}$ is the reduction area from port to peak, which is equal to the product profile area. The reduction work is then found by

$$\sigma_{port|peak}(\bar{\boldsymbol{\xi}}; \bar{v}_{peak}; \bar{v}_{port}) = \frac{h_{peak}}{2\lambda(\bar{\boldsymbol{\xi}}; \bar{v}_{peak})\alpha_f n_f} \tag{76}$$

$$\begin{aligned} F_{red,port|peak}(\bar{\boldsymbol{\xi}}; \bar{v}_{peak}; \bar{v}_{port}) = \pi R_{port}^2\Bigg(&K_{1,port|peak} \\ &+ K_{2,port|peak}\sigma_{red,port|peak}(\bar{\boldsymbol{\xi}}; \bar{v}_{peak}; \bar{v}_{port}) \\ &\cdot \log_e\left(\frac{\pi R_{port}^2}{A_{port|peak}}\right)\Bigg), \end{aligned} \tag{77}$$

and using (72) and (73) to get $\Psi_{port|peak}$ and $\tilde{\Psi}_{port|peak}$. For the remainder of the thesis, for tidiness, we denote the peak reduction work contributions $\Psi_{feeder|port}$, $\tilde{\Psi}_{feeder|port}$, $\Psi_{port|peak}$ and $\tilde{\Psi}_{port|peak}$ as Ψ_{port}, $\tilde{\Psi}_{port}$, Ψ_{peak} and $\tilde{\Psi}_{peak}$ respectively.

4.1.5 Average Section Speeds

The average speed in each aluminium section is simply based on the ram speed and the cross sectional area of the section, and are derived from conservation of mass and the assumption of

incompressibility;

$$\begin{aligned}
\mathcal{Q}_{billet} &= \mathcal{Q}_{feeder} = \mathcal{Q}_{port} = \mathcal{Q}_{peak} \\
\bar{v}_{billet} A_{billet} \rho_{al} &= \bar{v}_{feeder} A_{feeder} \rho_{al} = \bar{v}_{port} A_{port} \rho_{al} = \bar{v}_{peak} A_{peak} \rho_{al} \\
\bar{v}_{billet} A_{billet} &= \bar{v}_{feeder} A_{feeder} = \bar{v}_{port} A_{port} = \bar{v}_{peak} A_{peak} \\
\bar{v}_{billet} \pi R_{billet}^2 &= \bar{v}_{feeder} \pi R_{feeder}^2 = \bar{v}_{port} \pi R_{port}^2 = \bar{v}_{peak} A_{peak},
\end{aligned} \tag{78}$$

where $\mathcal{Q}$, $\bar{v}$ and A represent the mass flow, average speed in the axial direction and the cross sectional area in each section. As described in Section 4.1.4, the cross sectional area of the peak section/product profile is calculated via the extrusion ratio, and is equal to the reduction area between the port and peak section found in (74). We have $A_{peak} = A_{port|peak}$. The average speed in the billet section is assumed to be equal to the ram speed;

$$\bar{v}_{billet} = v_{ram}. \tag{79}$$

The remaining average speeds are then calculated as

$$\bar{v}_{feeder} = v_{ram} \frac{R_{billet}^2}{R_{feeder}^2} \tag{80}$$

$$\bar{v}_{port} = v_{ram} \frac{R_{billet}^2}{R_{port}^2 N_{cavities}} \tag{81}$$

$$\bar{v}_{peak} = v_{ram} E_{ratio}. \tag{82}$$

4.2 Model Discretization

The implementation of the model into a simultaneous dynamic optimization problem requires that the model is discretized, or transcribed, in both space and time. That is, the dynamic system should be described by a finite dimensional state array evaluated at a set of time points within the horizon of the DOP. The process of discretizing in space turns the partial differential equations into a system of ordinary differential equations, which is explained further in Section 4.2.1. This Section describes the discretization of the system state into state arrays, and the discretization of the system state in time is described in Section 5.

To represent the the system state, $T(\bar{\boldsymbol{\xi}}; t)$, as an array $\boldsymbol{T}(\cdot)$, we choose a set of coordinates at which we evaluate the system state, and organize the temperatures at those coordinates in an array. The coordinates at which the temperatures are evaluated lie on a rectangular lattice within each section, and the temperatures are assumed to be the average temperatures within subdomains/control volumes of the section, called 'cells'. The division of sections into cells is depicted in Fig. 14, which shows a cut view of a billet divided into several rings. In the reduced cylindrical coordinates, these rings appear rectangular, as depicted in Fig. 15.

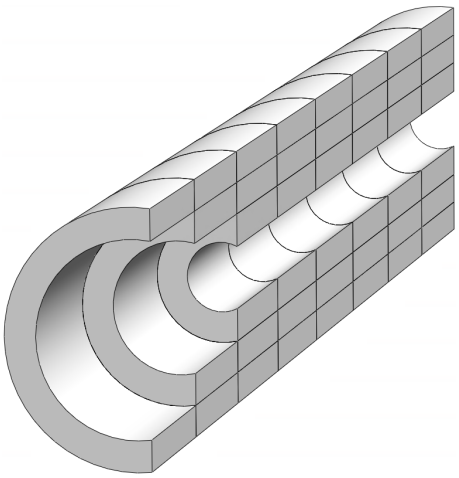


Figure 14: An illustration of the way the billet is divided into control volumes, referred to as 'cells'. The image shows a cut view of a subset of the billet cells, whereas the true discretizations fill the full volume of the different sections.

For the sake of simplicity and tidiness, we introduce the discrete coordinates; $\boldsymbol{\xi} = (r, x) \in \mathbb{Z}_{1,n_r} \times \mathbb{Z}_{1,n_x}$ for each section, which reference the cell numbering shown in Fig. 15. The values of n_r and n_x are determined by the relevant section, and the section-subscript is therefore omitted. The section state arrays can then be written as

$$\boldsymbol{T}_{section}(t) = \begin{bmatrix} T_{section}(1,1)(t) & \cdots & T_{section}(1,n_x)(t) \\ \vdots & & \vdots \\ T_{section}(n_r,1)(t) & \cdots & T_{section}(n_r,n_x)(t) \end{bmatrix} \in \mathbb{R}^{n_r \times n_x}. \tag{83}$$

The complete system state is then the collection of the section state arrays;

$$\boldsymbol{T}(t) = \begin{Bmatrix} \boldsymbol{T}_{billet}(t) \\ \boldsymbol{T}_{container}(t) \\ \boldsymbol{T}_{feeder}(t) \\ \boldsymbol{T}_{port}(t) \\ \boldsymbol{T}_{peak}(t) \\ \boldsymbol{T}_{plate}(t) \\ \boldsymbol{T}_{die}(t) \\ \boldsymbol{T}_{exit}(t) \end{Bmatrix}, \tag{84}$$

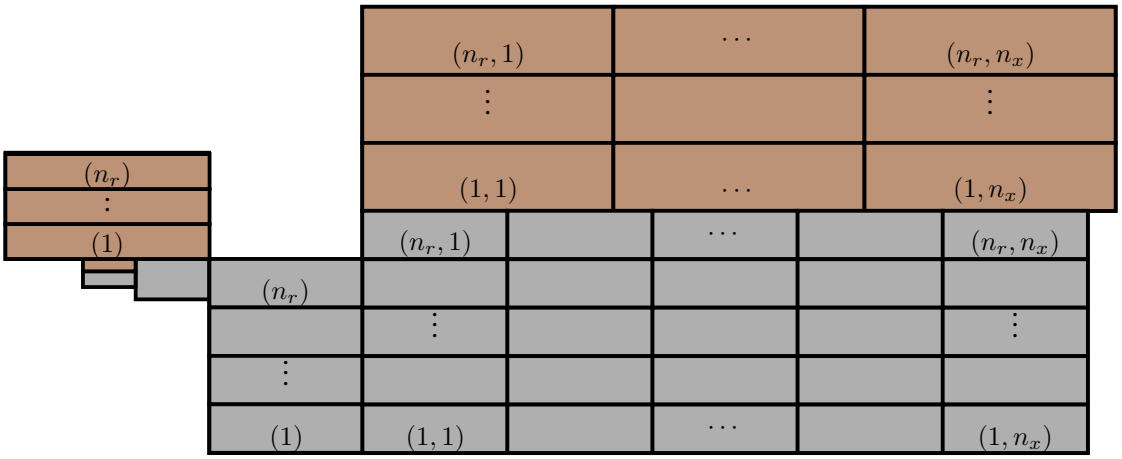


Figure 15: Depiction of discretization of sections into 'cells'. The coordinates represent the numbering of the cells, where n_r and n_x represent the number of cells in the radial and axial dimensions respectively in the relevant section section.

In this thesis, we choose to impose certain limitations on some of the discretization parameters, seen in Table 14, whereas the non-limited parameters are adjusted to optimize the trade-off between model accuracy and the solve-time of the resulting NLP, which is explored in Section 6.

Section	n_r	n_x
Billet	ND	ND
Container	ND	ND
Feeder	$\leq n_{r,billet}$	1
Port	1	1
Peak	1	1
Plate	1	1
Die	ND	1

Table 14: Resolution of section discretizations. ND reads "Not Determined", and means that the parameter is tuned/chosen for each dynamic optimization problem (DOP).

Additionally, we introduce the parameters $\Delta r_{section}$ and $\Delta x_{section}$, which are the thicknesses of all cells of the section indicated by the subscript in the radial and axial dimensions respectively, meaning that all cells have the same width in both directions. The thicknesses are found as

$$\Delta r_{billet} = \frac{R_{billet}}{n_r} \tag{85}$$

$$\Delta x_{billet} = \frac{L_{billet}}{n_x} \tag{86}$$

$$\Delta r_{container} = \frac{R_{container} - R_{billet}}{n_r} \tag{87}$$

$$\Delta x_{container} = \frac{L_{container}}{n_x} \tag{88}$$

$$\Delta r_{feeder} = \Delta r_{billet} \tag{89}$$

$$\Delta x_{feeder} = L_{feeder} \tag{90}$$

$$\Delta r_{port} = R_{port} \tag{91}$$

$$\Delta x_{port} = L_{port} \tag{92}$$

$$\Delta x_{plate} = L_{plate} = L_{peak} \tag{93}$$

$$\Delta r_{die} = \frac{R_{die} - R_{port}}{n_r} \tag{94}$$

$$\Delta x_{die} = L_{die}. \tag{95}$$

Note that the radius of the feeder section in the discrete model may not be equal to its radius in the continuous model due to the definition of the radial width of the feeder cells, which is defined in this way to simplify the discrete model dynamics. The centers, surface areas and volumes of the different cells, except peak and plate, are then found as

$$\bar{r}_r = (r - \frac{1}{2})\Delta r \tag{96}$$

$$\bar{x}_x = (x - \frac{1}{2})\Delta x \tag{97}$$

$$A_r(r) = 2\pi r \Delta r \Delta x \tag{98}$$

$$A_x(x) = 2\pi \Delta r^2 (r - \frac{1}{2}) \tag{99}$$

$$V(r) = A_x(r)\Delta x, \tag{100}$$

where $(\bar{r}_r, \bar{x}_x)$ is the center, and $A_r(r)$, $A_x(r)$ and $V(r)$ are the surface areas of the outer radial surface and axial surface and volume of cell (r, x) respectively. The areas and volume for peak and

plate are

$$A_{r,peak} = 2h_{peak} + 2w_{peak} \tag{101}$$
$$A_{x,peak} = A_{port|peak} \tag{102}$$
$$V_{peak} = A_{x,peak}\Delta x_{peak} \tag{103}$$
$$A_{r,plate} = 2\pi R_{plate}\Delta x_{plate} \tag{104}$$
$$A_{x,plate} = \pi R_{plate}^2 - A_{x,peak} \tag{105}$$
$$V_{plate} = A_{x,plate}\Delta x_{plate}. \tag{106}$$

4.2.1 Discretized Dynamic Equations

Upon discretizing the system state in space, the PDEs governing the system dynamics are also discretized in space, to obtain a system of ODEs that are only continuous in time. In order to remain close the the model implementation by Halås (unpublished), finite difference schemes(Strikwerda 2004) are used for this purpose;

$$\left.\frac{\partial T}{\partial \bar{r}}\right|_{(r,x)} \approx \frac{T(r+1,x) - T(r-1,x)}{2\Delta r} \tag{107}$$
$$\left.\frac{\partial T}{\partial \bar{x}}\right|_{(r,x)} \approx \frac{T(r,x+1) - T(r,x)}{\Delta x} \tag{108}$$
$$\left.\frac{\partial^2 T}{\partial \bar{r}^2}\right|_{(r,x)} \approx \frac{T(r+1,x) - 2T(r,x) + T(r-1,x)}{\Delta r^2} \tag{109}$$
$$\left.\frac{\partial^2 T}{\partial \bar{x}^2}\right|_{(r,x)} \approx \frac{T(r,x+1) - 2T(r,x) + T(r,x-1)}{\Delta x^2}, \tag{110}$$

where (107) is a central difference, (108) is a forward difference, and (109) and (110) are second-order central differences. The terms of the continuous model (50) and the boundary conditions presented in Tables 7-13 are rewritten terms of finite differences. The advection/translation term becomes

$$v(\xi;t)\frac{\partial T(\xi;t)}{\partial \bar{x}} \approx v(\xi;t)\frac{T(r,x+1;t) - T(r,x;t)}{\Delta x}. \tag{111}$$

The second order axial derivative from (50) is rewritten using (110) directly, and the second radial derivative becomes

$$\begin{aligned}\frac{1}{\bar{r}}\frac{\partial}{\partial \bar{r}}\left(\bar{r}\frac{\partial T(\xi;t)}{\partial \bar{r}}\right) &= \frac{1}{\bar{r}}\left(\frac{\bar{r}}{\partial \bar{r}}\frac{\partial T(\xi;t)}{\partial \bar{r}} + \bar{r}\frac{\partial^2 T(\xi;t)}{\partial \bar{r}^2}\right) \\ &= \frac{1}{\bar{r}}\frac{\partial T(\xi;t)}{\partial \bar{r}} + \frac{\partial^2 T(\xi;t)}{\partial \bar{r}^2} \\ &\approx \frac{T(r+1,x;t) - T(r-1,x;t)}{2\bar{r}\Delta r} + \frac{T(r+1,x) - 2T(r,x) + T(r-1,x)}{\Delta r^2}.\end{aligned} \tag{112}$$

On the boundary of a section, $\mathcal{B} = \mathcal{B}_r \cup \mathcal{B}_x$, $\mathcal{B}_r = \{(r,x)|r = 1 \vee r = n_r\}$, $\mathcal{B}_x = \{(r,x)|x = 1 \vee x = n_x\}$, the second-order central difference is not available as there is no adjacent cell beyond the boundary of that section. A potential solution to this is to use second-order forward and backward differences. Another solution is, if there is a cell beyond the boundary, but from another section, one may use this cell in the equation instead. The latter approach does not work in all cases, as one cannot generally say that the interfacing surface area, cell volumes, and even material are the same or equivalent. The latter approach is modified to yield

$$\begin{aligned}\left.\frac{\partial^2 T(\xi;t)}{\partial \bar{x}^2}\right|_{\xi\in\mathcal{B}_x} &\approx \frac{T(r,x_{adj};t) - 2T(r,x;t) + T_{adj}(t)}{\Delta x^2} \\ &\approx \frac{T(r,x_{adj};t) - T(r,x;t)}{\Delta x^2} + \frac{T_{adj}(t) - T(r,x;t)}{\Delta x^2} \\ &\approx \frac{A_r}{V_r}\frac{T(r,x_{adj};t) - T(r,x;t)}{\Delta x} + \frac{A_{(r,x)|adj}}{V_r}\frac{T_{adj}(t) - T(r,x;t)}{\Delta x}\end{aligned} \tag{113}$$

for the x-dimension, and similarly for the r-dimension. The subscript "adj" refers to the appropriate adjacent cell, where (r, x_{adj}) is the adjacent cell within the same section as (r, x), and T_{adj} is the temperature of the adjacent cell in the neighboring section. Equation (113) functions as the discretization of the boundary conditions between sections of similar material as well. If the boundary is between two section of different materials, A and B, the second term in (113) is replaced by

$$\frac{h_{A|B}A_{(r,x)|adj}}{\alpha_A V_r \rho_A c_{p,A}}\big(T_{adj}(t) - T(r,x;t)\big), \tag{114}$$

where A is the material of the the relevant section, and B is the material of the neighboring section. The one exception to this is the heat transfer between the billet and the container, which replaces the term with

$$\frac{T_{adj}(t) - T(r,x;t)}{\alpha_A V_r \rho_A c_{p,A}\Omega}, \tag{115}$$

where Ω is the thermal resistance between the billet and container, and is found by

$$\Omega = \frac{\log_e\left(\frac{2R_{billet}}{2R_{billet}+\Delta r_{billet}}\right)}{2\pi\Delta x_{billet}\kappa_{al}} + \frac{\log_e\left(\frac{2R_{billet}+\Delta r_{container}}{2R_{billet}}\right)}{2\pi\Delta x_{container}\kappa_{steel}} + \frac{1}{2\pi\Delta x_{billet}R_{billet}h_{al|steel}}. \tag{116}$$

The last exception is when there is no neighboring cell, such that no heat transfer is modelled, in which case the term is simply neglected.

4.2.2 Discretization Under 'Diminishing Billet' Conditions

A detail that has not been explored yet is that of the diminishing billet during extrusion. In fact, during extrusion, the billet undergoes translation, which is accounted for by (111) in the discretized dynamics. Although, this implies that the billet gradually disappears from the back end, which is not accounted for yet. The most important consideration regarding this phenomenon, is how one chooses to model the billet once it has been shifted by ΔL. Fig. 16 shows the three main contenders of how to discretize the billet after a shift of ΔL to the left. The top instance depicts the initial, non-shifted billet discretization, which has been considered up until this point. The three bottom instances depict three ways to discretize the billet after a shift, where the first uses a fixed discretization grid in overall system reference frame, where the back most cells, (r, n_x), shrink by ΔL. The second instance depicts a moving discretization grid which is fixed on the moving billet, and the front cells, $(r, 1)$, are shrunk by ΔL. The third and last instance depicts a grid where all cells shrink by the same amount, $\frac{\Delta L}{n_x}$, to maintain a homogeneous discretization grid.

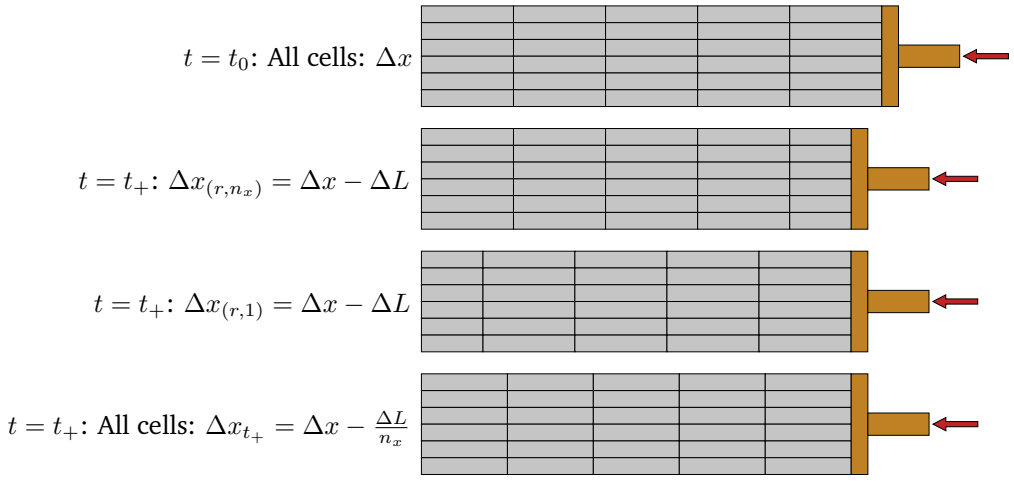


Figure 16: Illustration of three ways to adjust the spatial discretization of the billet upon it diminishing during extrusion. The figure shows four instances of the billet and its discretization, the top being the initial/non-shifted version, and the three bottom ones being the three options after a shift of ΔL.

The chosen method to handle the diminishing billet is the first option from Fig. 16, where the backmost cells, (r, n_x), are shrunk. To that end, we denote the length of the backmost cells Δx_{end}, and simply rewrite the equations from Sections 4.1.4, 4.1.1 and 4.2 that regard the diminished cells using Δx_{end} instead of Δx_{billet} where appropriate. Note that once $\Delta L > \Delta x_{billet}$, the backmost cells disappear entirely, and the cells to their left become the new backmost cells, and the discretization parameter n_x is updated as

$$n_x \leftarrow n_x - 1. \tag{117}$$

This is illustrated in Fig. 17, where the the billet and its discretization is shown for three different time instances, and the backmost cells disappear, or 'die'. The width of the backmost cells is then $\Delta x_{end} = \Delta x_{billet} - (\Delta L - \Delta x_{billet} N_{dead})$, where N_{dead} is the number of cells that have already died, and ΔL is the total amount of shift from the initial state of the system.

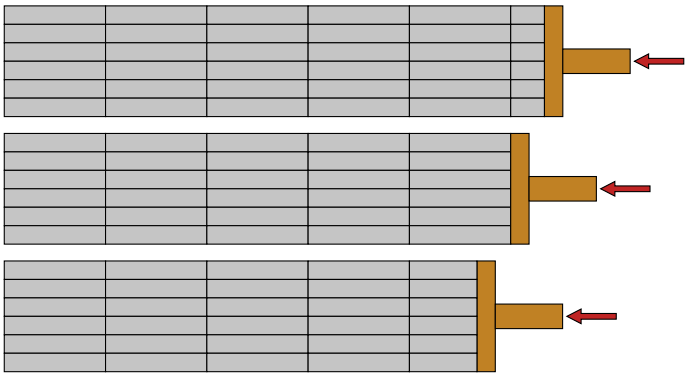


Figure 17: Illustration of the backmost cells disappearing, or 'dying', during the extrusion process.

Another consideration is how to modify the container interaction with the dead cells. In the real system, the container is interacting with air where the a cell has died, though the model assumes that no heat exchange happens at this interface after the cell dies.

4.2.3 Coolant Model

Lastly, the effect of coolant flow is modelled. The coolant used for the die cooling channels is liquid nitrogen (LN_2), and the control variable associated with its flow is the valve opening, $z_{valve} \in [0, 1]$, as a decimal fraction, where

$$\begin{aligned} z_{valve} = 0 &\rightarrow \text{valve fully closed} \\ z_{valve} = 1 &\rightarrow \text{valve fully open} \\ 0 < z_{valve} < 1 &\rightarrow \text{valve partially open.} \end{aligned}$$

The cooling channels appear in the model through their effect on the plate section, as a negative power contribution that is linear in coolant mass flow. The mass flow rate is computed from

$$\dot{m}_{LN2}(z_{valve}) = k_{max} z_{valve}^{s} \frac{\rho_{LN2}}{N_{cavities}}, \quad s > 0, \tag{118}$$

where k_{max} is the maximum volumetric flow, and ρ_{LN2} is the coolant mass density. The exponent $s\,[-]$ is a unit-less number that captures the nonlinear relationship between the valve opening and mass flow rate. The cooling capacity is then found via

$$Q_{LN2}(z_{valve}) = -\dot{m}_{LN2} \Delta h_{vap,LN2}, \tag{119}$$

and ultimately the contribution to the rate of change of temperature, Ξ, is found via

$$\Xi(z_{valve}) = \frac{\eta Q_{LN2}}{\rho_{steel} c_{p,steel} \Delta V_{plate}}. \tag{120}$$

5 Optimization

As the motivation behind this thesis is to gain understanding and experience with the use of direct collocation as a tool for optimization of complex and nonlinear industrial processes, this Section explores the formulation and implementation of an NLP that optimizes the extrusion process. The goal is to develop a reliable optimization formulation that finds the initial billet taper that allows a control scheme to extrude a given billet as fast as possible while satisfying any requirements on the temperature of the system. This algorithm will compliment a preexisting NMPC control scheme that optimizes the ram speed and coolant flow throughout the extrusion period, to minimize the extrusion time while maintaining the peak temperature within a set of specified requirements, seen in Fig. 3. The result should be an open loop optimization algorithm that facilitates the best possible initial taper for the extrusion phase, so as to allow the smallest possible extrusion time.

The Section starts with an overview of the general control problem, that is, what the goal of the extrusion cycle optimization layer is, in terms of an optimal control problem (OCP). Then, in Section 5.1.1, the problem is simplified by separating the overall OCP into two OCPs representing the heating- and extrusion phase, connected via the transit phase as an initial value problem (IVP). Under some assumptions, the separated problems are equivalent to the original problem, allowing one to solve a much simpler control problem, while still obtaining the optimal solution with respect to the overall control problem. Following is a Section that covers the conversion of the OCP into an NLP. Various considerations about the formulation of the equivalent NLP are presented in Section 5.2.2. The section ends with an overview of the implementation of the NLP in MATLAB, and some considerations thereof.

5.1 Optimal Control Problem

The objective of the proposed control scheme is to satisfy the following objectives;

i) maintain the desired peak temperature according to a set of specified requirements throughout the extrusion phase,

ii) and minimize the total extrusion period $\mathcal{T}_{ext}$.

The available control inputs are the ram speed and coolant-valve opening; $v_{ram}(t) \in [\underline{v}_{ram}, 5mm/s]$ and $z_{valve}(t) \in [0, 0.2]$. In addition, the initial taper is available as a control variable since this optimization takes places in the ECOL (see Fig. 3), which means that it occurs before the billet enters its extrusion cycle. The reference temperature for the heating phase is therefore free to be optimized. The overall control problem encompasses the heating phase, transit phase and extrusion phase, and can be written as;

$$
\begin{aligned}
&\min_{\boldsymbol{w}} \quad \mathcal{T}_{ext} && && \text{(121a)}\\
&\text{s.t.} \\
&T_{peak}(t) \in \mathcal{F}_{peak}(t) && \forall t \in [0, t_f], && \text{(121b)}\\
&\boldsymbol{T}(t) \in \langle 0, T_{max}] && \forall t \in [0, t_f], && \text{(121c)}\\
&v_{ram}(t) \in [\underline{v}_{ram}, 5mm/s] && \forall t \in [0, t_f], && \text{(121d)}\\
&|\dot{v}_{ram}(t)| \leq 0.4mm/s^2 && \forall t \in [0, t_f], && \text{(121e)}\\
&z_{valve}(t) \in [0, 0.2] && \forall t \in [0, t_f], && \text{(121f)}\\
&|\dot{z}_{valve}(t)| \leq \bar{z}_{rate} && \forall t \in [0, t_f], && \text{(121g)}\\
&\boldsymbol{V}_h(t) \in \mathcal{V} && \forall t \in [t_0, t_0 + t_h]. && \text{(121h)}
\end{aligned}
$$

where $\boldsymbol{w} = \{\boldsymbol{V}_h(t), \boldsymbol{T}_{init}, v_{ram}(t), z_{valve}(t)\}$, $\boldsymbol{V}_h(t)$ being the control variables in the heating phase, $\mathcal{V}$ is the domain of the heater control inputs, and $\boldsymbol{T}_{init} = \boldsymbol{T}_{billet}(0)$ being the initial taper. Note that (121c) is element wise, and applies to all relevant sections. Lastly, $\mathcal{F}_{peak}(t)$ represents

the feasible region for the peak control volume/cell at time t, and T_{max} is the maximum allowed temperature, which is typically a little lower than the melting point of the current aluminium alloy, as previously discussed.

5.1.1 Separation of Problem Formulation

Control of the complete extrusion cycle can be considered a *multistage problem*, as described by Diehl and Gros (2011), due to the 'multistage' nature of the heating-, transit-, and extrusion phase. Such systems are also reminiscent of *hybrid systems*, as described by Biegler (2010) and Borrelli (2003). Typically, multistage/hybrid systems require implementation of logic to model the various stages, such as in the Mixed Logic Dynamical (MLD) Systems described by Borrelli (2003). In the context of direct collocation, Biegler (2010) uses *switching profiles* and *guard/switching functions* to implement transitions between stages. In this thesis, however, we attempt to avoid dealing with hybrid systems entirely by separating the control problem, (31), into smaller control problems representing the different phases, while maintaining overall optimality. By separating the overall optimal control problem into smaller OCPs, no logic is required to distinguish between the three phases, and they can be implemented as standard dynamic NLPs.

To this end, we make use of the observation that, in practice, we have; $\mathcal{T}_{ext} \approx t_{RL} + t_f$, which can be seen in Fig. 9, where t_{RL} is the reload time and t_f is the extrusion time, as defined in Table 1. This is the case, as long as the extrusion phase is the 'bottle neck' of the extrusion cycle. That is, since the extrusion phase plus reload phase of different cycles cannot overlap, the next cycle must be delayed by at least $t_f + t_{RL}$. Furthermore, since there are 2 heaters in our case study, neither can the heating phases of three cycles overlap. Therefore, the extrusion period must also be delayed by at least $\frac{1}{2}t_h$. The extrusion period is therefore assumed to be expressed as

$$\mathcal{T}_{ext} \approx \max\left\{\frac{1}{2}t_h,\, t_{RL} + t_f\right\}. \tag{122}$$

Based on internal data from Hydro and the work by Gabrielsen (2022), the heating phase can be completed in $t_h \approx 300s$ for a non-optimized cycle, whereas the extrusion time is typically $t_f \approx 330s$. The reload time is estimated to be $t_{RL} \approx 23s$. Accounting for the parallel heater phases, the extrusion period is therefore $\mathcal{T}_{ext} \approx max\{150s, 353s\} = 353s$. To finish, we make an assumption that the heating time, t_h, is generally not heavily affected by the reference taper for the heater, such that the initial taper that minimizes t_f, does not result in a significant increase in heating time. We then have a reduction margin of $353s - 150s = 203s$ before the heating phase becomes the 'bottle neck', and the production rate cannot be increased further by decreasing the extrusion time. Fig. 18 illustrates this concept, where the extrusion time is shortened from what is depicted in Fig. 9, resulting in a shorter extrusion period and higher production rate. In the figure, the heating phases of two cycles overlap, which is permitted in this case study, whereas three simultaneous heating phases would not be possible and cause 'bottle necking'. It is also important to note that by optimizing the heating time, the reduction margin may be even greater. The current estimate of $t_h = 300s$ is based on the generic heater optimization by Gabrielsen (2022), and in practice, the non-optimized heating phase is currently in the range of $500s$ to $600s$.

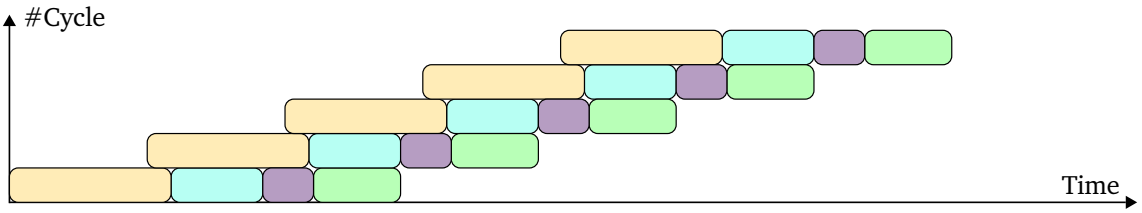
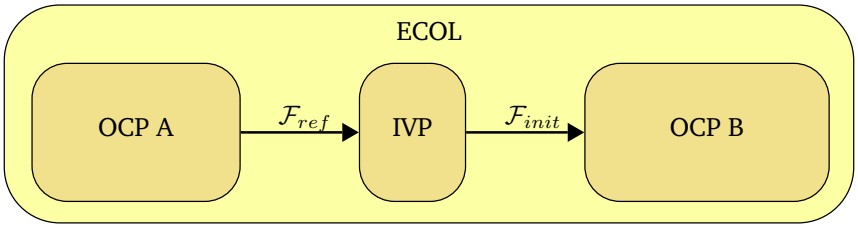
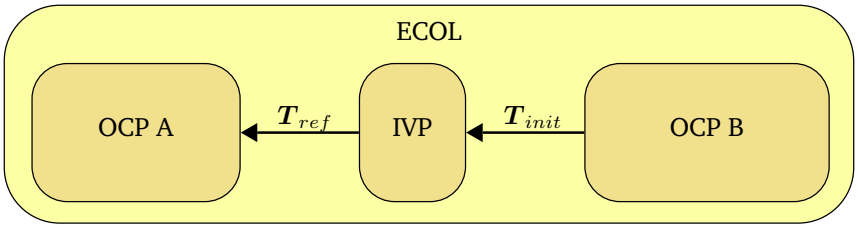


Figure 18: Phases: -Heater, -Transit, -Extrusion, -Reload Time.
Illustration of the extrusion cycles with small extrusion times. The figure showcases how the heating time must be very large in comparison to the extrusion time before it affects the extrusion period.

Under these assumptions, the optimal control problem can be separated into two parts; the heating phase (OCP A), $[t_0, t_0 + t_h]$, and the extrusion phase (OCP A), $[0, t_f]$. The transit phase, which here consumes the reload phase, is completely unactuated, and occurs as an intermediary IVP between the two parts. This approach yields two separate optimization problems, where the extrusion phase is optimized first with respect to extrusion time, and the heating phase is optimized second, with the primary goal of achieving the appropriate billet taper within the $t_h \leq 2(t_{RL} + t_f)$, to guarantee optimality. The separation is illustrated in Fig. 19. To ensure that the solution of OCP B is feasible with respect to OCP A, the feasible end-states of the heater must be transformed via the transit IVP to find the respective feasible initial tapers for OCP B, as illustrated in Fig. 19a.



(a) The feasible end-states of the heater, $\mathcal{F}_{ref}$, are fed into the transit IVP to compute the feasible set of initial tapers, $\mathcal{F}_{init}$.



(b) The solution of OCP B is fed to the transit phase IVP to compute the reference taper for the heating phase.

Figure 19: Illustrating the separation of the OCP into two parts, connected via the transit phase. The second part (B) encompasses the extrusion phase and is solved first, and the first part (A) encompasses the heating phase and is solved second, based on the solution of OCP B and the dynamics of the transit phase (the IVP).

An important observation is that after OCP B has been solved to find an initial taper, one can compute the respective heating time, and verify that that the heating phase has not become the bottle neck; $\frac{1}{2}t_h \not> t_{RL} + t_f$, thus preserving optimality. As both single shooting and collocation techniques have been extensively tested for the heater, by Gabrielsen (2022), and the transit IVP

is a trivial extension of that implementation, the remaining part of interest is that of the extrusion phase. To reduce the workload, this thesis will not cover the implementation of the OCP A, nor the transit IVP, and rather focus on the core part of the ECOL, which is OCP B, the extrusion phase. The OCP B problem is

$$\begin{aligned}
&\min_{\boldsymbol{T}_{init},\, v_{ram}(t),\, z_{valve}(t)} && t_f && && (123a)\\
&\text{s.t.} && && && \\
&\boldsymbol{T}_{init}(t) && \in \mathcal{F}_{init}(t) && \forall t \in [0, t_f], && (123b)\\
&T_{peak}(t) && \in \mathcal{F}_{peak}(t) && \forall t \in [0, t_f], && (123c)\\
&\boldsymbol{T}(t) && \in \langle 0, T_{max}] && \forall t \in [0, t_f], && (123d)\\
&v_{ram}(t) && \in [\underline{v}_{ram}, 5mm/s] && \forall t \in [0, t_f], && (123e)\\
&|\dot{v}_{ram}(t)| && \leq 0.4mm/s^2 && \forall t \in [0, t_f], && (123f)\\
&z_{valve}(t) && \in [0, 0.2] && \forall t \in [0, t_f], && (123g)\\
&|\dot{z}_{valve}(t)| && \leq \bar{z}_{rate} && \forall t \in [0, t_f]. && (123h)
\end{aligned}$$

where $\boldsymbol{T}$ represents the temperature array of any section, $0 < \underline{v}_{ram} < 5mm/s$, and

$$\mathcal{F}_{peak}(t) = \begin{cases} [-273.15^oC, 610^oC], & t \in [0, \delta t\rangle \\ [600^oC, 610^oC], & t \in [\delta t, t_f] \end{cases} \tag{124}$$

for some $\delta t \in \langle 0, t_f \rangle$. Time δt amounts to zone I) from Fig. 7, and should be as small as possible to maximize the amount of high quality product yielded form the billet. It remains a design choice whether or not to implement measures to ensure that δt is small, and depending on these measures, one may introduce an implicit or explicit trade-off between making δt small and making t_f small. The choice has been made to not include a constraint nor cost associated with δt in OCP B, to leave this design choice open.

5.2 Nonlinear Programming

As stated in Section 1.1, part of the motivation behind the thesis is to gain understanding of the use of simultaneous approaches to formulating nonlinear programs (NLPs) to optimize complex dynamical industrial processes, with focus on direct collocation. In order to reformulate (123) as an NLP, or to *transcribe* the problem, one must first discretize the model in time;

$$\{\boldsymbol{T}(0), \boldsymbol{T}(t_1), \ldots, \boldsymbol{T}(t_N)\} \to \{\boldsymbol{T}_0, \boldsymbol{T}_{t_1}, \ldots, \boldsymbol{T}_{t_N}\} \tag{125}$$

$$\{v_{ram}(0), v_{ram}(t_1), \ldots, v_{ram}(t_{N-1})\} \to \{v_{ram,0}, v_{ram,t_1}, \ldots, v_{ram,t_{N-1}}\} \tag{126}$$

$$\{z_{valve}(0), z_{valve}(t_1), \ldots, z_{valve}(t_{N-1})\} \to \{z_{valve,0}, z_{valve,t_1}, \ldots, z_{valve,t_{N-1}}\}, \tag{127}$$

for all section temperature arrays $\boldsymbol{T}(t)$. The system trajectories can then be approximated by using using orthogonal collocation (Diehl and Gros 2011), as seen in Section 2.3, by implementing the constraints

$$\left.\begin{aligned}
\boldsymbol{p}'(\tau_1, V_k) - \Delta t_k \cdot \boldsymbol{f}_{k,1}(\boldsymbol{p}(\tau_1, V_k), \boldsymbol{u}_k) &= 0 \\
\boldsymbol{p}'(\tau_2, V_k) - \Delta t_k \cdot \boldsymbol{f}_{k,2}(\boldsymbol{p}(\tau_2, V_k), \boldsymbol{u}_k) &= 0 \\
\vdots & \\
\boldsymbol{p}'(\tau_d, V_k) - \Delta t_k \cdot \boldsymbol{f}_{k,d}(\boldsymbol{p}(\tau_d, V_k), \boldsymbol{u}_k) &= 0 \\
\boldsymbol{T}_{k+1} - \boldsymbol{p}(1, V_k) &= 0
\end{aligned}\right\} \quad \forall k \in \mathbb{Z}_{0,N-1}. \tag{128}$$

The variable $\boldsymbol{T}_{k+1}$ represents the relevant temperature variable at discretization point $k+1$. Note that $\boldsymbol{T}_k - \boldsymbol{v}_{k,0} = 0$ has been omitted, as addressed in Section 2.3.2, and that the discretization interval size Δt_k is not necessarily uniform across the horizon. As described in Section 4.2.1 and

Section 4.2.2, the model equations describing the temperature dynamics are dependent on the cell length Δx, which for the backmost billet cells, changes as the billet is being extruded, thus making the dynamics time varying. Therefore, $f_{k,j}$ is used to denote the discretized model dynamics at time $t_{k,j} = t_k + \tau_j \Delta t_k$. Furthermore, the dynamics of the boundary cells of a section are different from the the internal cells, implying a discontinuous change in the model as the backmost cells die, and the next set of cells become the new backmost cells. Implementing (128) is therefore not trivial, as the extrusion length at time $t_{k,i}$ is unknown prior to solving the optimization problem.

One approach to overcoming this obstacle may be to simply add a variable for the 'extrusion length', L, and impose

$$L(t) = \int_0^t v_{ram}(\tau)\, d\tau, \tag{129}$$

which allows one to define the dynamics as functions of the extrusion length. Upon discretizing the extrusion length variable;

$$\{L(0), L(t_1), \ldots, L(t_f)\} \to \{L_0, L_1, \ldots, L_f\}, \tag{130}$$

and imposing a constraint for each new variable, one increases the number of variables and constraints by the number of discretization points, which is undesirable. Compared to the size of the rest of the NLP, however, this increase in variables and constraints is minuscule. The greater issue with the approach is that the discontinuities in the dynamic functions cannot be smoothly described by the introduction of the extrusion length as a variable, thus logical statements or some type of approximations are required to implement the dynamic constraints. In the former case, the NLP is converted into a mixed integer nonlinear program (MINLP), which are generally much harder to solve. Instead, another solution, that circumvents these obstacles, is chosen for the work in this thesis, which is presented in Section 5.2.1.

5.2.1 Progressor Transformation of the Extrusion Model

To address the practical problem of implementing discontinuous, 'extrusion-length-varying' dynamic constraints, we first introduce the term *'progressor'*:

Definition. *Let the state, $\boldsymbol{x}_A(\gamma) \in \mathbb{R}^n$, of the dynamical system A be defined on the continuous set $\gamma \in [\gamma_0, \gamma_f] \subseteq \mathbb{R}$ by the Initial Value Problem*

$$\frac{d\boldsymbol{x}_A(\gamma)}{d\gamma} = \boldsymbol{f}_A(\boldsymbol{x}_A(\gamma), \boldsymbol{u}(\gamma)), \quad \boldsymbol{x}_A(\gamma_i) = \boldsymbol{x}_i, \tag{131}$$

where $\gamma_i \in [\gamma_0, \gamma_f]$, and $\boldsymbol{u}(\gamma) \in \mathbb{R}^m$ is a control trajectory. Then, any variable $\lambda(\boldsymbol{x}_A(\gamma), \boldsymbol{u}(\gamma)) \in [\lambda_0, \lambda_f] \subseteq \mathbb{R}$ is a progressor of A under control trajectory $\boldsymbol{u}(\gamma)$ iff there exists a continuously differentiable, bijective function $g : [\gamma_0, \gamma_f] \mapsto [\lambda_0, \lambda_f]$, such that

$$\lambda = g(\gamma). \tag{132}$$

It is natural to think of time as a 'progressor' of a dynamical system, and the extrusion model described in this thesis is, in fact, defined in terms of time. Time, $t \in [0, t_f]$, is therefore a progressor of the extrusion model. We then make the important observation that, by requirement, we have $v_{ram}(t) > 0 \ \forall t \in \langle 0, tf]$, making the extrusion length monotonically increasing in time on the domain of the extrusion model. This implies a bijective mapping between the extrusion length and time, which in turn means that the extrusion length is also a progressor of the extrusion model. Figure 20 illustrates this concept by depicting a ram position curve that maps the extrusion length at the model discontinuities to points in time as a continuously differentiable, bijective mapping. Note that upon discretizing the system, we utilize a piecewise constant ram speed, implying a continuous, but not differentiable position curve at the discretization points. This is, however, not a problem. The figure illuminates the fact that the points in time at which the model discontinuities occur are dependent on the ram position curve, which is uniquely defined by the ram velocity curve, by (129). Since part of the objective of the NLP implementation is to solve for ram velocity over the duration of the extrusion phase, one does not know the time points of the discontinuities prior

to solving the NLP itself. Therefore, one cannot simply implement the model constraints on a grid of discretization points in time.

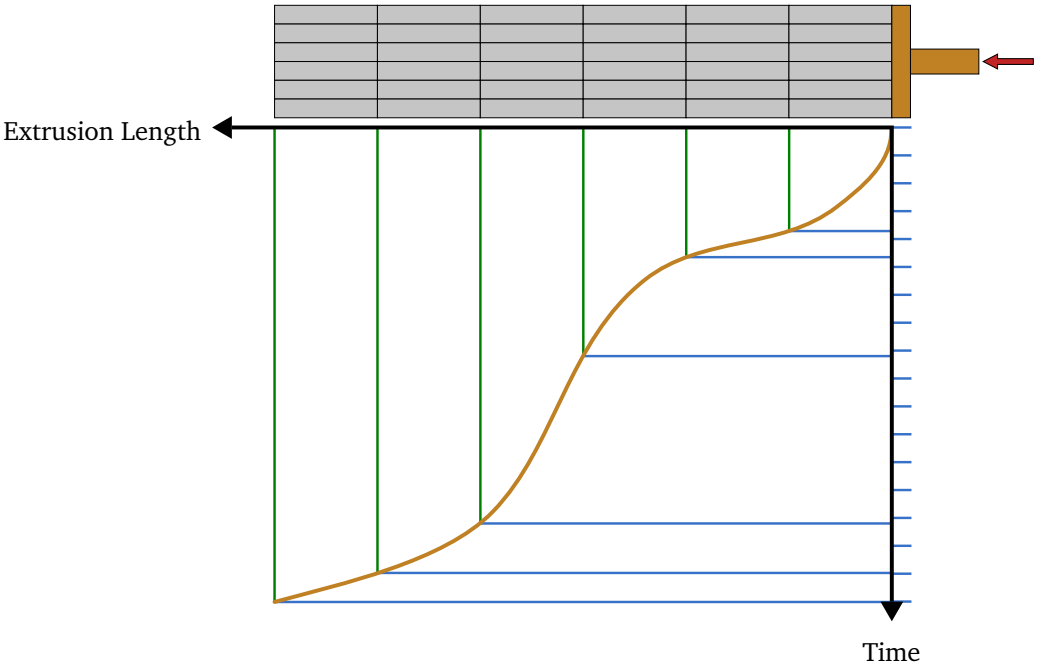


Figure 20: A conceptual sketch of the relationship between progress in time and progress in extrusion length, as given by the ram position curve (shown in orange). The figure illustrates how the times at which the discontinuities of the model occur are dependent on the ram velocity curve, and therefore unknown prior to solving for the optimal ram speed.

By transforming the extrusion model to be progressed by extrusion length rather than time, one can discretize the system in extrusion length instead, allowing one to align the discretization points with the discontinuities of the model. This concept is illustrated in Fig. 21, where a set of arrows indicate a uniform discretization grid of a billet that aligns with the axial cell intersections, which is where the model exhibits discontinuous changes. Such a transformation circumvents several issues, since;

- the model equations are known at the discretization points prior to knowing the ram speed curve, and are therefore implementable as constraints in an NLP,
- the discretization points may be aligned with the discontinuous model changes, effectively 'hiding' the discontinuities and avoiding logical statements in the NLP,
- and lastly, by knowing exactly what cells are active at each discretization point, one does not have to allocate variables for all cells at every discretization point, which reduces the size of the NLP significantly.

The latter point is discussed further in Section 7.1.2.

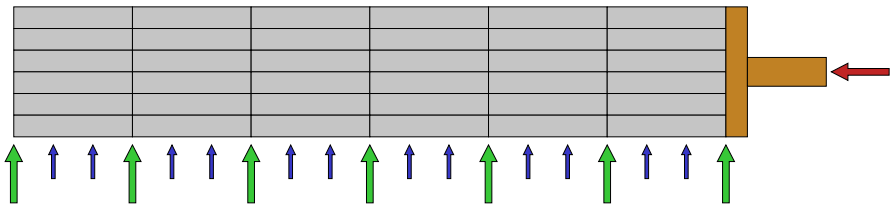


Figure 21: An illustration of a discretization grid in extrusion length, that aligns with all model discontinuities of the model. These discontinuities occur at the axial billet cell intersections, where the green arrows point. The blue arrows indicate additional discretization points. The red arrow indicates the direction of extrusion.

The mapping from time, t, to extrusion length, L, is described by (129). By applying the chain rule,

$$\frac{d\boldsymbol{x}}{dt} = \frac{dL}{dt}\frac{d\boldsymbol{x}}{dL}, \tag{133}$$

we get the extrusion-length-progressed model

$$\frac{d\boldsymbol{x}}{dL} = \frac{1}{v_{ram}(L)}\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, L), \tag{134}$$

which is known, and therefore implementable. The model constraints are then imposed as

$$\left.\begin{aligned} v_{ram,k} \cdot \boldsymbol{p}'(\tau_1, V_k) - \Delta L \cdot \boldsymbol{f}_{k,1}(\boldsymbol{p}(\tau_1, V_k), \boldsymbol{u}_k) &= 0 \\ v_{ram,k} \cdot \boldsymbol{p}'(\tau_2, V_k) - \Delta L \cdot \boldsymbol{f}_{k,2}(\boldsymbol{p}(\tau_2, V_k), \boldsymbol{u}_k) &= 0 \\ \vdots & \\ v_{ram,k} \cdot \boldsymbol{p}'(\tau_d, V_k) - \Delta L \cdot \boldsymbol{f}_{k,d}(\boldsymbol{p}(\tau_d, V_k), \boldsymbol{u}_k) &= 0 \\ \boldsymbol{q}_{k+1} - \boldsymbol{p}(1, V_k) &= 0 \end{aligned}\right\} \quad \forall k \in \mathbb{Z}_{0,N-1}, \tag{135}$$

where ΔL is the size of the discretization interval $[L_k, L_{k+1}]$, and the discretization of extrusion length is defined by

$$L_0 = 0 \tag{136}$$

$$L_N = L_{billet} \tag{137}$$

$$L_{k+1} = L_k + \Delta L \quad \forall k \in \mathbb{Z}_{0,N-1}. \tag{138}$$

Notice that, in this thesis, the spacial discretization of the billet is uniform, thus the interval size, ΔL, is constant.

5.2.2 Various NLP Considerations

This section covers various considerations regarding the constraints and objective in regard to reformulating (123) as an NLP.

Ram Acceleration and Valve Rate

The acceleration of the ram position and the rate of change of the valve opening must be approximated in order to constrain them in an NLP. The 'best' way to do this is not clear, and various approaches may have varying properties that are either desirable or undesirable. One possible approach is that of finite differences. Although, when the discretization grid is not uniform in time, it is not trivial to conclude on what finite difference should be used. A natural difference to consider is a 'time-center-to-center' difference, as illustrated by the violet graph in Fig. 22.

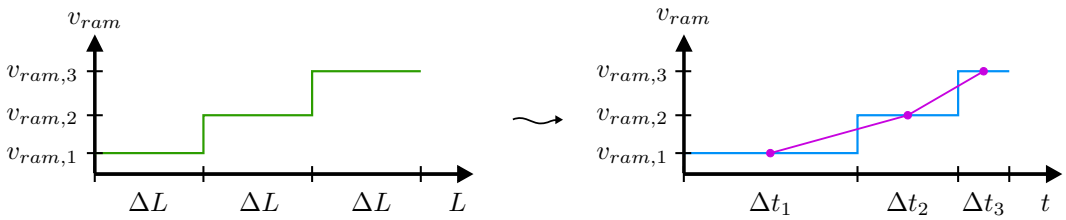


Figure 22: The figure depicts a piecewise constant ram speed curve over both extrusion length. L, and time, t. The piecewise constant speeds are of uniform length in extrusion length, both not in time, making a time-center-to-center time derivative approximation (violet) nonlinear in ram speed.

The ram acceleration can be approximated by the slope of the time-center-to-center curve as

$$\dot{v}_{ram}(t_{k+1}) \approx \frac{v_{ram,k+1} - v_{ram,k}}{\frac{1}{2}\left(\Delta t_k + \Delta t_{k+1}\right)}. \tag{139}$$

By rewriting

$$\Delta t_i = \frac{\Delta L}{v_{ram,i}}, \tag{140}$$

one gets

$$\dot{v}_{ram}(t_{k+1}) \approx \frac{v_{ram,k+1} - v_{ram,k}}{\frac{1}{2}\left(\frac{\Delta L}{v_{ram,k}} + \frac{\Delta L}{v_{ram,k+1}}\right)} = \frac{2v_{ram,k}v_{ram,k+1}}{\Delta L}\left(\frac{v_{ram,k+1} - v_{ram,k}}{v_{ram,k+1} + v_{ram,k}}\right). \tag{141}$$

As seen from (141), a property of the time-center-to-center acceleration approximation is its nonlinear dependence on the velocity itself, which is typically not desired. A much simpler, though somewhat naive, approach, is to use a space-center-to-center difference, resulting in

$$\dot{v}_{ram}(t_{k+1}) \approx \frac{v_{ram,k+1} - v_{ram,k}}{\Delta L}. \tag{142}$$

This approach is linear and thus more desirable in an NLP. However, the interpretation of (142) is less consistent with an average acceleration over several discretization points, and it deviates quickly from the time-center-to-center approach with large values of $v_{ram,k+1} - v_{ram,k}$. Similar methods can be considered for approximating the rate of change of the valve opening.

Of course, both of these approximations fall somewhat short, since the model assumes a piecewise constant ram speed and valve opening, and their derivatives are subsequently infinite at the discretization points, and zero elsewhere. The real extrusion press is of course unable to track such references, and will in practice have to operate at the practical maximum rate of change near the discretization points to accommodate the step changes, as is illustrated in Fig. 23. The regions in which the press accelerates, denoted $t_{ac.,k}$and $t_{ac.,k+1}$in the aforementioned figure, are regions where the model deviates further from the real system then in the regions of constant speeds. It is therefore desirable to minimize these regions, in order to maximize the model accuracy. As seen from Fig. 23, the sizes of these regions are proportional to the step size of the reference, thus constraining the sizes of these regions is more consistent with (142), although the bounds should then be reconsidered to represent a bound on the minimum acceptable region of acceleration. In addition, as mentioned in Section 3, in this thesis we assume that the real system is able to track the set points adequately, such that we may assume to be able to directly set the ram speed itself. It is also worth noting that at high ram speeds, the discretization interval size in time, Δt_k, is itself small, and the regions of acceleration should be smaller than the discretization interval, so as to not overlap;

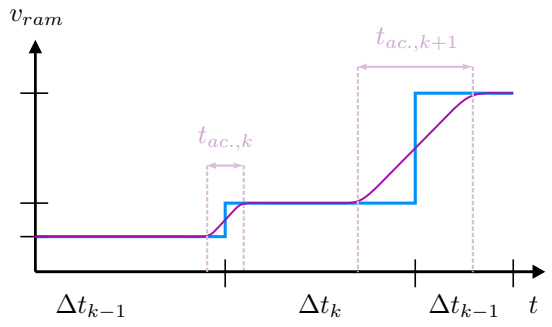


Figure 23: An illustration of how the piecewise constant ram speed reference will be tracked in practice by the hydraulic ram system. The blue graph represents the piecewise constant ram speed reference, and the purple graph represents the hydraulic ram system tracking the reference by availing maximum acceleration near the steps in the reference, denoted $t_{ac.,k}$and $t_{ac.,k+1}$.

$$\frac{1}{2}\left(t_{ac.k} + t_{ac.k+1}\right) \leq \Delta t_k \quad \forall k \in \mathbb{Z}_{1,N-2}. \tag{143}$$

If this is not satisfied, the system may fail to track the reference entirely, because it cannot reach the current set point before the next step. Of course, when using fine discretization, the piecewise constant reference may be filtered to produce a smooth reference representing an average. In this case, (143) does not apply, and other considerations must be made to ensure feasibility. However, the implementation in this thesis is expected to use a coarse discretization grid and minimal smoothing.

We denote the approximation of ram acceleration and valve opening rate of change by $\boldsymbol{d}_{ram}(\cdot)$ and $\boldsymbol{d}_{valve}(\cdot)$ respectively. The limitations of the rate of change of the valve opening is unknown for the case study considered in this thesis, and subsequently we assume that the valve can change infinitely fast and no constraint is imposed on it. Nevertheless, the bound $\boldsymbol{d}_{valve}(\cdot) \in [-\bar{z}_{rate}, \bar{z}_{rate}]$ is included in the problem formulation, to indicate that such a constraint is possible and indeed natural to consider. For the ram acceleration, a version of the simple linear approach (142) is used due to its simplicity. By assuming that the ram speed is consistently near some value $\tilde{v}$, such that $v_{ram,k} \approx v_{ram,k+1} \approx \tilde{v}$, one can approximate (141) by

$$\frac{2\tilde{v}\tilde{v}}{\Delta L(\tilde{v}+\tilde{v})}\left(v_{ram,k+1} - v_{ram,k}\right) = \frac{\tilde{v}}{\Delta L}\left(v_{ram,k+1} - v_{ram,k}\right) \tag{144}$$

to obtain a scaled version of (142), and impose the constraint

$$|v_{ram,k+1} - v_{ram,k}| \leq \frac{\Delta L}{\tilde{v}}\, 0.4mm/s^2. \tag{145}$$

It is clear from (145) that larger values of $\tilde{v}$ are more conservative. A value equal to the maximum ram speed is therefore chosen; $\tilde{v} = 5mm/s$.

Constraints on the Initial Taper

The initial temperature taper of the billet at the start of the extrusion phase ("initial taper") is limited by the capabilities of the heater and the transit time, and these limitations should be reflected by constraints in the final NLP formulation. The exact limitations themselves are not exactly known, as they rely on the region of feasible end-tapers of the heating phase, which has not been investigated. Nonetheless, under some assumptions, a set of simple constraints are implemented. First off, we assume that radial variations in the initial taper have limited effect on the optimal extrusion time, and subsequently all radial differences can be set to zero;

$$\boldsymbol{T}_{init}(r,x) = \boldsymbol{T}_{init}(r+1,x) \;\; \forall (r,x) \in \mathbb{Z}_{1,n_r-1} \times \mathbb{Z}_{1,n_x}, \tag{146}$$

where $\boldsymbol{T}_{init} = \boldsymbol{T}_{billet,0}$. This has the effect of regularizing the initial taper, which helps avoid a region of 'nearly equally good' solutions and may speed up the optimization algorithms somewhat. The initial taper is then reduced to axial variations in temperature. By requirement, the initial taper must possess an axial temperature taper greater than some minimum taper $\delta T^{-}_{init} > 0$, such that

$$\left.\frac{\partial T_{billet}(t=0)}{\partial \bar{x}}\right|_{(\bar{r},\bar{x})} \leq -\delta T^{-}_{init} \;\; \forall (\bar{r},\bar{x}) \in [0, R_{billet}] \times [0, L_{billet}]. \tag{147}$$

This is approximated in terms of the established billet cells as

$$\boldsymbol{T}_{init}(r,x) \geq \boldsymbol{T}_{init}(r,x+1) + \Delta T^{-}_{init} \;\; \forall (r,x) \in \mathbb{Z}_{1,n_r} \times \mathbb{Z}_{1,n_x-1}, \tag{148}$$

where $\Delta T^{-}_{init} = \delta T^{-}_{init} \Delta x_{billet}$. If another, more strict minimum taper is imposed, such as to accommodate the feasible heater end taper region, ΔT^{-}_{init} takes a new value, corresponding to the new minimum taper. The upper limit to the initial taper will be imposed by the heating phase, and we assume that such a constraint is sufficiently represented by

$$\boldsymbol{T}_{init}(r,x) \leq \boldsymbol{T}_{init}(r,x+1) + \Delta T^{+}_{init} \;\; \forall (r,x) \in \mathbb{Z}_{1,n_r} \times \mathbb{Z}_{1,n_x-1}, \tag{149}$$

where $\Delta T^{+}_{init} \geq \Delta T^{-}_{init}$ is some maximum taper dictated by the heating phase combined with the transit phase. Note that the constraints imposed by the heating/transit phase, (149), amounts to the part of $\mathcal{F}_{init}$ that is re-assessed with every extrusion cycle in the proposed extrusion cycle control scheme in Section 5.1.1.

Control Variables

The control variables $\boldsymbol{u}_k = [v_{ram,k},\, z_{valve,k}]^\top$ must be bounded according to the specifications in Section 3. For the ram speed, we impose

$$v_{ram,k} \in [\underline{v}_{ram}, 5mm/s] \;\; \forall k \in \mathbb{Z}_{0,N-1}, \tag{150}$$

where $\underline{v}_{ram} > 0$ is close to, but sufficiently above, zero. This is both by press requirement, and to ensure the existence of the Jacobian of the dynamic constraints, which is further discussed in Section 5.3. In this thesis, we choose

$$\underline{v}_{ram} = 0.125mm/s. \tag{151}$$

For the valve opening, $z_{valve,k}$, we perform a variable transformation;

$$\hat{z} \triangleq z^{s}_{valve}, \tag{152}$$

and rewrite (118) as

$$\dot{m}_{LN2}(\hat{z}) = k_{max} \hat{z} \frac{\rho_{LN2}}{N_{cavities}}, \tag{153}$$

to obtain a linear relationship between the valve opening variable, $\hat{z}$, and the cooling term, Ξ, in (120). Notice that on the domain $z_{vavle} \in [0,\, 0.2]$, given by Table 2, the the transformation (152) is bijective. The constraints are then also transformed as

$$\hat{z}_k \in [0, 0.2^s] \;\; \forall k \in \mathbb{Z}_{0,N-1}. \tag{154}$$

To recover the valve opening after solving the NLP, simply apply

$$z_{valve,k} = \sqrt[s]{\hat{z}_k} \quad \forall k \in \mathbb{Z}_{0,N-1}, \tag{155}$$

which is possible because of the bijective nature of the variable transformation on its domain.

Temperature Bounds

According to Table 4, all temperatures must stay below the melting temperature of aluminium $T_{melt} = 660^oC$. Of course, no temperature can be below $0K/-273.15^oC$, and the system should not be able to be driven below this limit, since it would not be consistent with the law of physics, and the same assumptions are made for the system model. Nevertheless, it may be useful to constraint the temperature variables from below, to avoid needless iterations in the regions of low temperatures. In fact, we assume that all temperatures will be above room temperature, and impose the element wise constraint

$$\boldsymbol{T}_{\zeta,k} > 20^oC \quad \forall k \in \mathbb{Z}_{0,N} \quad \forall \zeta \in \mathcal{S}, \tag{156}$$

where $\mathcal{S} = \{billet, container, feeder, port, peak, plate, die\}$.

In addition to the hard constraints on all temperatures, the peak temperature should lie within a certain region to maintain isothermal extrusion. Table 4 requires that the peak temperature is maintained within the interval $[\underline{T}_{peak}, \overline{T}_{peak}]$, where $\underline{T}_{peak} = 600^oC$ and $\overline{T}_{peak} = 610^oC$. The exception is that there must at least be a time period $[0, \delta t]$ where the peak temperature rises from its initial value and into the isothermal region. In this thesis, soft constraints are chosen, in order to accommodate initial deviations, yet motivate isothermal extrusion when possible. Using soft constraints rather than choosing some time δt, after which we impose hard constraints, has the benefit of not having to make assumptions about the rise time of the peak temperature. This is a major advantage, since one does not have to re-assess δt for every extrusion cycle, after receiving information about the incoming billet. Of course, choosing δt too small would render the NLP infeasible, and choosing δt too large would allow non-isothermal extrusion for longer than necessary. The soft constraints are imposed as

$$T_{peak,k} \in [\underline{T}_{peak} - \epsilon_k, \overline{T}_{peak} + \epsilon_k], \epsilon_k \geq 0 \quad \forall k \in \mathbb{Z}_{1,N}, \tag{157}$$

where $\epsilon_k \quad \forall k \in \mathbb{Z}_{1,N}$ are decision variables that are optimized simultaneously to the rest of the NLP. Under (157), we have isothermal extrusion whenever ϵ remains equal to zero, and ϵ increases linearly with deviations from the isothermal region. We therefore add

$$J_\epsilon = \gamma_\epsilon \sum_{k=1}^{N} \epsilon_k \tag{158}$$

to the objective in order to motivate isothermal extrusion, where $\gamma_\epsilon >> 0$ is some large coefficient.

Port and Peak in Steady State

The equations governing the dynamic behavior of the port and peak sections of the model are

$$\begin{aligned}\frac{dT_{port,k}}{dt} &= \bar{v}_{port,k}\frac{\boldsymbol{T}_{feeder,k}(r_{extrusion}) - T_{port,k}}{L_{port}} \\ &+ h_{al|steel}A_{port|die}\frac{\boldsymbol{T}_{die,k}(1) - T_{port,k}}{\rho_{al}c_{p,al}\Delta V_{port}} \\ &+ \alpha_{al}A_{feeder|port}\frac{\boldsymbol{T}_{feeder,k}(r_{extrusion}) - T_{port,k}}{\Delta V_{port}L_{port}} \\ &+ \alpha_{al}A_{port|peak}\frac{T_{peak,k} - T_{port,k}}{\Delta V_{port}L_{port}} \\ &+ \tilde{\Phi}_{port,k} + \tilde{\Psi}_{port,k}\end{aligned} \tag{159}$$

$$\begin{aligned}\frac{dT_{peak,k}}{dt} &= \bar{v}_{peak,k}\frac{T_{port,k} - T_{peak,k}}{L_{peak}} \\ &+ h_{al|steel}A_{peak|plate}\frac{T_{plate,k} - T_{peak,k}}{\rho_{al}c_{p,al}\Delta V_{peak}} \\ &+ \alpha_{al}A_{port|peak}\frac{T_{port,k} - T_{peak,k}}{\Delta V_{peak}L_{peak}} \\ &+ \tilde{\Phi}_{peak,k} + \tilde{\Psi}_{peak,k}\end{aligned} \tag{160}$$

for all $k \in \mathbb{Z}_{1,N}$, where $r_{extrusion}$ is the cell number that aligns with the port section, that is;

$$r_{extrusion} = \left\lceil \frac{R_{extrusion}}{n_{r,feeder}} \right\rceil, \tag{161}$$

and $v_{port,k}$ and $v_{peak,k}$ are the average aluminium flow speeds in the port and peak sections respectively. Due to the small volumes and high flow rates, the terms $\tilde{\Phi}_{port,k}$, $\tilde{\Psi}_{port,k}$, $\tilde{\Phi}_{peak,k}$ and $\tilde{\Psi}_{peak,k}$ become very large, resulting in very large derivatives. The model therefore acts as a stiff system, which implicit integration schemes such as collocation are typically suited for. However, the large derivatives may result in poorly conditioned constraint Jacobians. By setting the left hand side of (159) and (160) to zero, we assume that the port and peak sections are always at steady state, and convert the model into a differential algebraic equation (DAE). The temperature variables T_{port} and T_{peak} are then regarded as algebraic variables. We define the right hand sides of (159) and (160) as $g_{port,k,i}(\cdot)$ and $g_{peak,k,i}(\cdot)$ respectively for the algebraic relationship between the variables at collocation point i in discretization interval k. The algebraic equation

$$\boldsymbol{g}_{k,i}(\cdot) = \begin{bmatrix} g_{port,k,i}(\cdot) \\ g_{peak,k,i}(\cdot) \end{bmatrix} = \boldsymbol{0} \tag{162}$$

is then imposed as a constraint on the NLP for all discretization points and collocation points except the initial discretization point $(k, i) = (0, 0)$.

Characteristic Length: λ

The characteristic length, λ, is required to find the both the viscous dissipation energy and reduction work, $\tilde{\Phi}_{peak}$ and $\tilde{\Psi}_{peak}$, of the peak section, and is defined by (64), which is an implicit equation. Solving lambda must therefore be done numerically via an iterative method. By defining a set of variables representing the characteristic length at the various discretization points, $\{\lambda_0, \lambda_1, \ldots, \lambda_N\}$, one can impose (64) as a constraint for every discretization point, and thereby solve numerically for lambda simultaneously to solving the optimization problem. One may define separate variables for the characteristic length at every collocation point (k, i), although in this thesis, the characteristic length is assumed constant over the discretization interval.

Before implementing such a scheme, two key modifications are made, in terms of how the characteristic length appears in the model equations. Firstly, solving (64) for T_{peak} yields

$$T_{peak} = \frac{Q_f}{R_f}\left(\log_e\left(\left(\frac{1}{C_\lambda(\bar{v}_{peak})}\right)\left(\frac{2\lambda}{h_{heak}}\right)^2\left(e^{\frac{h_{peak}}{2\lambda}}\left(\frac{h_{peak}}{2\lambda}-1\right)+1\right)\right)\right)^{-1}, \tag{163}$$

where

$$C_\lambda(\bar{v}_{peak}) = \frac{2^{n_f+1}}{\sqrt{3}h_{peak}A_f}\bar{v}_{peak}. \tag{164}$$

Then, inserting (163) into the equation for the viscous dissipation energy, (67), then simplifies the expression to

$$\Phi_{peak} = -\frac{\bar{v}_{peak}h_{heak}\bar{r}}{2n_f\alpha_f\sqrt{3}\lambda^3}\left(e^{\frac{h_{peak}}{2\lambda}}\left(\frac{h_{peak}}{2\lambda}-1\right)+1\right)^{-1}e^{\frac{\bar{r}}{\lambda}}, \tag{165}$$

which renders the viscous dissipation energy in the peak section completely independent of the peak temperature itself. Of course, the dependency is still implicitly present through the definition of the characteristic length. Furthermore, via (82) we see that the viscous dissipation energy is linear in the ram speed, v_{ram}. Note that the viscous dissipation energy is distributed across the height of the peak section, $\bar{r}$, though an average energy over several values of $\bar{r}$ is used. Accompanied by the fact that the peak reduction work is on the form

$$\Psi_{peak} = W_\psi\frac{1}{\lambda}\bar{v}_{peak}, \tag{166}$$

where $W_\psi > 0$ is a constant, we see that (160) comes out to be linear in T_{peak} and v_{ram}.

The second key modification is a variable transformation of the characteristic length. By applying the transformation

$$\hat{\lambda} \triangleq \frac{1}{\lambda}, \tag{167}$$

one may rewrite (165) and (166) as

$$\Phi_{peak} = -\frac{\bar{v}_{peak}h_{heak}\bar{r}}{2n_f\alpha_f\sqrt{3}}\hat{\lambda}^3\left(e^{\frac{\hat{\lambda}h_{peak}}{2}}\left(\frac{\hat{\lambda}h_{peak}}{2}-1\right)+1\right)^{-1}e^{\bar{r}\hat{\lambda}} \tag{168}$$

$$\Psi_{peak} = W_\psi\hat{\lambda}\bar{v}_{peak}. \tag{169}$$

After the transformation, the viscous dissipation energy, (168), is somewhat simplified due to the lack of reciprocal appearances of λ. The reduction work, (169), has become linear in $\hat{\lambda}$, which further simplifies the equations. The value of $\hat{\lambda}$ can then be found by substituting $\hat{\lambda} = \frac{1}{\lambda}$ in the definition of the characteristic length, (64);

$$\left(\frac{2}{\hat{\lambda}h_{peak}}\right)^2\left(e^{\frac{\hat{\lambda}h_{peak}}{2}}\left(\frac{\hat{\lambda}h_{peak}}{2}-1\right)+1\right)-\frac{2^{n_f+1}\bar{v}_{peak}e^{\frac{Q_f}{R_fT_{peak}(\bar{\xi})}}}{\sqrt{3}A_fh_{peak}} = 0. \tag{170}$$

Objective

The objective of OCP B is to minimize the extrusion time, t_f, and the objective of the NLP should reflect this. Because the ram speed is assumed piecewise constant, the extrusion time, t_f, is

$$t_f = \sum_{k=0}^{N-1}\Delta t_k = \sum_{k=0}^{N-1}\frac{\Delta L}{v_{ram,k}} = \Delta L\sum_{k=0}^{N-1}\frac{1}{v_{ram,k}}. \tag{171}$$

In addition to minimizing extrusion time, though not specified, it is natural to want to minimize the use of coolant. Since there is a 'band' of equally good temperatures for the peak section, (157), there is some 'wiggle room' for the coolant flow and initial taper, of equally well performing

solutions. For the purpose of regularizing the NLP, some weight is added to the valve opening variable, $\hat{z}$, in the cost;

$$J_{\hat{z}} = \gamma_{\hat{z}} \sum_{k=0}^{N-1} \hat{z}_k. \tag{172}$$

By minimizing the coolant used, the initial taper is likely to be of lower temperature as well, thus potentially saving cost on both coolant and preheating.

When combining (171), (172), and (158), we get the complete objective

$$J(\mathcal{V}, \mathcal{P}_\epsilon, \mathcal{Z}) = t_f + J_\epsilon + J_{\hat{z}} = \sum_{k=0}^{N-1} \frac{1}{v_{ram,k}} + \gamma_\epsilon \sum_{k=1}^{N} \epsilon_k + \gamma_{\hat{z}} \sum_{k=0}^{N-1} \hat{z}_k, \tag{173}$$

where $\mathcal{V} = \{v_{ram,0}, v_{ram,1}, \ldots, v_{ram,N-1}\}$, $\mathcal{P}_\epsilon = \{\epsilon_1, \epsilon_2, \ldots, \epsilon_N\}$, $\mathcal{Z} = \{\hat{z}_0, \hat{z}_1, \ldots, \hat{z}_{N-1}\}$, and the constant ΔL has been omitted by rewriting $\gamma_{\epsilon,\hat{z}} \leftarrow \frac{\gamma_{\epsilon,\hat{z}}}{\Delta L}$.

5.2.3 Final NLP Formulation

The optimal control problem representing the extrusion phase, (123), has been reformulated as a nonlinear program via several assumptions, considerations, and simplifications presented earlier in this section. The final resulting NLP is summarized here:

$$\min_{\mathcal{W}} \quad \sum_{k=0}^{N-1} \frac{1}{v_{ram,k}} + \gamma_\epsilon \sum_{k=1}^{N} \epsilon_k + \gamma_{\hat{z}} \sum_{k=0}^{N-1} \hat{z}_k \tag{174a}$$

s.t.

$$v_{ram,k}\boldsymbol{p}'(\tau_i, V_{\zeta,k}) = \Delta L \boldsymbol{f}_{\zeta,k,i}(\boldsymbol{p}(\tau_i, V_{\zeta,k}), \boldsymbol{u}_k) \quad \forall i \in \mathbb{Z}_{0,d},\ \ \forall k \in \mathbb{Z}_{0,N-1},\ \ \forall \zeta \in \mathcal{S}^d, \tag{174b}$$

$$\boldsymbol{q}_{k+1} = \boldsymbol{p}(1, V_k) \qquad \forall k \in \mathbb{Z}_{0,N-1}, \tag{174c}$$

$$\boldsymbol{g}_{k,i} = \boldsymbol{0} \qquad \forall (k,i) \in \mathbb{Z}_{0,N} \times \mathbb{Z}_{0,d} \setminus (0,0), \tag{174d}$$

$$\boldsymbol{T}_{init}(r,x) = \boldsymbol{T}_{init}(r+1,x) \qquad \forall (r,x) \in \mathbb{Z}_{1,n_r-1} \times \mathbb{Z}_{1,n_x}, \tag{174e}$$

$$\boldsymbol{T}_{init}(r,x) \geq \boldsymbol{T}_{init}(r,x+1) + \Delta T^-_{init} \qquad \forall (r,x) \in \mathbb{Z}_{1,n_r} \times \mathbb{Z}_{1,n_x-1}, \tag{174f}$$

$$\boldsymbol{T}_{init}(r,x) \leq \boldsymbol{T}_{init}(r,x+1) + \Delta T^+_{init} \qquad \forall (r,x) \in \mathbb{Z}_{1,n_r} \times \mathbb{Z}_{1,n_x-1}, \tag{174g}$$

$$T_{peak,k} \in [\underline{T}_{peak} - \epsilon_k, \overline{T}_{peak} + \epsilon_k] \qquad \forall k \in \mathbb{Z}_{1,N}, \tag{174h}$$

$$\epsilon_k \geq 0 \qquad \forall k \in \mathbb{Z}_{1,N}, \tag{174i}$$

$$\boldsymbol{T}_{\zeta,} \in [20^o C,\, T_{melt}] \qquad \forall k \in \mathbb{Z}_{0,N}\ \ \forall \zeta \in \mathcal{S}, \tag{174j}$$

$$v_{ram,k} \in [\underline{v}_{ram}, 5mm/s] \qquad \forall k \in \mathbb{Z}_{0,N-1}, \tag{174k}$$

$$\hat{z}_k \in [0,\, 0.2^s] \qquad \forall k \in \mathbb{Z}_{0,N-1}, \tag{174l}$$

$$\boldsymbol{d}_{ram}(\mathcal{V}) \in [-0.4,\, 0.4][mm/s^2], \tag{174m}$$

$$\boldsymbol{d}_{valve}(\mathcal{Z}) \in [-\bar{z}_{rate},\, \bar{z}_{rate}], \tag{174n}$$

where

$$\mathcal{W} = \mathcal{V} \cup \mathcal{Z} \cup \mathcal{D} \cup \mathcal{P} \tag{175a}$$

$$\mathcal{V} = \{v_{ram,0}, v_{ram,1}, \ldots, v_{ram,N-1}\} \tag{175b}$$

$$\mathcal{Z} = \{\hat{z}_0, \hat{z}_1, \ldots, \hat{z}_{N-1}\} \tag{175c}$$

$$\mathcal{D} = \bigcup_{sec \in \mathcal{S}} V_{sec} \tag{175d}$$

$$V_{sec} = \{V_{sec,0}, V_{sec,1}, \ldots, V_{sec,N}\} \tag{175e}$$

$$V_{sec,k} = \{\boldsymbol{T}_{sec,k}, \boldsymbol{v}_{sec,k,1}, \ldots, \boldsymbol{v}_{sec,k,d}\}\ \ \forall k \in \mathbb{Z}_{0,N-1} \tag{175f}$$

$$\mathcal{S} = \mathcal{S}^d \cup \mathcal{S}^a \tag{175g}$$

$$\mathcal{S}^d = \{billet,\, container,\, feeder,\, plate,\, die\} \tag{175h}$$

$$\mathcal{S}^a = \{port,\, peak\} \tag{175i}$$

$$\mathcal{P} = \mathcal{P}_\lambda \cup \mathcal{P}_\epsilon \tag{175j}$$

$$\mathcal{P}_\lambda = \{\lambda_0, \lambda_1, \ldots, \lambda_N\} \tag{175k}$$

$$\mathcal{P}_\epsilon = \{\epsilon_1, \epsilon_2, \ldots, \epsilon_N\} \tag{175l}$$

where $\boldsymbol{v}_{sec,k,i}$ is state collocation variable i in discretization interval k of the section denoted by the subscript in place of "sec". Additionally, all constraints are element wise.

5.3 Analysis on Differentiability

The existence of the Jacobian and Hessian of the constraint expressions and the objective function, used by the optimization algorithm, requires that the constraints and objective function are twice differentiable on the point of evaluation. Here we inspect the differentiability of the constraints and objective function with respect to the decision variables $\mathcal{W}$.

First off, we notice that all temperature difference terms are on the form

$$C\big(\boldsymbol{T}_a - \boldsymbol{T}_b\big), \tag{176}$$

where C is a positive constant, which is infinitely differentiable. Secondly, we recognize that the collocation polynomial (37) is also infinitely differentiable everywhere. The notable terms of the dynamic equations are the viscous dissipation energy terms, Φ, as computed in (56) and (67), and the reduction work terms, Ψ, as computed in (72). Note that these terms are proportional to the dynamic temperature contributions, $\tilde{\Phi}$ and $\tilde{\Psi}$, via (57) and (73) respectively. We write the terms, as deduced in Section 4.1.1, 4.1.3 and 5.2.2, on their general form;

$$\Phi = -(v_{ram})^{\frac{n+1}{n}} e^{\sigma T^{-1}} C_\phi \tag{177}$$

$$\Psi = -(v_{ram})^{\frac{n+1}{n}} e^{\sigma T^{-1}} C_\psi \tag{178}$$

$$\Phi_{peak} = -(v_{ram})^{\frac{n+1}{n}} \hat{C}_\phi \hat{\lambda}^3 \Big(e^{\hat{\lambda}h}\Big(\hat{\lambda}h - 1\Big) + 1\Big)^{-1} e^{\bar{r}\hat{\lambda}} \tag{179}$$

$$\Psi_{peak} = \hat{C}_\psi \hat{\lambda} v_{ram}, \tag{180}$$

where C_ϕ, C_ψ, $\hat{C}_\phi$, $\hat{C}_\psi$, n, σ, and $\bar{r}$ are positive constants. Note that T represents a different cell temperature depending on the specific instance of the energy terms. By differentiation with respect to the present decision variables, one obtains

$$\frac{\partial \Phi}{\partial v_{ram}} = -\Big(\frac{n+1}{n}\Big)(v_{ram})^{\frac{1}{n}} e^{\sigma T^{-1}} C_\phi \tag{181}$$

$$\frac{\partial \Phi}{\partial T} = (v_{ram})^{\frac{n+1}{n}} \frac{\sigma}{T^2} e^{\sigma T^{-1}} C_\phi \tag{182}$$

$$\frac{\partial^2 \Phi}{\partial v_{ram}^2} = -\Big(\frac{n+1}{n^2}\Big)(v_{ram})^{\frac{1}{n}-1} e^{\sigma T^{-1}} C_\phi \tag{183}$$

$$\frac{\partial^2 \Phi}{\partial T^2} = -(v_{ram})^{\frac{n+1}{n}} \frac{\sigma}{T^2}\big(\sigma + 2T\big) e^{\sigma T^{-1}} C_\phi \tag{184}$$

$$\frac{\partial^2 \Phi}{\partial v_{ram} \partial T} = \Big(\frac{n+1}{n}\Big)(v_{ram})^{\frac{1}{n}} \frac{\sigma}{T^2} e^{\sigma T^{-1}} C_\phi \tag{185}$$

for the derivatives of (177), and the derivatives of (178) have the same structure. It is clear that none of the equations above are defined for $T = 0$, because of the reciprocal appearance of T in all equations. Additionally, we have $n = n_f > 1$, thus $(v_{ram})^{\frac{1}{n}} = \sqrt[n]{v_{ram}}$ is not defined for $v_{ram} < 0$. Furthermore, $(v_{ram})^{\frac{1}{n}-1} = \frac{(v_{ram})^{\frac{1}{n}}}{v_{ram}}$ is not defined for $v_{ram} \leq 0$, and we subsequently require

$$T \neq 0 \tag{186}$$

$$v_{ram} > 0 \tag{187}$$

for the existence of the Jacobian and Hessian of the constraint equations. Differentiation of (179) and (180) with respect to both v_{ram} and T is rather involved because of the complex dependence on the characteristic length, and is not shown here. Also worth noting, is the effect of the coolant valve opening variable, z_{valve}, which after the transformation (152), is linearly dependent on $\hat{z}$ and all derivatives exist everywhere.

Both requirements (186) and (187) are satisfied on the feasible region of (174). Although, the optimization algorithm used, *IPOPT*, may try to evaluate the constraints outside the feasible region, despite being an "*interior* point"-method. In fact, when availing a simultaneous method, such as

direct collocation, the solution is usually not feasible before the algorithm has converged. Therefore, we cannot guarantee the existence of the Jacobian on the iteration points, and subsequently cannot guarantee convergence of the algorithm. However, by supplying reasonable initial guesses to the algorithm, convergence is not a major issue, and in the event of convergence-failure, minor adjustments to the initial guess may be made until convergence is successful.

Note that the existence of the Hessian may not be necessary, if one uses a Hessian approximation technique, such as BFGS.

5.4 Implementation

This section covers aspects of the implementation of (174) in code. The implementation is done in MATLAB (The MathWorks Inc. 2023), with CasADi (Andersson et al. 2019) for algorithmic differentiation, interfacing to IPOPT (Wächter and Biegler 2006). The main outline of the implementation is presented, along with the main part of the code that composes the NLP. By reading this section, the reader will be able to understand and recreate the core aspects of the implementation, to such an extent that they should be able to reproduce similar results to what is presented in Section 6. Note that some aspects of the extrusion -process and -model, particularly parameter values, are confidential, thus the exact results presented in this thesis are not reproducible. Nevertheless, the reader should be able avail the ideas, methods, and implementation techniques used in this thesis to approach and solve similar problems.

The code is structured such that a class *"ExtrusionProcess()"* contains all information about the extrusion process to be optimized. Such information includes geometric properties, material properties, initial temperatures, process constraints, and miscellaneous other properties. Furthermore, the class contains all information about the corresponding NLP that should be built in order to optimize the process. This information includes the discretization of both time/extrusion length and the various sections, the number of collocation points per discretization interval, what optimization algorithm and linear solvers to use, initial guesses, cost function weights, and more. The idea is that, based on the information received about the incoming billet and the current state of the process, the class constructor defines these properties appropriately, along with the desired NLP properties. Then, one of its methods, *"BuildNLP()"*, is called to build the corresponding NLP, followed by *"SolveNLP()"*, which solves the NLP. This is illustrated in Fig. 24, which depicts a flow chart of the defining, calling, and use of the *ExtrusionProcess* class. Listing 5, in Appendix A, shows an example code that defines various properties of the extrusion process and NLP specifications, then proceeds to instantiate an *ExtrusionProcess*, and build and solve the NLP.

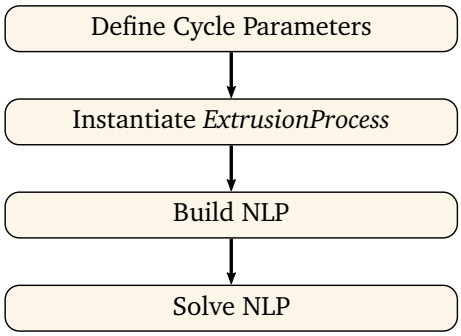


Figure 24: Flow chart depicting the process of defining cycle specific properties, and making and solving the corresponding NLP.

CasADi provides two main symbolic variable types, SX and MX variables. SX variables form scalar expressions that are computationally efficient, while MX variables form matrix expressions that are memory efficient. In this implementation, SX variables are used for defining the functions/mappings relating the states to their derivatives, and other mappings, MX variables are used for the optimization variables.

5.4.1 MATLAB Implementation

Here we display code that highlights the parts of the *BuildNLP()* method of the *ExtrusionProcess*-class that define important variables, constraints, and objective terms. The outline of the *BuildNLP()* method is summarized in Fig. 25, where the subroutines are presented in order, along with their respective Listings. The Listings that are not presented in this section can be found in Appendix A. Three of the Listings are run once, before a *for*-loop iterates over the discretization points and define the respective state, control and parameter variables and constraints for that discretization point and the respective collocation points. Lastly, the cost/objective is computed as an expression of the relevant decision/optimization variables.

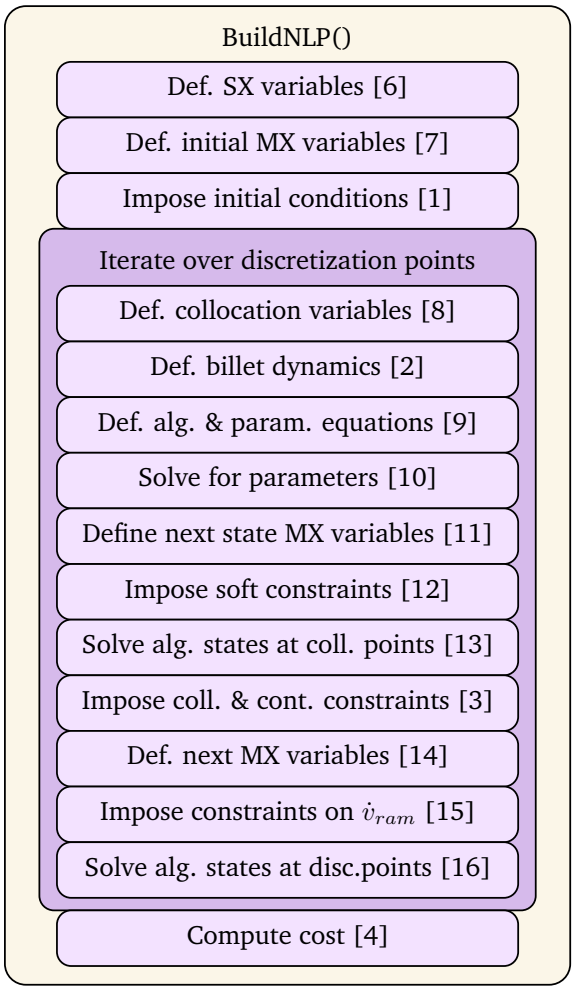


Figure 25: Structure of the BuildNLP routine of the *ExtrusionProcess*-class. The related Listings are given in square brackets.

In the presented code, the variable *NLP* represents the *ExtrusionProcess*-class instance, where *CLC* ("collocation") is a property "struct" thereof, containing various elements of the direct collocation NLP being built. The sub-struct *CLC.S* contains the symbolic variables and expressions defined in the *BuildNLP* routine, which is divided into, among others;

- *S.x*: state variables
- *S.u*: control variables
- *S.p*: parameter variables
- *S.coll*: collocation variables
- *S.w*: decision variables
- *S.g*: constraint expression
- *S.J*: cost expression.

Before being collected into *NLP.CLC.S.g*, constraint expressions are stored into either $g\{\}$ or $h\{\}$, for storing constraints on the form $g(\cdot) = 0$ and $lbg \leq h(\cdot) \leq ubg$, respectively. Geometrical properties are stored in the property struct *G* ("geometry"). Lastly, "parameters" refer to the characteristic length, λ, and the soft constraint parameter, ϵ.

Listing 6 defines the scalar-expression type variables that are used to form CasADi-Functions/maps that describe the various relationships such as system dynamics, implicit parameter equations, and cost/objective values. Similarly, Listing 7 defines the initial matrix-expression type variables that represent the system state, control and parameter values at discretization point 0. As the initial state of the extrusion phase is given, initial values are imposed via constraints on all sections except for the billet, whose initial temperatures is considered a control variable. Instead, the initial billet temperatures are constrained to be homogeneous in the radial dimension, and constrained to a minimum and maximum taper in the axial dimension. The initial value constraints are imposed as shown in Listing 1. This part of the code is central in the implementation, because this is where one implements the feasible region for the initial taper, which is an important consideration. The feasible region of the initial taper is dependent on the available heater time, and the transit dynamics which may vary from cycle to cycle due to, for example, temperature changes in the air, and the properties of the incoming billet. Therefore the constraints on the initial taper may need to be adjusted between cycles, which is done in this part of the code.

Listing 1: Imposing initial conditions, where $\Delta T_{init}^{-} \leq h_{billet} \leq \Delta T_{init}^{+}$

```
for name = ⟨loop over sections⟩
    if name ~= billet
        nr = NLP.G.nr.(name);               % Retrieve nr for section
        nx = NLP.G.nx.(name);               % Retrieve nx for section
        g{end+1} = T.(name) - ones(nr*nx,1)*NLP.initial_value.(name);
    else
        nr = NLP.G.nr.billet;               % Retrieve nr for billet
        nx = NLP.G.nx.billet;               % Retrieve nx for billet
        if ⟨using free initial taper⟩
            % Enforce homogenous radial temperature distribution
            for r = 1:nr-1
                for x = 1:nx
                    g{end+1} = T.(name)((x-1)*nr+r) - T.(name)((x-1)*nr+r+1);
                end
            end
            % Describe initial taper (top row of cells):
            for x = 1:nx-1
                h_billet{end+1} = T.(name)((x-1)*nr+1) - T.(name)(x*nr+1);
            end
        else ⟨using fixed linear initial taper⟩
            range = ⟨difference from back-end to front-end of billet⟩
            bottom = ⟨temperature of back-end of billet⟩
            g{end+1} = T.(name) - reshape(ones(nr,1)*(nx:-1:1)*range/nx + bottom,[],1);
        end
    end
end
```

Next, all variables and constraints are defined iteratively for the discretization points and collocation points throughout the optimization horizon, where the current discretization point is denoted k, and "next" is used about discretization point k+1. The loop starts by defining the collocation variables for the current discretization interval as MX variables, as shown in Listing 8. It is worth mentioning that if we have $n_{x,billet}(k+1) = n_{x,billet}(k) - 1$, the next billet section variable has $n_{r,billet}$ fewer elements than the current variable, thus the continuation constraint (174c) cannot be implemented for all elements of the current billet section variable. Nevertheless, the collocation equation, (174b), is implemented for all elements of the current billet variable, to account for all cells over the integration step. Therefore, the collocation variables corresponding to all cells of the current billet variable are defined, as can be seen in Listing 8.

Since the dynamics are dependent on the current billet length, and subsequently the current discretization point, the billet dynamics, and the dynamics of all other sections that depend on the billet state, are recalculated for every discretization point. This is shown in Listing 2. This part of the code is central because it is the core of the progressor transformation idea, and why it works. By transforming the extrusion model to be progressed by extrusion length rather than time, we were able to 'hide' the model discontinuities at the discretization points, and, importantly, predefine the dynamics on a discretization interval, since they are known in extrusion length. In this part of the code, we define the dynamics specific to each discretization interval, which allows the implementation of direct collocation.

Listing 2: Defining billet dynamics.

```
nr = NLP.G.nr.billet;
nx = ⟨retrieve remaining numer of axial cells in billet⟩;
if nx ~= 0
    T_SX.billet = SX.sym(['T_SX_billet_',k_str],nr*nx);
    ⟨construct Input_Cell{}⟩
    for name = ⟨loop over sections dependent on the billet section⟩
        dT_SX.(name) = NLP.dT.(name)(NLP,T_SX,C_SX,P_SX,k);
        dT.(name)    = Function(['dT_',char(name),'_',k_str],Input_Cell,{dT_SX.(name)});
    end
end
```

The next parts of the code deal with; defining the non-billet-dependent-section, algebraic, and parameter equations in terms of CasADi functions (Listing 9), creating the symbolic parameter expressions and adding them to the set of equality constraints g{} (Listing 10), defining the MX variables for all sections at the next discretization point, to be used in the collocation equation (Listing 11), Imposing soft constraints on the next peak temperature (Listing 12), and creating the algebraic expressions for the collocation points as functions of the state, control and parameter

variables at the respective collocation points (Listing 13).

The, perhaps, most central part of the implementation is the implementation of the collocation equation and continuation constraints, (174b) and (174c), which is shown in Listing 3. The code makes use of a set of predefined coefficients, *NLP.p.Li{}* and *NLP.p.dLi{}*, which correspond to Lagrange polynomials, to construct the collocation polynomials for the different cells of each section iteratively. The differential of the particular cell is found by mapping the current system state to the differential via the CasADi-function made in Listing 2 and Listing 9. In addition, the differential of the collocation polynomial, *dp*, is evaluated at each collocation point, by multiplying by an array corresponding to the exponents of the collocation points; *NLP.p.tau_i* $= [\tau_i^{d-1}, \ldots, \tau_i^0]$. The collocation equation is then imposed by setting their difference to zero, while scaling the state differential, *dT_temp* by the length of the discretization interval *NLP.nlp.dL* $= \Delta L$, and scaling the collocation polynomial differential by the ram speed *C.ram* $= v_{ram,k}$. Lastly, the continuation constraint is imposed by evaluating the collocation polynomial at $\tau = 1$, as *sum(p.(name))*, and setting it equal to the next state variable via an equality constraint.

Listing 3: Impose collocation and continuation constraints.

```
for name = ⟨loop over differential sections⟩
  nr      = NLP.G.nr.(name);
  nx      = ⟨nx of section at current disc. point⟩
  nx_next = ⟨nx of section at next disc. point⟩

% Loop over all cells of section:
  for rx = 1:nr*nx
          % Define collocation polynomial p, and its derivative dp:
           p.(name) = NLP.p.Li{ 1}.*T.(name)(rx);
          dp.(name) = NLP.p.dLi{1}.*T.(name)(rx);
          for d = 1:NLP.nlp.d
               p.(name) =  p.(name) + NLP.p.Li{ d+1}.*T_coll{d}.(name)(rx);
              dp.(name) = dp.(name) + NLP.p.dLi{d+1}.*T_coll{d}.(name)(rx);
          end
          % Define collocation equation:
          for d = 1:NLP.nlp.d
              ⟨construct Input_Cell{}⟩
              dT_temp = dT.(name)(Input_Cell{:});
              g{end+1} = (NLP.nlp.dL) .* dT_temp(rx) - dp.(name)*NLP.p.tau_{d}.*C.ram; % collocation equation
          end
          % Define continuation constraint:
          if rx <= nr*nx_next % Only if next variable has corresponding cell
              g{end+1} = sum(p.(name)) - T_next.(name)(rx);
          end
  end
end
```

The remaining part of the iterative loop, that loops over the discretization points, are concerned with; updating the state section variables; *T.(name)* $\leftarrow$ *T_next.(name)*, and defining the next control and parameter variables (Listing 14), imposing constraints on the step sizes in the ram speed (Listing 15), and creating and imposing constraints on the algebraic expressions for the current discretization point (Listing 16).

The last part of the implementation that is presented, is the construction of the cost/objective, which is shown in Listing 4. The code adds all the reciprocals of the ram speeds, the soft constraint parameters, ϵ_k, scaled with a large and somewhat arbitrary value of 100000, and the valve opening variables $\hat{z}$, with a small weight of 10.

Listing 4: Computing cost.

```
J_ram_SX   = NLP.cost.ram.*(1./C_SX.ram);
J_ram      = Function('J_ram',{C_SX.ram},{J_ram_SX});

NLP.CLC.S.J = MX.zeros;

%%% Extrusion Time:
for ram = [ NLP.CLC.S.u.ram{:}]
    NLP.CLC.S.J = NLP.CLC.S.J + J_ram(Cost_ram);
end

% Cost of violating peak soft constraint
for epsilon = NLP.CLC.S.p.epsilon
    NLP.CLC.S.J = NLP.CLC.S.J + epsilon{1}*100000;
end

% Regularizing cost on valve opening
for z_hat = NLP.CLC.S.u.valve
    NLP.CLC.S.J = NLP.CLC.S.J + z_hat{1}*10;
end
```

5.4.2 NLP Horizon

In practice, as mentioned in Section 3, there is a reload phase where a small remaining piece of the billet, called the "butt", is sheered of before the next billet is inserted into the container. This implies that the billet is not fully extruded, thus the horizon of the NLP does not need to be equal to the full length of the billet, and should in fact be slightly shorter. The length of the butt may be varying for different products, and the horizon should therefore be adjusted accordingly. In Section 5.2.3, we presented the final NLP formulation, (174), which assumed the horizon N. Although we until now have considered $N = L_{billet}/(n_{x,billet} n_{disc.})$, where $n_{disc.}$ is the number of discretization points per axial billet cell, in practice we choose to not optimize over all N discretization points. The implementation covered in this thesis optimizes over $\{0, 1, \ldots, N-1\}$.

5.4.3 Notable IPOPT options

When calling ipopt for solving an NLP, many options are available (Wächter, Vigerske et al. n.d.), to customize the algorithm. Through these options, one may tailor the solver to the specific NLP one is solving, to, for example, reduce the solve time/number of iterations needed or increase the accuracy of solution. Some of the most notable options that are used in the implementation in this thesis are listed in Table 15. Variables not listed take their default values.

Option	Value
nlp_scaling_method:	"gradient-based" [a]
nlp_scaling_max_gradient:	100 [a]
nlp_scaling_min_value:	10^{-8} [a]
print_level:	3 [b]
acceptable_tol:	10^{-6} [a]
linear_solver:	"mumps"
linear_system_scaling:	"none"
hessian_approximation:	"limited_memory"
limited_memory_update_type:	"bfgs" [a]

[a] default value
[b] The print level is relevant to the solve-time of the algorithm, but not the number of iterations used.

Table 15: IPOPT options.

5.4.4 Scaling of variables

By default, as seen from Table 15, ipopt scales the (augmented) system variables, such that the gradient is maximum 100. Normally, one should scale the variables in you system to be in the range $[0.01,\ 100]$, to avoid ill-conditioned linear algebra. The order of magnitude of the various variables in the system are shown in Table 16. They span from 10^{-4} to 10^{2}, which is somewhat of a wide spread, however, the solver will scale the system variables if necessary. Therefore, external scaling is only considered if convergence issues are experienced.

Variables	Order of Magnitude
$\boldsymbol{T}_{sec}$	10^{2}
v_{ram}	10^{-3}
$\hat{z}$	10^{-1}
ϵ	10^{2a}
λ	10^{-4}

[a] This value is typically zero on the solution, if isothermal extrusion is achieved, though may be larger during iterations.

Table 16: Variable Magnitudes.

5.4.5 Sparsity Patterns

Another important aspect to consider is the sparsity and sparsity patterns of the NLP. Some optimization algorithms exploit the sparsity of the constraints and objective Jacobians to speed up the computations, and uses linear solvers for which the sparsity is of varying importance. Therefore, the sparsity is important for the solve time of the NLP, and here, we inspect the sparsity of the direct collocation formulation, (174), built by the implementation covered in Section 5.4.1. We consider a typical instance of the NLP, defined as in Table 17. The sparsity pattern of the corresponding constraint Jacobian is shown in Fig. 26.

Property	Value
d	2
n_L	2
$(n_r, n_x)_{billet}$	(3,6)
$(n_r, n_x)_{container}$	(2,2)
$n_{r,die}$	6
coolant	yes
ram speed	free
initial taper	free

Table 17: Particular configuration for the NLP of which the constraint and objective Jacobians are inspected in Fig. 26 and Fig. 27.

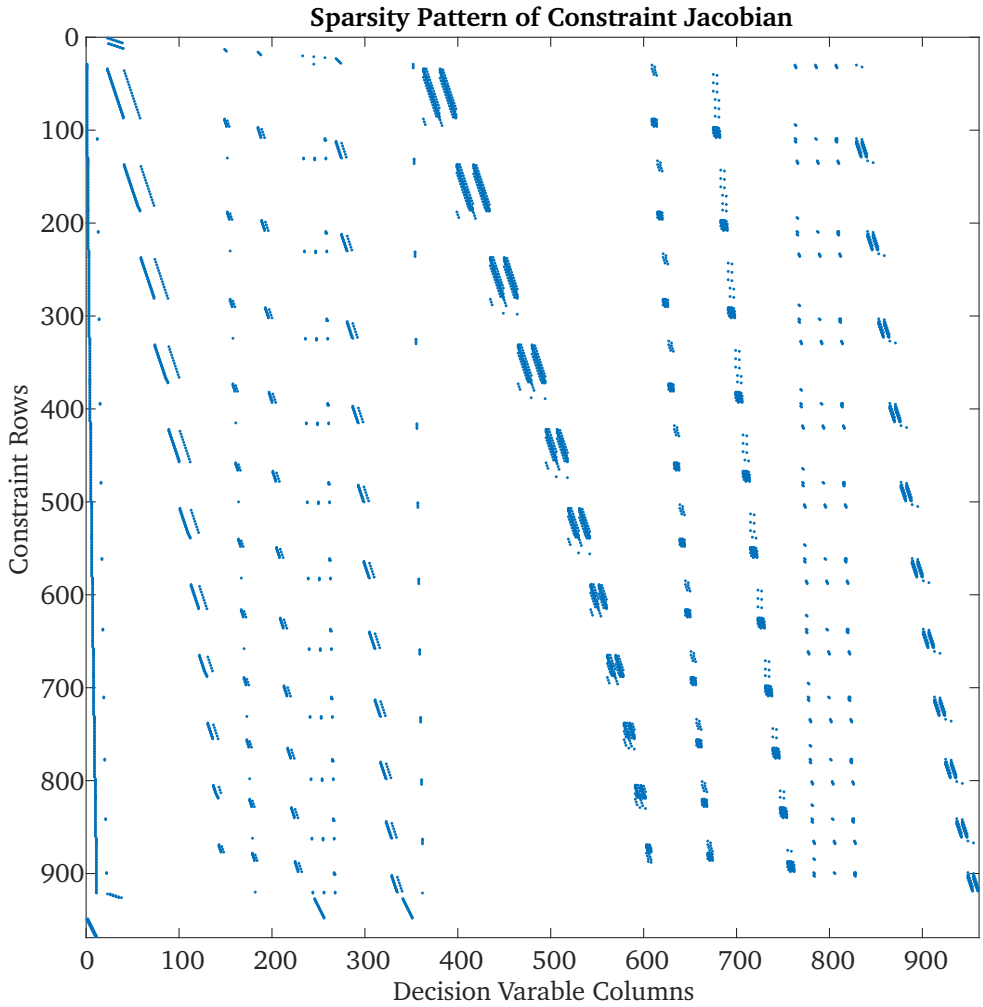


Figure 26: Sparsity pattern of the constraints Jacobian of the implementation of (174) with parameters as specified in Table 17. Blue dots indicate all elements with potentially non-zero values upon evaluation. Density: 0.0058023.

The number of constraints is for the particular NLP is 968, and the number of decision variables is 960, making the total number of elements equal to $968 \cdot 960 = 929280$. The number of non-zero elements is 5392, thus the density of the constraint Jacobian shown in Fig. 26 is $5392/929280 = 0.0058023$.

The Jacobian is not notably diagonal, which is typically desired, since linear solves are able to exploit the diagonal structure of matrices to solve linear systems efficiently. It is not clear whether or not rearranging the variables to yield a diagonal structure will yield significant reductions in the solve time, due to the preprocessing that occurs in the linear solvers. Conflicting opinions have been provided by professors that have been queried on this topic, therefore, to reduce the work load, rearranging the variables to obtain a diagonal structure of the constraint Jacobian is not prioritized.

For completeness, the Jacobian of the objective function with respect to the decision variables is also shown, seen in Fig. 27

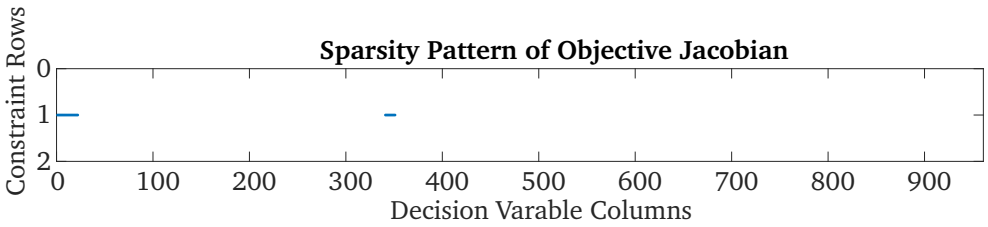


Figure 27: The sparsity of the respective objective Jacobian to the NLP defined in Table 17. The Jacobian consists only of a single row.

The objective Jacobian, however, is insignificant compared to the constraint jacobian for the type of NLPs solved in this thesis. We also emphasize that the sparsity of the constraint Hessian is also relevant. Although, we know that it has, at most, the density of the Jacobian.

6 Results

This Section presents the solutions to various versions of the optimal control problem (121), where an increasing degree of control freedom is optimized. Section 6.1 addresses the validity of the model, in terms of its specific applicability to the case study process of this thesis. Section 6.2 presents simulations, using typical control trajectories. The optimization is presented in Sections 6.3, 6.4, and 6.5, which cover the optimization of only the ram speed, the ram speed and coolant flow, and solutions to the complete optimal control problem (121) as implemented as (174) respectively.

All bounds on the initial taper are constant for the results presented in this Section, and take the values; $\Delta t_{init}^{-} = 10$ and $\Delta t_{init}^{+} = 20$. This means that the practical bounds on the initial taper change with the axial billet discretization. Of course, one may also scale the initial taper bounds, Δt_{init}^{-} and Δt_{init}^{+}, to accommodate the discretization, and achieve bounds that better reflect the same continuous tapers.

6.1 Model Tuning and Validation

The extrusion phase model that is implemented, as described in this thesis, is mainly based on modelling work by Halås (unpublished), and is meant to represent the case study extrusion process. Preferably, the model should be compared and validated against the case study extrusion press, however, such a validation is not done. Another viable option is to compare the model against the implementation by Halås (unpublished), which is tuned to coincide well with the case study process. No rigorous comparison is made between the two implementations, nevertheless, the model used in this thesis is tuned to match that of Halås by means of crude comparison between results. The tuning process resulted in the following scaling of the reduction work of the port section, and the reduction work and viscous dissipation terms of the peak section:

$$\Psi_{port} \leftarrow 0.135 \cdot \Psi_{port} \tag{188}$$

$$\Phi_{peak} \leftarrow 0.135 \cdot \Phi_{peak} \tag{189}$$

$$\Psi_{peak} \leftarrow 0.135 \cdot \Psi_{peak}. \tag{190}$$

This is a notable reduction of the terms, which should be understood. The exact cause of the deviations between the two models is unknown, although there are several known discrepancies between the models, that may contribute to the deviations. These discrepancies include the integration scheme that is used to integrate the models, the several types of discretization of the models (spatial discretization of each section, and progressor discretization), the omission of the exit temperature, and different means of computing the reduction and viscous dissipation energy terms for the port and peak section. These sources of model discrepancies are further discussed in Section 7.3. In consultation with Halås, M.Sc., based on their experience with the reference model and the case study extrusion process, the model used in this thesis is considered to be reasonable and to sufficiently represent the case study process, in order to claim relevance of the results presented in this thesis to the case study, and to the aluminium extrusion process in general.

6.2 Simulation Results

First, we fix all control inputs, to simulate a non-optimized extrusion process. These solutions form a basis on which to compare the model to existing data from extrusion processes. Furthermore, they highlight the significance of the optimization scheme, by providing contrasting non-optimized trajectories and respective extrusion times.

As is common in industry, the ram speed is kept constant for the simulations, and four different speeds are tested. Cooling is not included in the dynamics, and the initial taper is fixed and shown in Fig. 28. The results of the tests are presented in Fig. 29 and Table 18

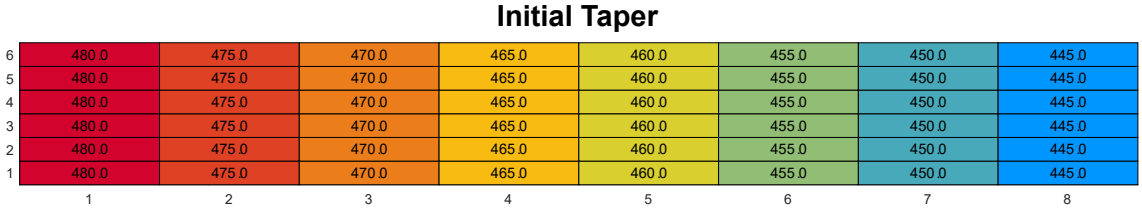


Figure 28: The fixed initial taper used for all instances where a fixed taper is used in this thesis. The temperature of a cell is displayed inside that cell in $[^oC]$. The cell-color is a visual aid, where the color represents the relative temperature compared to the rest of the billet, and highlight the temperature distribution inside the billet. The discrete radial and axial coordinates are displayed along the sides of the billet. The figure corresponds to the structure presented in Fig. 12, such that the extrusion happens from right to left.

Simulation:	**1**	**2**	**3**	**4**
Billet	$(6, 12)$	$(6, 12)$	$(6, 12)$	$(6, 12)$
Container	$(3, 3)$	$(3, 3)$	$(3, 3)$	$(3, 3)$
Die	6	6	6	6
d	2	2	2	2
n_L	3	3	3	3
N_L	16	16	16	16
n_w	2172	2172	2172	2172
n_g	2185	2185	2185	2185
n_{eq}	2127	2127	2127	2127
n_{in}	58	58	58	58
v_{ram}	0.0015	0.0025	0.0035	0.0045
t_f	562.5	337.5	241.1	187.5
t_{sol}	0.7	1.6	3.1	2.7
N_{iter}	11	24	49	41

Table 18: Simulation results, where Billet, Container, and Die are the number of cells in the respective sections, d is the number of collocation points, n_L is the number of discretization points per axial billet cell, thus $N_L = n_{x,billet} \cdot n_L$ is the number of discretization points along the progressor, n_w is the total number of decision variables, n_g is the total number of constraints, n_{eq} is the number of equality constraints, n_{in} is the number of inequality constraints, v_{ram} is given in m/s, t_f is the extrusion time in seconds, t_{sol} is the solve time in seconds, and N_{iter} is the number of iterations before convergence. Entries in black are defining properties, entries in green (■) are recessive properties, and entries in red (■) are the results.

The solve times and number of iterations used are highly affected by the initial guess given to the algorithm, which is further discussed in Section 7.5. The extrusion time of simulation 2 is interesting in that it represents a realistic extrusion time, since we notice that simulation 2 achieves isothermal extrusion at the desired peak temperature. The results, in terms of the temperature evolution shown in Fig. 29, is used to inspect the dynamical behavior of the system for various ram speeds, and aids in validating the model applicability. We see that, that the higher the ram speed, the higher the peak temperature. The radial temperature gradient in the billet is also greater for higher ram speeds, which corresponds with the intuition that more heat is generated towards the radial walls due to higher radial velocity gradients. The feeder also experiences greater radial temperature gradients, and also greater overall temperature, which corresponds with our intuition about the reduction work. Notably, all simulations seem to achieve isothermal extrusion at different temperatures, only with minor decrease in peak temperature throughout the extrusion phase.

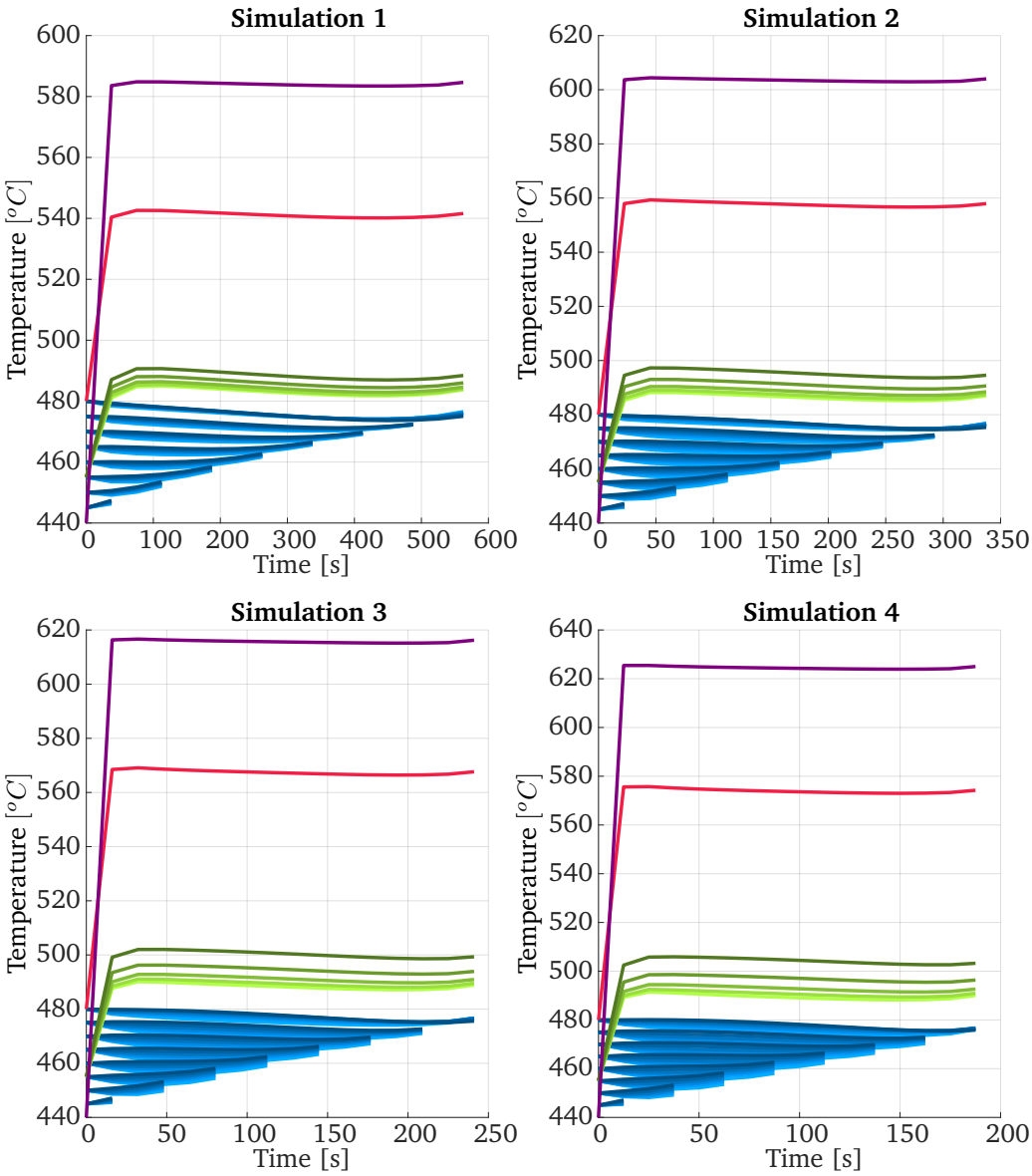


Figure 29: The simulation results of the tests defined in Table 18. Temperatures: ■-peak, ■-port, ■-feeder, ■-billet. Within a color-scheme, darker colors represent cells at higher radial coordinates. The shortest billet graphs represent the rightmost cells, that die first, and the longest graphs represent the leftmost cells, that die last.

6.3 Optimal Ram Speed

Here, we inspect solutions where the initial taper is fixed, and no coolant is used. These solutions therefore only optimize the ram speed, with respect to minimum extrusion time. These solutions provide a means of comparing the optimization scheme and its current implementation to other ram speed optimization schemes and solutions. Additionally, they further emphasize the difference between pure ram speed optimization and multi-control optimization that includes coolant and initial taper. The fixed initial taper shown in Fig. 28 is also used for the ram speed optimization NLPs. The problem is solved for 5 different discretizations, seen in Table 19, which also show the respective extrusion times, solve times, and iteration counts. The solutions are shown in Fig. 30, which shows the optimal ram speeds, and Fig. 31, which shows the respective temperature trajectories.

Ram-NLP:	**1**	**2**	**3**	**4**	**5**
Billet	$(3, 8)$	$(4, 8)$	$(5, 8)$	$(4, 10)$	$(4, 12)$
Container	$(2, 2)$	$(2, 2)$	$(2, 2)$	$(2, 2)$	$(2, 2)$
Die	6	6	6	6	6
d	2	2	2	2	2
n_L	2	2	2	2	2
N_L	16	16	16	20	24
n_w	1438	1698	1958	2374	3146
n_g	1451	1711	1971	2391	3167
n_{eq}	1393	1653	1913	2317	3077
n_{in}	58	58	58	74	90
t_f	285.4	296.2	309.2	307.1	316.4
t_{sol}	2.9	1.2	2.4	2.8	10.0
N_{iter}	98	31	38	38	89

Table 19: Results of the Ram Speed Optimization NLPs. Symbols and colors have the same meaning as in Table 18.

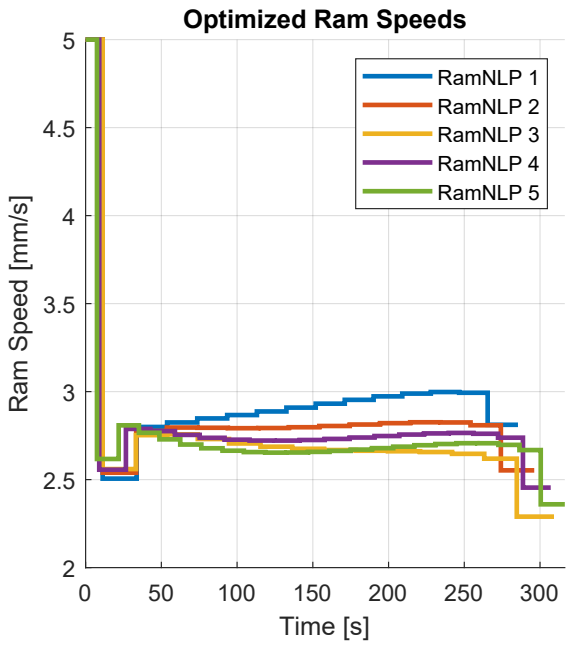


Figure 30: The ram speed trajectory results of the tests defined in Table 19.

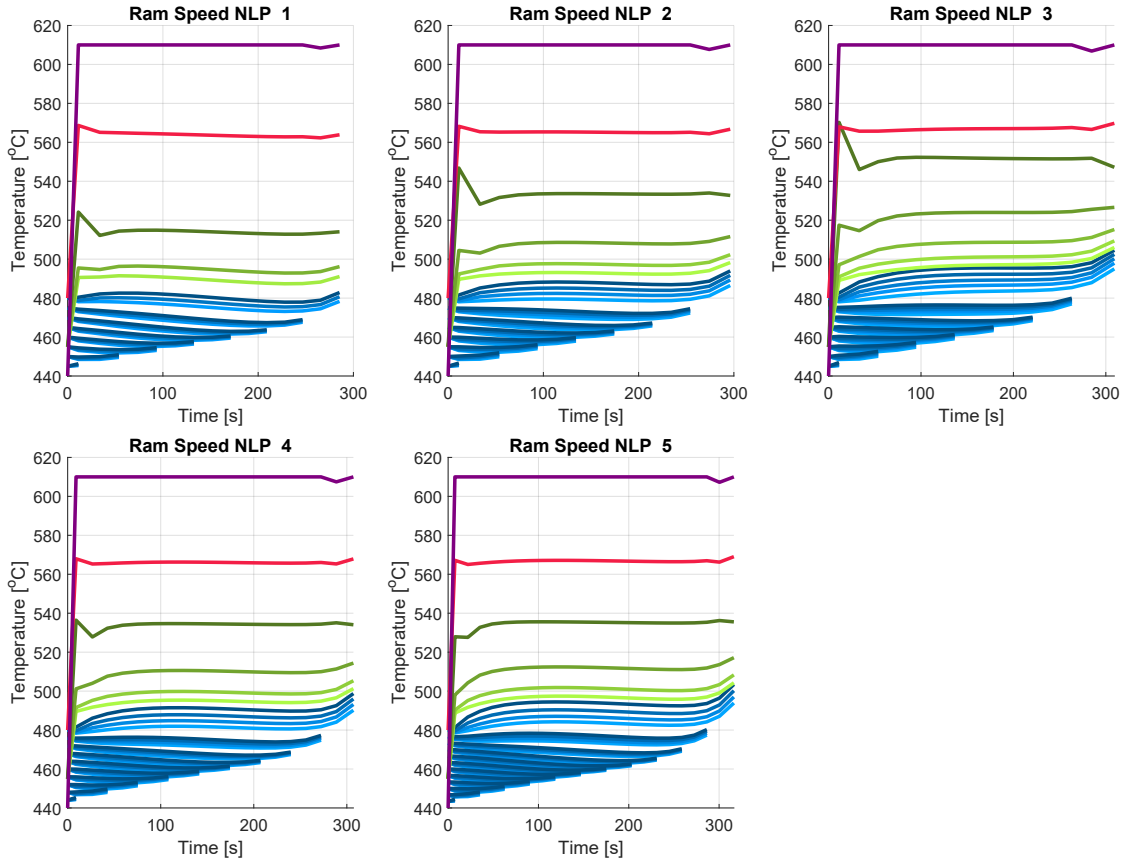


Figure 31: The temperature trajectory results of the tests defined in Table 19.
Temperatures: ■-peak, ■-port, ■-feeder, ■-billet.

It is clear that the spatial discretization of the billet, and subsequently the progressor discretization, affects the solution notably. Even though all NLPs yield quite similar temperature trajectories, and achieves isothermal extrusion at $T_{peak} \approx 610^oC$, we see that the billet temperatures are somewhat different, and the ram speed trajectories are not the same. These differences in the ram speed trajectories result in a spread in the total extrusion times; $[4min\,45s,\ 5min\,16s]$, which is a difference of $31s$, or relative uncertainty of $\pm 5\%$. Nevertheless, the ram speed trajectories all exhibit the same general shape, which will serve as a general reference/starting point for an MPC control scheme during extrusion, thus the exact optimal speed curve is not crucial to obtain.

The solve times are all in the range $[1.2s,\ 2.9s]$, except the last NLP, "Ram-NLP 5", which has the finest discretization and most decision variables and constraints, and is solved in $10s$. Thus, indicating that the solve time is generally low, although, is apparently vulnerable to unfortunate discretization. This phenomenon is also apparent by the drastic increase in number of required iterations, from 31 to 98, when decreasing the number of radial billet cells from 4 to 3 in Ram-NLP 2 and Ram-NLP 1.

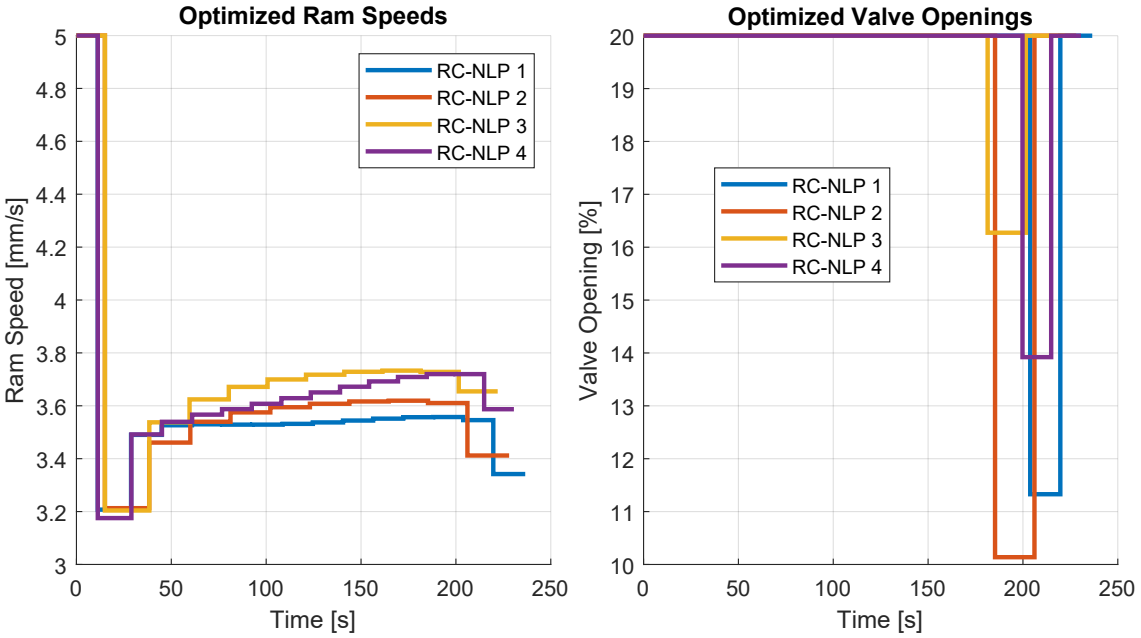
Generally, by optimizing the ram speed, we achieve extrusion times of around $5min$, which is equivalent of an $1 - \frac{300s}{337.5s} \approx 11\%$ improvement from the second simulation, which used $5min\,37.5s$ while also achieving isothermal extrusion at the desired peak temperature.

6.4 Optimization with Coolant

In this Section, both ram speed and coolant flow trajectories are optimized with respect to minimum extrusion time, given the same fixed initial taper as in the simulations and the ram speed NLPs. These solutions demonstrate the effect of also optimizing coolant flow, as opposed to only optimizing the ram speed, as suggested by Kulås (2022). Various NLPs are solved, with varying discretization, as seen in Table 20, where the extrusion times, solve times, and number of iterations are shown as well. The optimal ram speeds and coolant valve openings are shown in Fig. 32, and the temperature trajectories are shown in Fig. 33.

RC-NLP:	**1**	**2**	**3**	**4**
Billet	$(4, 8)$	$(4, 6)$	$(6, 6)$	$(3, 8)$
Container	$(2, 2)$	$(2, 2)$	$(2, 2)$	$(2, 2)$
Die	6	6	6	6
d	2	2	2	2
n_L	2	2	2	2
N_L	16	12	12	16
n_w	1698	1118	1400	1438
n_g	1711	1127	1409	1451
n_{eq}	1653	1085	1367	1393
n_{in}	58	42	42	58
t_f	236.5	228.1	222.0	230.6
t_{sol}	287.0	1.2	112.8	217.3
N_{iter}	7735	49	3535	8228

Table 20: Results of the Ram Speed Optimization NLPs. Symbols and colors have the same meaning as in Table 18.



(a) Optimal Ram Speeds

(b) Optimal Coolant Valve Openings

Figure 32: The optimal control trajectories of the NLPs defined by Table 20.

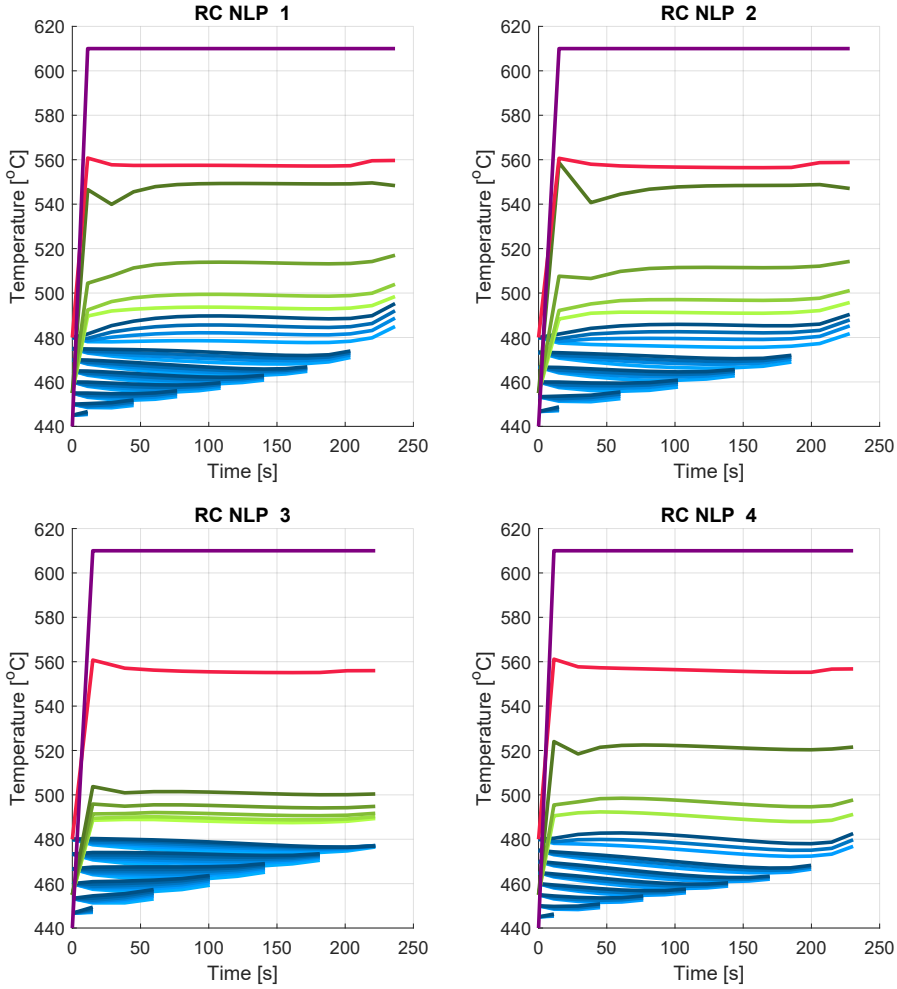


Figure 33: The temperature trajectory results of the tests defined in Table 20. RC $\rightarrow$ Ram Speed and Coolant NLP. Temperatures: ■-peak, ■-port, ■-feeder, ■-billet.

From Fig. 32a it is clear that the ram speed trajectories all resemble that of the solutions in Section 6.3, which only optimized ram speed. In the solutions presented here, however, the ram speeds are consistently higher, thus shortening the extrusion time, which is possible with the introduction of the coolant. Furthermore, form Fig. 32b, we see that the coolant flow is maximized over nearly the entire extrusion phase, with a reduction towards the end. This is consistent with the intuition that raising the ram speed and raising the coolant flow have opposite effects on the peak temperature, thus maximizing the coolant flow allows maximum ram speeds.

The extrusion times achieved when optimizing both coolant and ram speed are in the range $[3min\,42s,\ 3min\,56s]$, which is consistent, and improves upon the extrusion times obtained in Section 6.3 by $1 - \frac{230s}{300s} \approx 23\%$, and improves the non-optimized process by $1 - \frac{230}{337.5} \approx 32\%$. The solutions show a substantial improvement by the introduction of coolant to the extrusion process, however, as seen from Table 20, the solve times drastically increased from the sole ram speed optimization. The solve times after the introduction of coolant increased hundredfold, form the about $1s - 3s$ to about $100s - 300s$. This means that the solve times are in the same range as the extrusion times, which is not desirable, since the solve time must be smaller than the extrusion cycle period, which is assumed to be approximately equal to the extrusion time plus a small reload time. Although, we notice that for RC-NLP 2, the solve time remains in the same range as without cooling, and similarly with the iteration count. Since the solution of RC-NLP 2 is not notably different form the other solutions, the anomalous solve time is seemingly an artifact of 'randomness'. As in Section 6.3, the solve times and iteration counts seem sensitive to the initial guess and system discretization. This effect is also apparent by the difference in iteration count

between RC-NLP 3 and RC-NLP 4, which are 3535 and 8228 respectively, even though they have nearly the same number of decision variables and constraints.

6.5 Optimizing Initial Taper

This Section presents the main results, which are the solutions to the 'full NLP', (174), which expands upon the NLPs in Section 6.4 by allowing the use of the initial taper as a control variable. Various versions the full NLP are tested, where one batch ("Fine"/"FF") uses discretization similar to that of the discretization of the simulation NLPs in Section 6.2, and another batch ("Coarse"/"CF") explores the effects of using coarser discretization, with the goal of reducing the solve times. The Fine NLPs are presented in Table 21, and their solutions are presented in Fig. 34, Fig. 35, and Fig. 38, which show the coolant valve opening trajectories, the temperature trajectories, and the initial tapers respectively. All optimal ram speed trajectories of the full NLPs, both fine and coarse, are constant and equal to the maximum speed; $v_{ram,k} = 5mm/s \ \forall k \in \mathbb{Z}_{0,N-2}$ *(N is the number of state-discretization points before reducing NLP horizon due to a small unextruded "butt")*.

FF-NLP:	**1**	**2**	**3**	**4**	**5**[a]	**6**	**7**
Billet	$(6, 12)$	$(6, 12)$	$(6, 10)$	$(6, 10)$	$(6, 12)$	$(6, 10)$	$(6, 8)$
Container	$(2, 2)$	$(3, 3)$	$(3, 3)$	$(2, 2)$	$(2, 2)$	$(2, 2)$	$(2, 2)$
Die	6	6	6	6	4	4	6
d	2	2	2	2	2	2	2
n_L	2	2	2	2	3	3	4
N_L	32	30	36	36	20	24	24
N_w	4148	4498	3378	3088	6036	4482	4396
N_g	4168	4518	3394	3104	6068	4508	4424
N_{eq}	4067	4417	3311	3021	5919	4385	4295
N_{in}	101	101	83	83	149	123	129
t_f	172.5	172.5	171.0	171.0	171.0	174.0	174.4
t_{sol}	33.1	4655	50.2	6.7	2998	1030	8019
N_{iter}	143	18158	312	50	6658	4649	47027

[a] Return Status: "Solved_To_Acceptable_Level"

Table 21: Results of the fine full NLPs, (174), for various parameters. FF $\rightarrow$ Fine Full NLP. Symbols and colors have the same meaning as in Table 18.

The fine NLPs are considered the most 'accurate', since they utilize the finest discretization. From Fig. 35 we see that all the fine NLPs achieved isothermal extrusion at the desired temperature, and exhibit similar temperatures overall in the system progression. One difference between the fine NLP solutions to take note of is in the initial tapers. All the initial tapers exhibit linear descent towards the back-end of the billet, and agree on a temperature around 430^oC at the front-end, though the axial gradients vary somewhat. In fact, the initial taper descends by about 10^oC per cell, which is the minimum taper imposed on the initial taper; $\Delta t^-_{init} = 10^oC$, which means that all initial tapers are as 'flat' as possible, while maintaining a leftmost temperature of 430^oC to ensure isothermal extrusion. The varying axial billet discretization therefore causes the rightmost cells to be of varying temperature, since the constraint on the minimum taper is not scaled with the discretization.

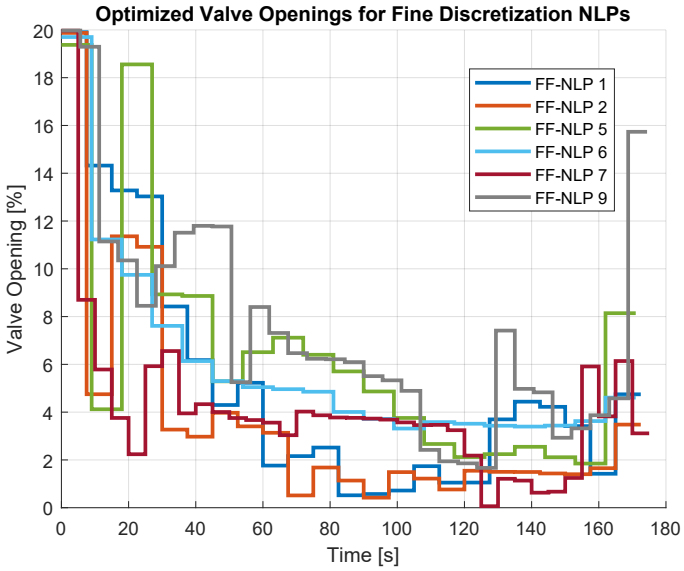


Figure 34: The valve opening trajectory results of the tests defined in Table 21. FF → Fine Full NLP.

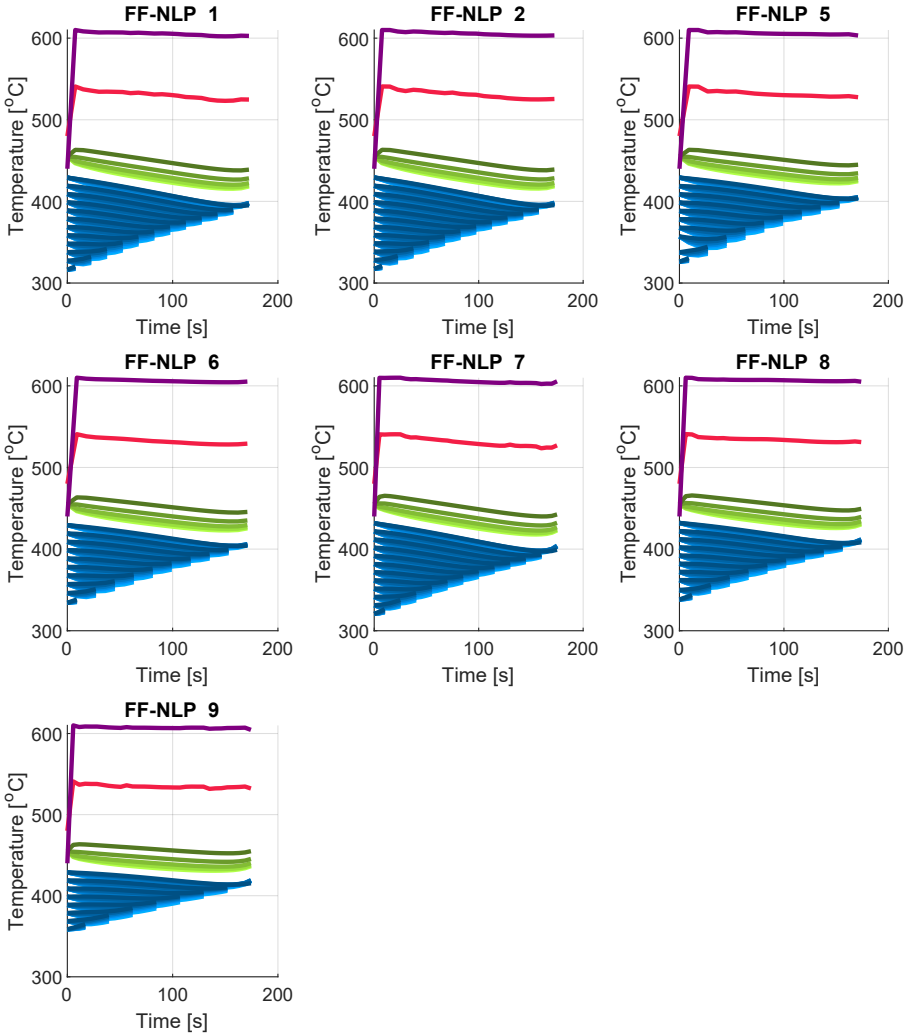


Figure 35: The temperature trajectory results of the tests defined in Table 21. FF → Fine Full NLP. Temperatures: ■-peak, ■-port, ■-feeder, ■-billet.

Furthermore, we notice that solve times have increased substantially from that of the simulations with similar discretization. The solve times of the 'fine full NLPs' range from $33s$ to over $2hours$. Of course, these solve time are too large, and must be reduced before implementing the ECOL. As with the previous results, the solve times vary drastically between configurations, and this effect should be understood to reliably solve the NLPs within the required time frame. The same effect is also present for the iteration count, as also observed for the previous results.

Since the ram speeds are maximized, the extrusion time is minimized by all the full NLPs. The equivalent extrusion time is approximately $2min\,50s$, with some variance because of the different discretizations. Note that the billet is not fully extruded in the implementations, but is only extruded until the next to last discretization point, and the length of the discretization interval therefore affects how much of the billet is not extruded in the various NLPs, thus causing minor variations in the extrusion times. By introducing the initial taper as a control variable, the extrusion time is reduced by $1 - \frac{170}{230} \approx 26\%$ from the case with only ram speed and coolant, and reduced by $1 - \frac{170}{337.5} \approx 50\%$ from the non-optimized process.

From Fig.34 we see that the various configurations produce similar overall behavior of the valve opening trajectories. All valve opening trajectories begin close to the maximum 20%, at the start of the press, then proceed to decrease as the process progresses, down to the $6\% - 2\%$ range, with a slight rise at the end. Nevertheless, the different trajectories experience fluctuations that are reminiscent of process noise, which is unique to each solution. Arguably, these fluctuations are not significant, as the exact nature of the trajectories do not have major impact on the initial taper, as evident by Fig. 38. Another, interesting observation is that, even though the the ram speed is maximized, the valve opening is not, as is the case in Section 6.4. The initial tapers found by the full NLPs are generally of much lower temperature than the fixed initial taper in Section 6.4, and likely compensates for the reduction in coolant flow, as was intended by (172).

The results of the 'coarse full NLP' batch are presented in Table 22, Fig. 36, Fig. 37, and Fig. 39, which present the discretizations, extrusion times, solve times, and integration count, the coolant valve opening trajectories, the temperature trajectories, and the inital tapers respectively.

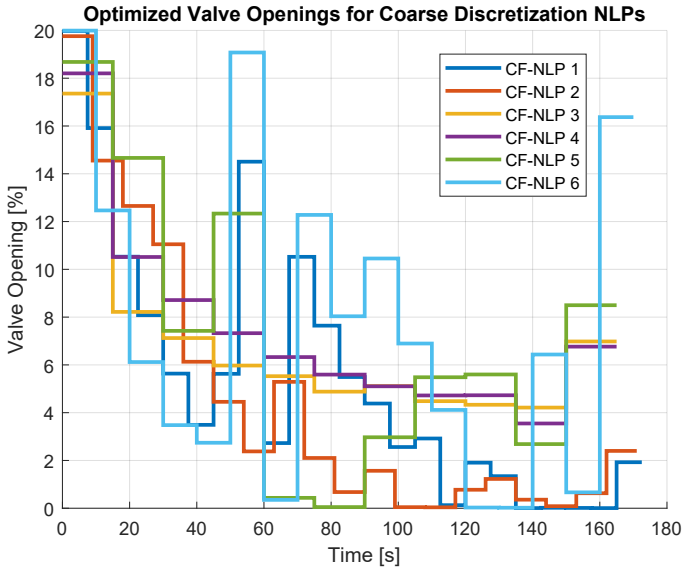


Figure 36: The valve opening trajectory results of the tests defined in Table 22. CF $\rightarrow$ Coarse Full NLP.

CF-NLP:	1	2	3	4	5	6
Billet	$(3, 12)$	$(3, 10)$	$(3, 6)$	$(3, 6)$	$(3, 6)$	$(3, 6)$
Container	$(2, 2)$	$(2, 2)$	$(2, 2)$	$(2, 2)$	$(2, 2)$	$(2, 2)$
Die	4	4	4	4	4	4
d	2	2	2	3	4	2
n_L	2	2	2	2	2	3
N_L	24	20	12	12	12	18
N_w	2470	1872	892	1169	1446	1357
N_g	2490	1888	900	1177	1454	1371
N_{eq}	2389	1805	853	1130	1407	1300
N_{in}	101	83	47	47	47	71
t_f	172.5	171.0	165.0	165.0	165.0	170.0
t_{sol}	65.5	653.2	2.2	4.8	10.7	67.8
N_{iter}	944	15811	110	175	265	2808

Table 22: Results of the coarse full NLPs, (174), for various parameters. CF $\rightarrow$ Coarse Full NLP. Symbols and colors have the same meaning as in Table 18.

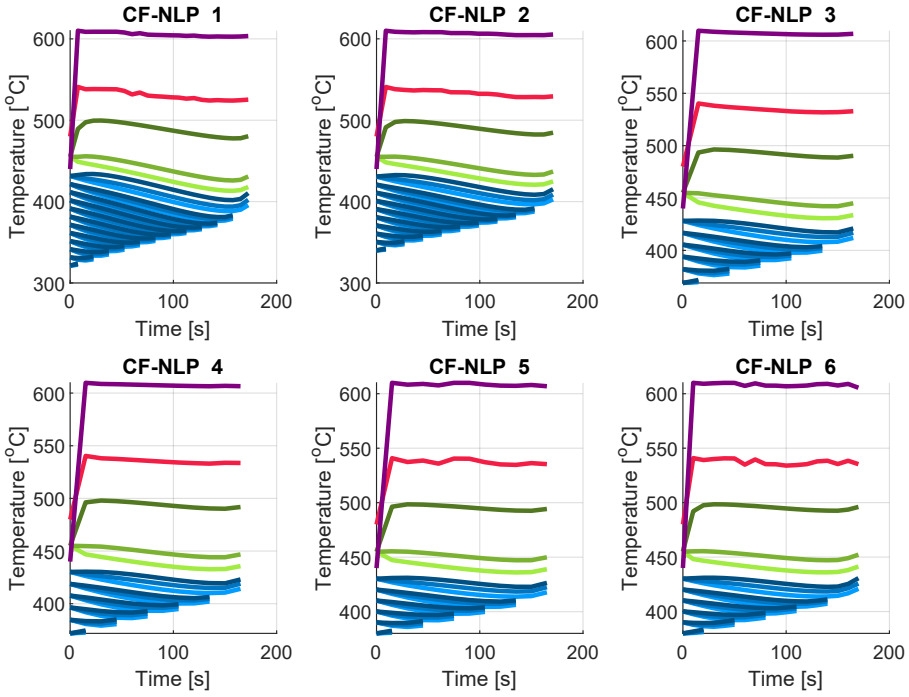


Figure 37: The temperature trajectory results of the tests defined in Table 22. CF $\rightarrow$ Coarse Full NLP. Temperatures: -peak, -port, -feeder, -billet.

The coarse full NLPs also all achieve isothermal extrusion at the desired peak temperature, as seen in Fig. 37. The point of using coarser discretization is to reduce the number of decision variables and constraints, and subsequently reduce the solve time. From Table 22, we see that the solve times are generally in the range of $1min$ or lower, with one exception at $10min\,53s$. These solve times are feasible with respect to a practical implementation of an ECOL at an extrusion plant, since the respective extrusion times are in still at about $2min\,50s$. However, we still see the unpredictable nature of the solve time, which further emphasizes the need to gain more understanding of the sensitive nature of the NLPs and how to construct NLPs with predictable solve times. Nonetheless, the main portion of the coarse NLPs have feasible solve times, and as seen in Fig. 36, Fig. 37, and Fig. 39, they all produce similar as solutions. We therefore obtain conceptually viable implementations of (174) by allowing somewhat coarse discretization.

The initial tapers consist of the same front temperature and linear axial gradient as with the fine

NLPs. The temperature differences between axial cells remain on the boundary of $\Delta t_{init}^{-} = 10^{o}C$, which means that the initial tapers remain as flat as possible, while maintaining $430^{o}C$ in the leftmost cells.

As with the fine NLPs, the coolant valve opening trajectories exhibit a 'noisy' behavior, and a somewhat general tendency to start high, and decrease as the process progresses. The fluctuating nature of the trajectories, however, is more pronounced for the coarse NLPs.

Lastly, we observe that the number of collocation points (2,3 or 4) does not seem to have any significant effect on the system trajectories, thus using $d = 2$ produces the same results with fewer variables and constraints, and shorter solve times.

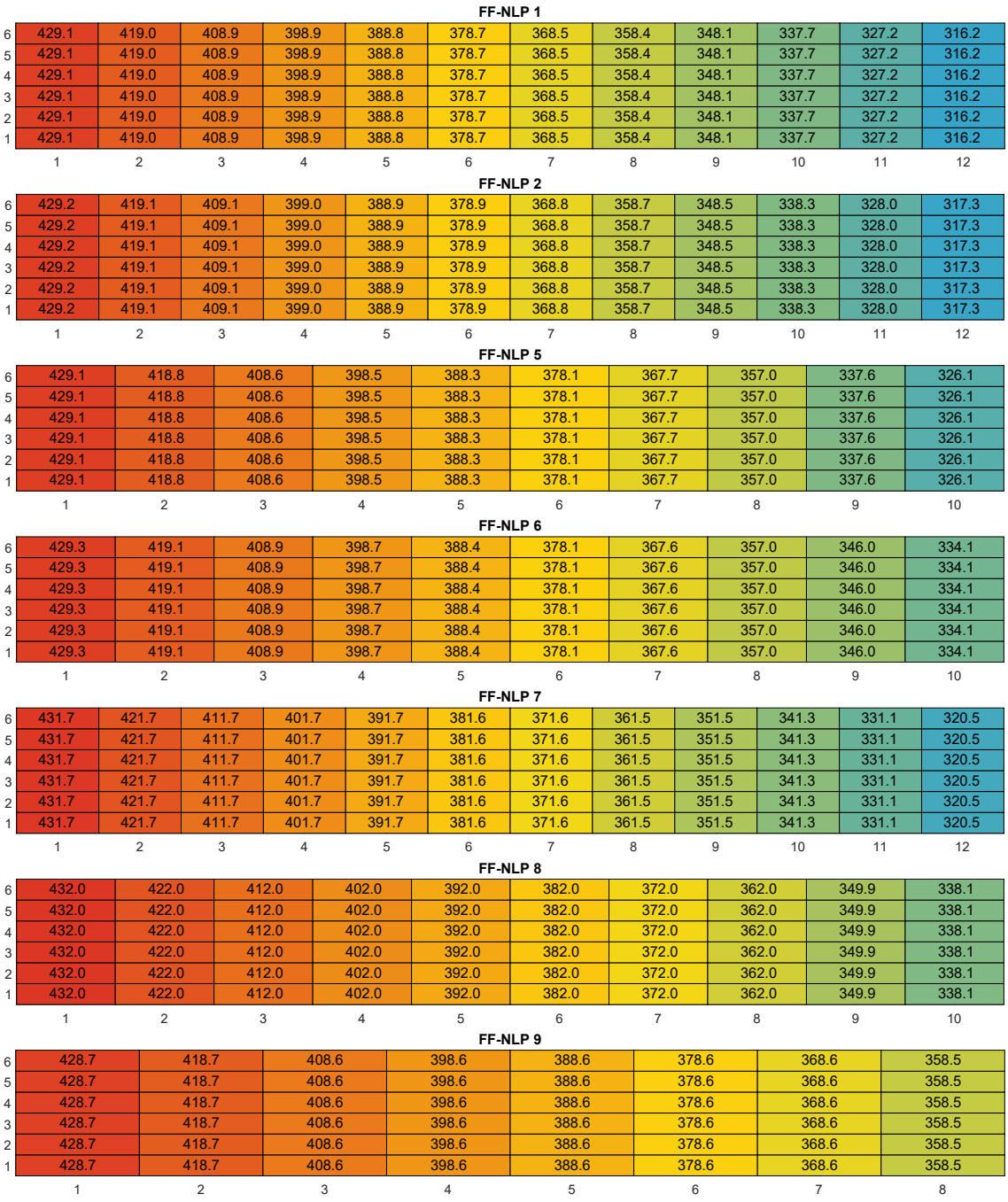


Figure 38: The optimal initial tapers with respect to the NLPs defined in Table 21. FF $\rightarrow$ Fine Full NLP. The color scheme is fixed with respect to the temperature values in all results of the set of Full NLPs.

Figure 39: The optimal initial tapers with respect to the NLPs defined in Table 22. CF $\rightarrow$ Coarse Full NLP. The color scheme is fixed with respect to the temperature values in all results of the set of Full NLPs.

7 Discussion

7.1 Advantages of the Proposed Method

7.1.1 Advantages of the Extrusion Cycle OCP Separation

The proposed extrusion cycle optimization scheme has several benefits, which are discussed here. First of all, the concept of having an optimization layer for the extrusion cycle is beneficial because it allows for the planning and tuning of initial conditions and reference trajectories for the various phases to minimize the overall extrusion cycle period, though this idea is not novel. This thesis presents a separation of the overall control problem into several smaller problems, and argues for the preserved optimality through the verifiable assumption that the required heating process trajectory for a given solution can be performed in less than double the extrusion time and reload time combined. The assumption is said to be verifiable through the implementation of the heating phase optimization by Gabrielsen (2022), or similar implementations. Additionally, assumptions were made on the billet taper constraints, which is further discussed in Section 7.4. The thesis presented the concept of the reduction margin in (122), complementing the separation principle, which is generalizable to any number of parallel available heater, as;

$$\mathcal{T}_{ext} \approx \max\left\{\frac{1}{n_h}t_h,\, t_{RL} + t_f\right\}, \tag{191}$$

where $n_h \geq 1$ is the number of parallel available heaters. Conveniently, this method can also be used to estimate the minimum necessary number of parallel heaters to optimize production, which is an interesting observation.

By separating the control problem into several smaller problems, we expect to reduce the overall solve time. For the sake of analysis, we assume that the solve time of an NLP is $\mathcal{O}(n^2)$, where n represents the number of decision variables (where we assume $n \propto n_g \propto n_w$), which is considered a conservative estimate of the solve time. This way, the solve time correspond nicely with the number of elements of the constraint Jacobian. By then considering the number of variables in an NLP corresponding to a non-separated NLP, $n_{w,tot}$, and the number of variables of the NLPs corresponding to the three separated phases; heating-, transit-, and extrusion phase, denoted $n_{w,h}$, $n_{w,t}$, and $n_{w,e}$ respectively, such that;

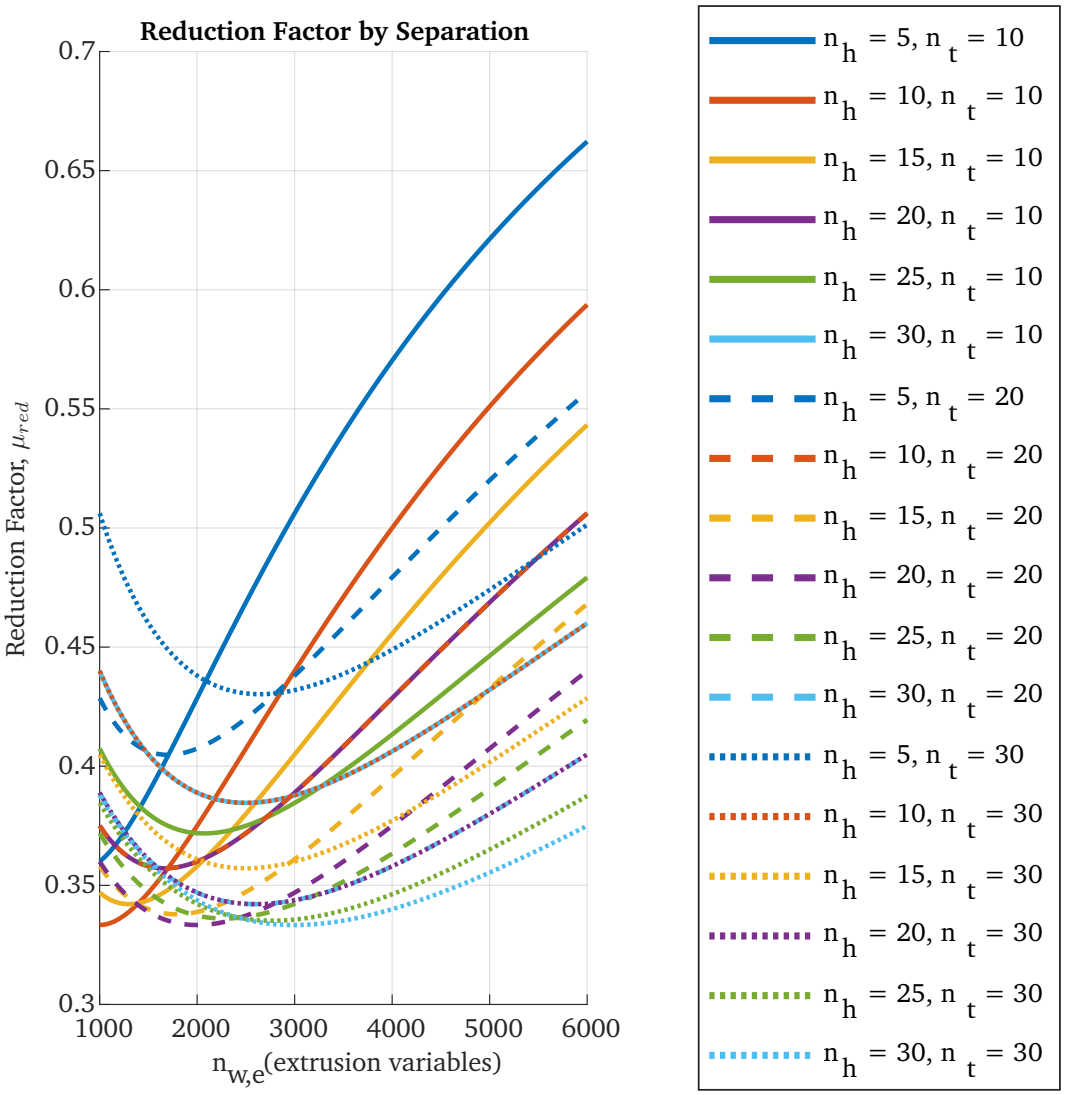
$$n_{w,tot} = n_{w,h} + n_{w,t} + n_{w,e}, \tag{192}$$

we get a solve time reduction factor of

$$\mu_{red} = \frac{n_{w,h}^2 + n_{w,t}^2 + n_{w,e}^2}{n_{w,tot}^2} = \frac{n_{w,h}^2 + n_{w,t}^2 + n_{w,e}^2}{(n_{w,h} + n_{w,t} + n_{w,e})^2}. \tag{193}$$

Example wise, Fig. 40 illustrates the reduction factor for various reasonable values of $n_{w,h}$, $n_{w,t}$, and $n_{w,e}$. We see that the under the assumed reasonable NLP sizes, the reduction factor is typically in the range $\mu_{red} \in [0.33,\, 0.5]$, which means that by separating the extrusion cycle OCP, we expect to have reduced the overall solve time by $50\% - 67\%$. This, of course, is a crude estimate. Another observation is that under the assumption that the heating time will not become the bottle neck, and using constant constraints on the initial taper, one may omit the open loop optimization of the heating phase and the simulation of the transit phase. In this case, the reduction factor is reduced to

$$\mu_{red} = \frac{n_{w,e}^2}{n_{w,tot}^2}. \tag{194}$$



(a) The reduction factor, μ_{red}.

(b) Legend corresponding to Fig. 40a.

Figure 40: The graphs represent the reduction factor arising from the separation of the extrusion cycle OCP, under the assumptions given in Section 7.1, for various values of $n_{w,h}$, $n_{w,t}$, and $n_{w,e}$. For simplicity of notation, (b) uses $n_{w,h} = 100n_h$, and $n_{w,t} = 100n_t$.

7.1.2 Problem Reduction by Progressor Transformation

Aside from the advantages of transforming the extrusion model to be progressed by extrusion length that are already discussed in Section 5.2.1, there is an additional advantage gained from employing this technique. As stated in Section 5.2.1, when the model is discretized in time, one does not know at what discretization points the various cells die, thus one must account for all cells at all discretization points. Technically, one can know something about at what discretization points they die, because of the upper and lower bound on the ram speed, yet one must still account for the worst case. By discretizing in extrusion length, one knows exactly at what discretization points all cells die, and one only needs to allocate variables for the correct number of cells.

By only accounting for the correct number of billet cells, the number of variables representing the billet section in the NLP is reduced by up to 50%, depending on the spatial discretization of the billet. We see this by considering the number of variables needed to account for the full length of the billet at all discretization points, including the collocation points, and the number of active cells when discretizing in extrusion length, to get

$$w_{L,billet} = \frac{1}{2}(1+d)n_r(n_x+1)N \tag{195}$$

$$w_{t,billet} = (1+d)n_r n_x N, \tag{196}$$

where $w_{L,billet}$ and $w_{t,billet}$ are the number of billet cells/variables in the extrusion length and time discretization cases respectively, d is the number of collocation points, (n_r, n_x) is the discretization of the billet, and N is the number of discretization points. The reduction is then measured by

$$\hat{\mu}_{red} = \frac{w_{L,billet}}{w_{t,billet}} = \frac{\frac{1}{2}(1+d)n_r(n_x+1)N}{(1+d)n_r n_x N} = \frac{1}{2} + \frac{1}{2n_x}, \tag{197}$$

which for a typical discretization of $n_x = 12$, amounts to $\hat{\mu}_{red} = \frac{1}{2} + \frac{1}{2\cdot 12} = 0.541\bar{6}$, meaning a reduction of $45.83\bar{3}\%$. We also see that with finer and finer axial discretization of the billet, the reduction approaches 50%;

$$\lim_{n_x \to \infty} \hat{\mu}_{red}(n_x) = \frac{1}{2}. \tag{198}$$

This is concept is illustrated in Fig. 41.

If we consider FF-NLP 1 from Section 6.5, the number of billet variables is $w_{L,billet} = 3744$, which is a substantial portion of the total $N_w = 4148$. The equivalent number of billet variables when accounting for all cells, active and inactive, is $w_{t,billet} = 6912$, which amounts to a total of $N_w^t = 7316$ decision variables. We then see that the overall reduction of problem size, in terms of decision variables, is reduced by $1 - \frac{4148}{7316} \approx 43\%$.

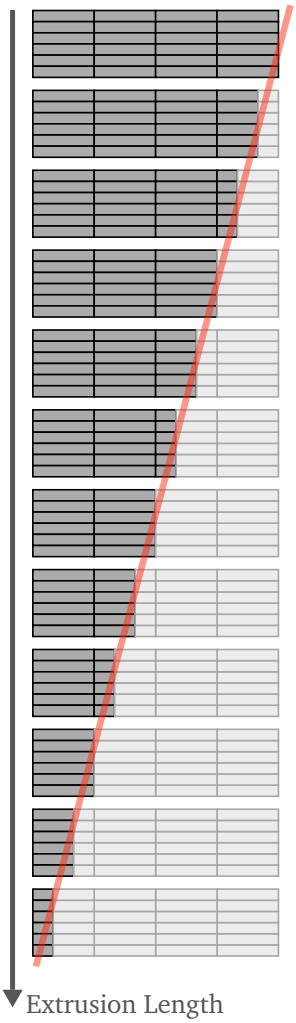


Figure 41: The figure illustrates how the proper allocation of variables for the billet cells throughout the NLP horizon contributes to reducing the necessary number of NLP variables. The dark cells indicate 'active' cells, and light cells indicate 'dead' cells. The red line illustrates the how the number of active cells are nearly half the total number of cells.

7.2 Applicability in a Real Industrial Setting

To the extent that one can regard the implemented extrusion model as a satisfactory mathematical twin of the real case study process, the method produced initial tapers and respective control trajectories that reduce the extrusion time by $1 - \frac{170s}{330s} \approx 48\%$ from the assumed $t_f = 330s$ of the current process. The new extrusion period is therefore $\mathcal{T}_{ext} \approx \max\{\frac{1}{2}300s,\, 23s + 170s\} = 193s$, and the new reduction margin is $193s - 150s = 43s \geq 0$, which means that the heating phase, if maintaining similar heating time, has not become the bottle neck. Of course, the heating phase problem may be solved as well, to confirm the that $\frac{1}{2}t_h^\star \not> 193s$. By assuming model validity, the method is therefore applicable in the sense that it contributes to the increase in production, by simply providing a tailored initial taper. By providing the respective control trajectories to the extrusion phase MPC scheme, the tailored initial taper may be exploited by the local MPC controller, in order to accomplish the optimal extrusion process predicted by the ECOL. It is also worth mentioning that even if the heating phase becomes the bottle neck, the solution may still be optimal with respect to the overall extrusion period, although this is not guaranteed.

If one considers the implementation to be representative of some generic extrusion process, and not necessarily the specific case study process, the effectiveness of the method may instead be measured by comparison to the simulations, which represent non-optimized extrusion phases of the hypothetical extrusion process. We therefore compare simulation 2 in Section 6.2 to the optimized processes in Section 6.3, Section 6.4, and Section 6.5. The overall reduction from the non-optimized simulation to the fully optimized process is, as stated in Section 6.5, about 50%, which is consistent with the reduction with respect to the case study process. We also saw that by the introduction of the initial taper as a control variable, versus only optimization of the ram speed and coolant flow, the extrusion time is reduced by about 26%, which further demonstrates the effect of an extrusion cycle optimization layer.

In terms of applicability, it is worth considering the necessity of an extrusion cycle optimization layer versus the difficulty of implementing it in practice. This aspect is brought to the readers attention because of the seemingly predictable nature of the optimal initial tapers. All the optimal initial tapers found, for the various discretizations, seem to follow a simple pattern; the leftmost radial column of cells should have the temperature that corresponds to the desired peak temperature at maximum speed, and the remaining cells should descend as little as possible, that is, along the minimum linear taper. Because of this pattern, a certain predictability emerges, that, perhaps, allows one to bypass the cumbersome implementation of an extrusion cycle optimization layer, and compute high performing initial tapers by following simple rules. Of course, finding the correct left-temperature is not straight forwards, though should be easier than implementing the entire ECOL. Another caveat to this simple bypass, is the lack of respective control trajectories, which may be necessary for the local MPC scheme to fully exploit the computed initial taper.

7.3 Model Discrepancies

There are several deliberate and known discrepancies between the model as developed by Halås (unpublished), hereby referred to as the *'base model'*, and the adapted version in this thesis, hereby referred to as the *'current model'*. These differences will cause deviations in the solutions, which are important to be aware of and understand upon comparing the models. Even though a rigorous comparison between the two model implementations is made in this thesis, it is useful to discuss some of the discrepancies between them to gain understanding of what differences to expect. Such expectations may help detect inherent flaws regarding the method used, and disregard small deviations caused by the different nature of the implementations, and importantly, explain the needed scaling factors shown in Section 6.1.

The most notable model discrepancies are;

- the current model is integrated using orthogonal collocation, while the base model is integrated using an explicit RK-scheme,
- the base model is discretized in regular time intervals, whereas the current model is discretized in regular intervals of the extrusion length,
- the sizes of the discretization intervals are not only regular in different progressors, but may also be of significantly different magnitude,
- the exit section is not modelled in the current model, while it is modelled in the base model,
- the spatial discretization of the various sections may vary between solutions and may therefore be a source of trajectory deviation,
- the base model uses 2 axial cells for the port section,
- and the base model uses the previous values for port and peak in the nonlinear terms of the reduction and viscous dissipation energy.

Using different integrators may have a significant effect on the calculated system trajectories, especially when using coarse discretization over long horizons. Because of the different integrators, in combination with the different and varying step sizes, the trajectories are expected to deviate somewhat in regions of rapid changes such as the start of the extrusion phase until a steady state is reached, corresponding to zone I in Fig. 7, and then only gradually deviate as the system progresses. Since the exit temperature is not modelled, the peak temperature is expected to decrease more in the base model during low ram speeds, as the exit speed of the aluminium is slower. However, because of the generally high exit speeds during normal operations, which negate the heat diffusion upstream towards the peak section from the exit section, this difference is not expected to cause any significant deviation. The accuracy of the approximation of the advection equation rapidly decreases with coarse discretization, which corresponds to the axial discretization of the billet. Different axial discretization of the billet between the base and current models may therefore cause the temperature taper to evolve differently for the two models. The division of the port section into 2 cells in the base model allows for a gradient across the section, but is not expected to cause great trajectory deviation. However, upon potentially experiencing difficulty validating the model in future work, one should consider matching the the number of port cells between the two models. Lastly, the fact that the base model uses previous port and peak temperatures in the nonlinear terms to allow the explicit calculation of the algebraic states is, after minor testing, expected change the dynamics somewhat, with most notable effect on the scaling necessary for the reduction and viscous dissipation energy. Since the current model avails a simultaneous method, the proper implicit equations are trivially implemented. The proper implicit algebraic equations are therefore used, and the scaling factors given in Section 6.1 are applied to tune the model. In total, given the many discrepancies, it is not exactly clear where the need for the significant scaling in Section 6.1 arises.

7.4 Transit IVP and Feasible Region of the Initial Taper

The extrusion cycle optimization layer (ECOL) is contingent on being able to solve the transit initial value problem (IVP). During the transit phase, the billet experiences heat diffusion and loss to its environment during its transportation to the press. This phase determines what the initial taper of the extrusion phase is, based on the end taper of the heating phase. This is formulated as an IVP, since there are no control variables in this phase, but rather only an initial condition and a set of ordinary differential equations (ODEs). We see that the feasible end tapers of the heater can be transformed through the IVP to their respective initial tapers, thus seemingly forming a feasible region for the initial taper, that may be imposed as constraints in the extrusion phase control problem. However, it is not obvious whether of not the the boundary of the set feasible end-tapers of the heating phase exactly maps to the boundary of the transformed set. Furthermore, it is not even clear how to define such a boundary, as the taper can take many complex shapes, and system limitations may be concerned with both maximum temperatures and maximum temperature gradients. Constructing the feasible region for the initial tapers is therefore is not trivial, and mapping out the feasible region of the heater end taper is not trivial in itself. In this thesis, a simple constraint is imposed on the maximum and minimum gradients of the initial taper, as well as the general system temperature constraints. To fully exploit the concept of optimizing the initial taper, it is important to understand the limitations of the heater and how it translates through the transit phase.

Importantly, the transit IVP must also be calculated backwards, with the initial taper as the initial condition, and the respective end taper of the heating phase as the variable to find. This way, one may compute the reference taper for the heater based on the optimal initial taper. By imposing the proper constraints on the initial taper of the OCP B control problem, one ensures that this reference is feasible with respect to the heating phase. The transit IVP must be solved for every cycle, since the optimal initial taper may change form cycle to cycle, depending on for example changes in the other initial temperatures of the extrusion phase, or updates in the model parameters. Additionally, the ambient temperature may affect the transit dynamics. Because of possibility of varying transit dynamics, and perhaps even updates in estimates of the alloy properties of the billet, the feasible region of the heater end tapers may be updated regularly as well, which should be taken into account when implementing an ECOL in an extrusion plant. Depending on the method chosen to compute the feasible initial taper, the solve time for this may be high, and should perhaps be done in parallel to the ECOL and updated every few cycles.

7.5 Initial Guess

The initial guess used for the various optimization problems solved in this thesis is based on an arbitrary, generic result, which is considered to be equally 'wrong' for all tests, and does not affect the general size of the solve time and iteration count. It may, however, affect how the solve times and iteration counts vary between the tests, as the initial guess may 'randomly' favor some of the NLPs. Especially, the simulations are affected by the initial guess, since the generic solution used as the initial guess achieves isothermal extrusion at the desired peak temperature, which is only the case for simulation 2.

The initial guess for all temperatures are interpolated based on the common generic solution, except for the billet, which uses a constant 500^oC as the initial guess. Additionally, ϵ and $\hat{\lambda}$ are guessed to be constant 0 and $6.9444 \cdot 10^3$ respectively, and the ram speed v_{ram} and coolant flow variable $\hat{z}$ are guessed to be constant and equal to $0.005m/s$ and 0.2 respectively, where relevant. In an extrusion plant implementation of the methods proposed in this thesis, the initial guess that is used may be adapted dynamically, as the solution to the previous cycle is likely to be close to the solution of the next cycle. Also, upon changing the product extrudate profile, or aluminium alloy, the initial guess may be based on a solution found when previously using similar alloys or profiles.

Upon minor testing with the implemented NLP, it seems that the initial guess is crucial in achieving low solve times. It turns out that a generic initial guess performs well, nevertheless, even better performance may be achieved by specializing the initial guess for every cycle. Of course, providing a reasonable, non-constant initial guess for the billet may also improve the performance notably,

as the billet cells account for a substantial portion of the decision variables.

7.6 Regularization

A small point is made about the lack of regularization of the extrusion phase NLP, even though some regularization is already present, as discussed in Section 5.2.2. As seen from Fig. 34 and Fig. 36, the coolant valve opening trajectories vary notably between solutions, even though a general shape seems to be present. To attempt to make the exact trajectories more predictable, and converge to a 'true' optimal trajectory upon increasing the number of discretization points, one may try to regularize the NLP even more. By regularizing the NLP, specifically in terms of the coolant valve opening, one transforms the cost of a region of nearly equally optimal valve opening trajectories such that it more strongly favors a particular solution in that region more than others. That way, minor variation in the problem otherwise will have less of an impact on the optimal valve opening trajectory. Regularization affects the overall NLP solution, however, and comes at the cost of distorting what is considered optimal. Therefore, it is not trivial to design regularization measures, as the resulting trade-off between regularization and optimality with respect to the original problem is not necessarily easy to understand, and it is not necessarily clear what is a desirable trade-off.

An example of a form of regularization is to penalize the difference between two consecutive valve opening values, which is also touched upon in Section 5.2.2, but is not done for the results in Section 6. Some testing with various regularization terms in the objective has been explored, though the results are not presented, and no remarkable improvement was found. If more predictable control trajectories, that is; similar across different NLP configurations, are deemed necessary in future work, some form of regularization should be considered.

7.7 Further Work

This thesis comprehensively covers the idea- and implementation of an extrusion cycle optimization layer that makes use of the initial taper to minimize the extrusion times beyond what is possible with only the ram speed and coolant flow while maintaining isothermal extrusion at the desired peak temperature, by accounting for the complete extrusion cycle. Nevertheless, there are several aspects of the method that may be further studied, to improve upon the idea and execution. Here, we list the most obvious, notable, and direct areas that one should study further to improve upon the concept and implementation:

- Model validation - Upon implementing an extrusion cycle optimization layer, one should properly validate the model against the real process, to ensure feasibility with respect to the physical extrusion process.
- Implement and test the complete ECOL - This thesis covers the concept of simplifying an ECOL implementation by separating the overall control problem into several smaller problems, and covers only the implementation of one of them; the extrusion phase control problem. The complete implementation of all phases and their connections to each other should be explored further, and implemented and tested.
- Understanding the feasible regions of the initial taper and the heater end-taper better - To properly implement constraints on the initial taper based on the feasible region of the heater end-taper, one must understand the limitations of the heater, and how to define a meaningful and useful boundary of the feasible heater end-taper region. This boundary should be such that it is easily convertible via the transit IVP, and implementable as constraints on the initial taper.
- Methods of specializing the initial guess - By choosing initial guesses that are close the solution, the algorithm is likely to converge faster. Therefore, it is beneficial to construct a scheme that produces a reasonable initial guess, in order to reduce the solve time of the extrusion phase NLP. One should further study how to use the information from previous solutions to make reasonable guesses for a given cycle.
- Solve time/Iteration count sensitivity to problem discretization - Upon testing several discretizations of the same problem, we observed the occasional spike in solve time and iteration count. In a practical implementation of the ECOL, it is desirable to be confident in the feasibility of the solve time, that is $t_{sol} < t_f + t_{RL}$, such that the production does not halt because the next cycle is waiting for the ECOL to find the optimal open loop trajectories. One should therefore understand why- and what makes the solve time spike for some configurations, and how to prevent this effect. This may be closely connected to the issue of finding good initial guesses, and perhaps regularization.
- Linear solvers - The optimization algorithm (IPOPT, Wächter and Biegler (2006)) used in this thesis, makes use of external linear solvers for the solving the large and sparse linear systems necessary to find the descent directions. This allows one to use linear solves that are suited for the particular problem in question. For simultaneous methods, these systems are usually highly sparse, and benefits from using linear solvers that exploit sparsity. Finding the best linear solver for the the particular problem may greatly reduce the solve time. The solver used in this thesis is MUMPS, and we recommend testing solvers such as the HSL (Harwell Subroutine Library) linear solvers; MA27, MA57, HSL_MA77, HSL_MA86 and HSL_MA97.
- Optimization Algorithms - It is natural to investigate whether there are other algorithms than IPOPT that may be advantageous when solving the NLPs implemented in this thesis.

8 Conclusion

The objective of this thesis is to conceptualize a practical solution to a extrusion cycle optimization layer, as part of a greater extrusion process control hierarchy (Fig. 3), and to partially implement a solution, so as to provide insight into the effectiveness of such a layer as proof of concept.

We have presented a practical simplification of the extrusion cycle optimization problem, which is based the separation of the overall process into three distinct phases; the heating-, transit-, and extrusion phase, and optimizing them in backwards order; first optimize the extrusion phase, then solve the transit IVP, and lastly, solve the heating phase problem. We found that by assuming that the heating time does not increase significantly when varying the initial taper, optimizing only the extrusion process is equivalent to optimizing the overall extrusion cycle under a reduction margin, after which the heating time may be confirmed by solving the transit IVP and heating problem, thus confirming the optimality of the solution. A difficulty regarding the feasible region of the initial taper and heater end-taper has been discussed, such as the difficulty of defining such a region, and how to convert it onto the equivalent feasible region for the initial taper. This region may vary from cycle to cycle, and should ideally be computed every cycle by the ECOL. However, because of the potential difficulty and time consumption of such a computation, a potential parallel computation is suggested, that may compute a feasible region over several cycles, and update the estimate every few cycles. Otherwise, simple linear constraints are also possible.

A partial implementation of the ECOL, specifically the extrusion phase, has been covered extensively, and several practical considerations and solutions to various aspect and obstacles have been addressed. The extrusion phase optimization problem has been converted into an equivalent NLP, by using a simultaneous approach; *direct collocation*. Using a simultaneous approach seemed problematic since part of the objective is to optimize the ram speed. Without knowing the ram speed prior to solving the NLP, one does not know the extrusion length as a function of time, and since the dynamic equations are dependent on the extrusion length, the model equations are unknown in time as well. However, by exploiting the inherent extrusion process criteria that the ram speed must always be strictly positive, we concluded that the extrusion length is a progressor of the extrusion phase. Since we only need to optimize the extrusion phase, as a result of the extrusion cycle OCP separation, this was sufficient to be able to perform a progressor transformation of the model equations. The progressor transformation then made it possible to efficiently implement the model dynamics as constraints in the extrusion phase NLP. Additionally, the spatial discretization of the billet resulted in discontinuous changes in the model dynamics at unknown times, which is also an artifact of the unknown ram speed. However, these discontinuities happen at known values of the extrusion length, thus after the progressor transformation, we were able to only integrate the system between the discontinuities, effectively 'hiding' the discontinuities at the discretization points. We also observed that, by performing a progressor transformation into extrusion length, one knows what billet cells are still active at every discretization point, meaning that one only has to allocate variables for the active cells. Depending on the discretization used, this observation allows us to reduce the number of billet cells by up to 50%, and we saw that for a typical implementation of the NLP, the overall number of decision variables was reduced by 43%.

Several results were presented, were different spatial and progressor discretizations, number of collocation points, and discretization points per billet cell were tested for different implementations of the extrusion phase NLP. The first results were of a version of the NLP where all control variables were fixed, thus simulating a non-optimized extrusion phase. Secondly, the ram speed was freed to be optimized, while the coolant valve opening and initial taper remained fixed, followed by implementations were the coolant valve opening was also freed. Lastly, the initial taper was freed to accommodate the complete extrusion phase optimization problem. We found that with each new control variable that was optimized, the achievable extrusion times became shorter. By comparing the non-optimized simulations to the fully optimized process, we found that the extrusion time was reduced by about 50%. Similarly, introducing initial taper optimization reduced the extrusion time by about 26% from the case with only ram speed and coolant flow optimization, and introducing coolant optimization yielded a 23% reduction from ram speed optimization only. To fully conclude on the effectiveness of the method, however, the model should be validated by comparison to the relevant real process, and experimental data should confirm the reduced extrusion times.

Bibliography

Andersson, Joel A E et al. (2019). 'CasADi – A software framework for nonlinear optimization and optimal control'. In: *Mathematical Programming Computation* 11, pp. 1–36.

Aukrust, Trond and Sami LaZghab (2000). 'Thin shear boundary layers in flow of hot aluminium'. In: *International Journal of Plasticity* 16.1, pp. 59–71. ISSN: 0749-6419. DOI: https://doi.org/10.1016/S0749-6419(99)00047-9. URL: https://www.sciencedirect.com/science/article/%20%5C%5C%20pii/S0749641999000479.

Betts, John T (2010). *Practical methods for optimal control and estimation using nonlinear programming*. SIAM.

Biegler, Lorenz T (2007). 'An overview of simultaneous strategies for dynamic optimization'. In: *Chemical Engineering and Processing: Process Intensification* 46.11, pp. 1043–1053.

— (2010). *Nonlinear programming: concepts, algorithms, and applications to chemical processes*. SIAM.

Birol, Yücel (2004). 'The effect of homogenization practice on the microstructure of AA6063 billets'. In: *Journal of Materials Processing Technology* 148.2, pp. 250–258.

Birol, Yucel (2014). 'Homogenization of direct chill cast AlSi1MgMn billets'. In: *International journal of materials research* 105.1, pp. 75–82.

Borrelli, Francesco (2003). *Constrained optimal control of linear and hybrid systems*. Vol. 290. Springer.

Chanda, Tapas, Jie Zhou and Jurek Duszczyk (2001). 'A comparative study on iso-speed extrusion and isothermal extrusion of 6061 Al alloy using 3D FEM simulation'. In: *Journal of Materials Processing Technology* 114.2, pp. 145–153.

Cuéllar Matamoros, Carlos Fernando (1999). 'Modeling and control for the isothermal extrusion of aluminium'. PhD thesis. ETH Zurich.

Diehl, Moritz and Sébastien Gros (2011). 'Numerical optimal control'. In: *Optimization in Engineering Center (OPTEC)*.

Evans, Lawrence C (2022). *Partial differential equations*. Vol. 19. American Mathematical Society.

Farjad Bastani, Amin, Trond Aukrust and Sverre Brandal (2010). 'Study of isothermal extrusion of aluminum using finite element simulations'. In: *International Journal of Material Forming* 3, pp. 367–370.

Gabrielsen, Trym Arve L. (2022). 'Methods for dynamic optimization with application to the aluminum extrusion process'.

Halås, Anne (unpublished). *Modelling of the extrusion process – Press*. Documentation of Cybernetica's heater model.

Han, CH, Dong-Yol Yang and M Kiuchi (1986). 'A new formulation for three-dimensional extrusion and its application to extrusion of clover sections'. In: *International journal of mechanical sciences* 28.4, pp. 201–218.

Hölker, Ramona et al. (2013a). 'Controlling heat balance in hot aluminum extrusion by additive manufactured extrusion dies with conformal cooling channels'. In: *International Journal of Precision Engineering and Manufacturing* 14, pp. 1487–1493.

— (2013b). 'Controlling heat balance in hot aluminum extrusion by additive manufactured extrusion dies with conformal cooling channels'. In: *International Journal of Precision Engineering and Manufacturing* 14, pp. 1487–1493.

Holven, Stian (2020). 'Modelling and Model Predictive Control of an Extrusion Press for High Precision Aluminium Profiles'. MA thesis. NTNU.

Kim, Young-Tae and Keisuke Ikeda (2000). 'Flow behavior of the billet surface layer in porthole die extrusion of aluminum'. In: *Metallurgical and Materials Transactions A* 31, pp. 1635–1643.

Kulås, Mats (2022). 'Dynamic Optimization and State- and Parameter Estimation used on an Extrusion Press for Aluminium'. MA thesis. NTNU.

Li, Q et al. (2003). 'Finite element investigations upon the influence of pocket die designs on metal flow in aluminium extrusion: Part I. Effect of pocket angle and volume on metal flow'. In: *Journal of Materials Processing Technology* 135.2-3, pp. 189–196.

Mahmoodkhani, Yahya et al. (2014). 'Numerical modelling of the material flow during extrusion of aluminium alloys and transverse weld formation'. In: *Journal of Materials Processing Technology* 214.3, pp. 688–700.

Murai, Tsutomu et al. (2003). 'Effects of extrusion conditions on microstructure and mechanical properties of AZ31B magnesium alloy extrusions'. In: *Journal of Materials Processing Technology* 141.2, pp. 207–212.

Özışık, M Necati (1993). *Heat conduction*. John Wiley & Sons.

Qamar, SZ (2010). 'Shape complexity, metal flow, and dead metal zone in cold extrusion'. In: *Materials and Manufacturing Processes* 25.12, pp. 1454–1461.

Rawlings, James Blake, David Q Mayne and Moritz Diehl (2017). *Model predictive control: theory, computation, and design*. Vol. 2. Nob Hill Publishing Madison, WI.

Saha, Pradip K (2000). *Aluminum extrusion technology*. Asm International.

Sellars, CM and WJ McG Tegart (1972). 'Hot workability'. In: *International Metallurgical Reviews* 17.1, pp. 1–24.

Sheppard, T and DS Wright (1979). 'Determination of flow stress: Part 1 constitutive equation for aluminium alloys at elevated temperatures'. In: *Metals Technology* 6.1, pp. 215–223.

Sheppard, Terry (2013). *Extrusion of aluminium alloys*. Springer Science & Business Media.

Strikwerda, John C (2004). *Finite difference schemes and partial differential equations*. SIAM.

The MathWorks Inc. (2023). *MATLAB Version: 9.14.0.2239454 (R2023a) Update 1*. Natick, Massachusetts, United States. URL: https://www.mathworks.com.

Tibbetts, Brian and John Wen (1995). 'Application of modern control and modeling techniques to extrusion processes'. In: *Proceedings of International Conference on Control Applications*. IEEE, pp. 85–90.

Valberg, Henry (1992). 'Metal flow in the direct axisymmetric extrusion of aluminium'. In: *Journal of Materials Processing Technology* 31.1-2, pp. 39–55.

Wächter, Andreas and Lorenz T Biegler (2006). 'On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming'. In: *Mathematical programming* 106.1, pp. 25–57.

Wächter, Andreas, Stefan Vigerske et al. (n.d.). *print_options_documentation*. URL: https://coin-or.github.io/Ipopt/OPTIONS.html#OPT_hsllib.

Wazwaz, Abdul-Majid (2002). *Partial differential equations*. CRC Press.

Zhou, Jie, Luo-xing Li and J Duszczyk (2004). 'Computer simulated and experimentally verified isothermal extrusion of 7075 aluminium through continuous ram speed variation'. In: *Journal of Materials Processing Technology* 146.2, pp. 203–212.

Appendix

A Listings

Listing 5: Code that instantiates an *ExtrusionProcess()*, builds the corresponding NLP, and solves it.

```
% Define porperties that do not take default value:
nlp.d = 2;              % Number of collocation points
nlp.nL = 2;             % Number of disc. points per billet cell
G.nx.billet = 10;       % Number of axial billet cells
G.nr.billet = 6;        % Number of radial billet cells
P.rho.aluminium = 2752;  % Density of the incoming aluminium billet

nlp.reference_solution = 'Reference.mat' % Provide initial guess to the solver

% Instantiate extrusion process:
myEP = ExtrusionProcess('G',G,'P',P,'nlp',nlp);

% Build NLP:
myEP.BuildNLP

% Solve NLP:
my.EP.SolveNLP
```

Listing 6: Defining SX variables.

```
for name = ⟨loop over sections⟩
    nr = NLP.G.nr.(name);               % Retrieve nr for section
    nx = NLP.G.nx.(name);               % Retrieve nx for section
    T_SX.(name) = SX.sym(['T_SX_',char(name),'_0'],nr*nx);
end
for name = ⟨loop over control variables⟩
    C_SX.(name)   = SX.sym([char(name),'_SX_0']);
end
for name = ⟨loop over parameters⟩
    P_SX.(name) = SX.sym([char(name),'_SX_0']);
end
```

Listing 7: Defining initial optimization variables.

```
for name = ⟨loop over sections⟩
    nr = NLP.G.nr.(name);               % Retrieve nr for section
    nx = NLP.G.nx.(name);               % Retrieve nx for section
    T.(name) = MX.sym(['T_',char(name),'_0'],nr*nx);
    NLP.CLC.S.x.(name){1} = T.(name);
end
for name = ⟨loop over control variables⟩
    C.(name)                = MX.sym([char(name),'_0']);
    NLP.CLC.S.u.(name){1}   = C.(name);
end
for name = ⟨loop over parameters⟩
    P.(name) = MX.sym([char(name),'_0']);
    NLP.CLC.S.p.(name){1}   = P.(name);
end
```

Listing 8: Defining collocation variables.

```
for name = ⟨loop over sections⟩
    nr = NLP.G.nr.(name);
    nx = ⟨retrieve nx of section at current disc. point⟩ % This value is not constant for the billet section
    for d = 1:NLP.nlp.d
            T_coll{d}.(name) = MX.sym(['T_',char(name),'_', k_str,'_coll_', num2str(d)],nr*nx);
            NLP.CLC.S.coll.(name){end+1} = T_coll{d}.(name);
    end
end
```

Listing 9: Defining algebraic and parameter equations.

```
for name = ⟨loop over sections that are independent of the billet section⟩
    dT_SX.(name) = NLP.dT.(name)(NLP,T_SX,C_SX,P_SX,k+1);
    dT.(name) = Function(['dT_',char(name)],Input_Cell_SX_algebraic,{dT_SX.(name)});
end
for name = ⟨loop over parameters⟩
    Pf_SX.(name) = NLP.Pf.(name)(NLP,T_SX,C_SX,P_SX,k);
    Pf.(name) = Function(['Pf_',char(name)],Input_Cell_SX_algebraic,{Pf_SX.(name)});
end
```

Listing 10: Solving for parameters.

```
⟨construct Input_Cell{}⟩
for name = ⟨loop over parameters⟩
    g{end+1} = Pf.(name)(Input_Cell{:});
end
```

Listing 11: Defining next state variables.

```
for name = ⟨loop over sections⟩
    nr = NLP.G.nr.(name);
    nx = ⟨retrieve next nx value of section⟩
    if nx ~= 0
        T_next.(name) = MX.sym(['T_',char(name),'_',k_str], nr*nx);
        NLP.CLC.S.x.(name){end+1} = T_next.(name);
    end
end
```

Listing 12: Imposing soft constraints.

```
epsilon = MX.sym(['epsilon_',k_str], 1);
NLP.CLC.S.p.epsilon{end+1} = epsilon;
h_peak{end+1} = T_next.peak - NLP.constraint.lb_soft.peak + epsilon; % h_peak > 0;
h_peak{end+1} = NLP.constraint.ub_soft.peak - T_next.peak + epsilon; % h_peak > 0;
```

Listing 13: Solve for algebraic states at collocation points.

```
for d = 1:NLP.nlp.d
  ⟨construct Input_Cell{}⟩
    for name = AlgebraicNames
        g{end+1} = dT.(name)(Input_Cell{:});
    end
end
```

Listing 14: Defining next optimization variables.

```
for name = [BilletDependentNames AlgebraicNames NonBilletDependentNames]
    if ⟨T_next.(name) exists⟩
        T.(name) = T_next.(name);
    end
end

if ⟨not the last discretization point⟩
    v_prev = C.ram;
    for name = InputNames
        C.(name)                    = MX.sym([char(name),'_', k_str]);
        NLP.CLC.S.u.(name){end+1}  = C.(name);
    end
end

if ⟨not the last discretization point⟩
    for name = ParameterNames
        P.(name)                     = MX.sym([char(name),'_', k_str]);
        NLP.CLC.S.p.(name){end+1}   = P.(name);
    end
end
```

Listing 15: Impose constraints on steps in ram speed.

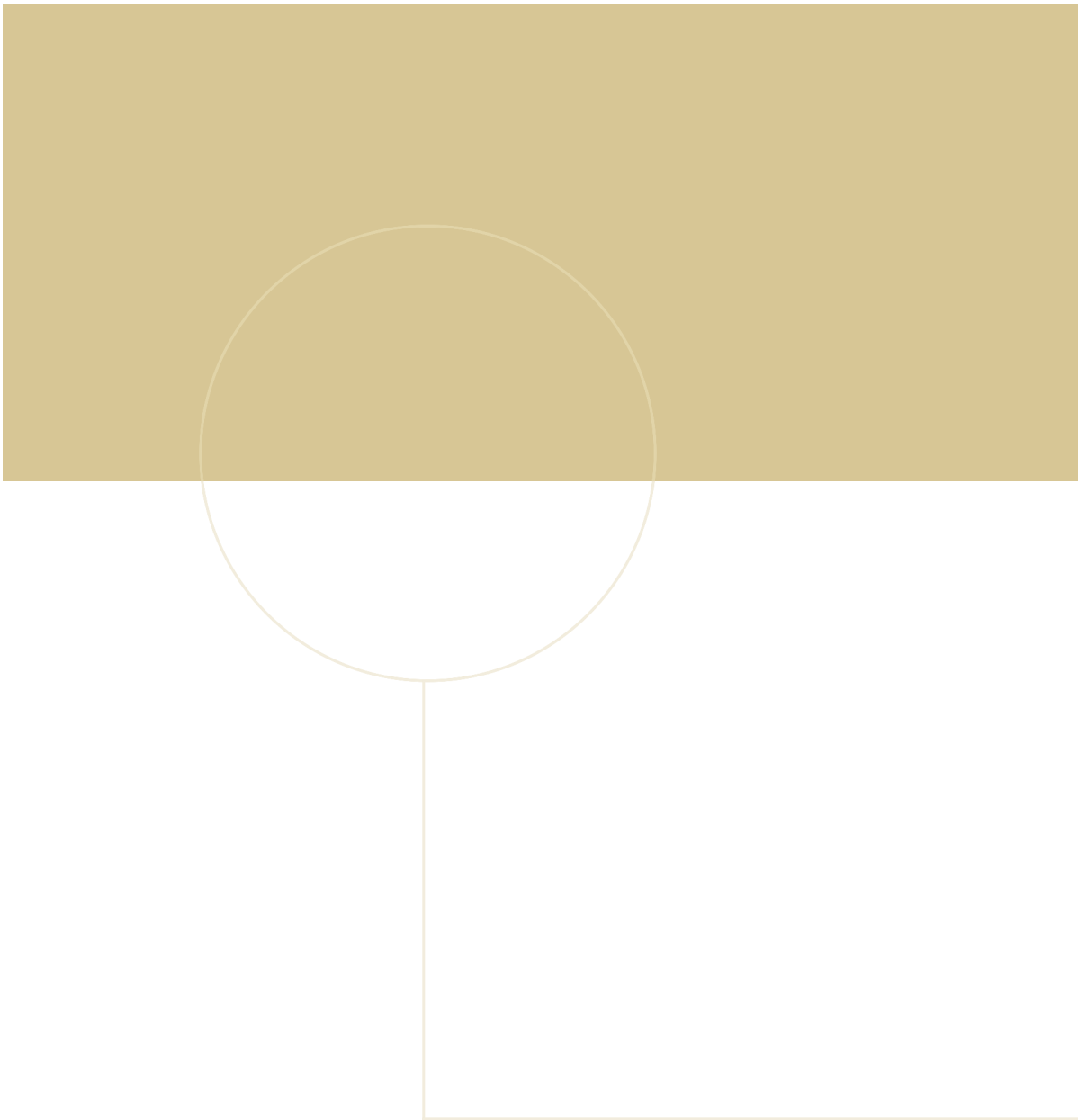
```
h_Dram{end+1} =  (C.ram - v_prev) + (NLP.nlp.dL/NLP.nlp.vtypical) * NLP.constraint.ub.Dram;
h_Dram{end+1} = -(C.ram - v_prev) + (NLP.nlp.dL/NLP.nlp.vtypical) * NLP.constraint.ub.Dram;
```

Listing 16: Solve for the algebraic states at the discretization points.

```
for name = AlgebraicNames
    g{end+1} = dT.(name)(Input_Cell{:});
end
```

Listing 17: Constructing *Input_Cell*{}.

```
Input_Cell = {};
for name = ⟨loop over relevant sections⟩ % This depends on the function the input cell if used for
    Input_Cell{end+1} = T.(name);
end
for name = ⟨loop over control variables⟩
    Input_Cell{end+1} = C.(name);
end
for name = ⟨loop over parameters⟩
    Input_Cell{end+1} = P.(name);
end
```



NTNU

Norwegian University of
Science and Technology