Applying feedback control to improve 3D printing quality

D. Gootjes

Master of Science Thesis

Delft University of Technology

Delft Center for Systems and Control

Applying feedback control to improve 3D printing quality

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

D. Gootjes

August 8, 2017

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

TU Delft Delft University of Technology

DCSC

Table of Contents

Acknowledgements **vii**

1 Introduction **1**

1-1 Problem statement 2

1-2 Thesis outline 3

2 Large-scale 3D printing **5**

2-1 Basics of thermoplast 3D printing 5

2-1-1 High quality 3D printing 6

2-2 Extrusion basics 8

2-2-1 First principle thermal modeling 8

2-3 Summary 10

3 System identification theory and practice **11**

3-1 Linear system identification 11

3-1-1 Output-error parametric model estimation 12

3-1-2 Subspace identification 14

3-1-3 Input signal selection 16

3-2 Large-scale 3D printer modeling and system identification 17

3-2-1 Exploratory step response experiment 17

3-2-2 MIMO thermal model identification 18

3-3 Extrusion pressure model identification 26

3-4 Summary 29

4 Modeling the volumetric output flow rate 31

4-1 Fluid dynamics introduction . . . 31

4-1-1 Laminar steady-state pipe flow of a newtonian fluid . . . 33

4-1-2 Laminar steady-state pipe flow of a non-newtonian fluid . . . 34

4-2 Computational fluid dynamics . . . 35

4-2-1 ANSYS Fluent Academic Research version 18.0 simulation . . . 36

4-2-2 Flow rate modeling using system identification . . . 37

4-3 Summary . . . 41

5 Model predictive control 43

5-1 Performance indices . . . 44

5-2 Constraints . . . 45

5-3 Optimization . . . 46

5-4 Tuning a model predictive controller . . . 47

5-5 Kalman filtering . . . 47

6 Simulation study 49

6-1 The flow rate reference tracking problem . . . 50

6-1-1 Translating the volumetric reference flow rate to its corresponding pressure reference . . . 51

6-2 Model predictive controller synthesis . . . 52

6-3 Reference tracking simulation . . . 53

6-4 Model predictive control for temperature regulation . . . 58

6-5 Discussion . . . 60

7 Conclusions and recommendations 61

7-1 Recommendations for future work . . . 61

A Glossary 67

List of Acronyms . . . 67

List of Figures

1-1 Schematic of two additive manufacturing processes 2

2-1 Additive manufacturing flowchart 5
2-2 Layer building visualised. 6
2-3 Extruder setup 9
2-4 Thermal cell model. Adapted from [1] 10

3-1 A selection of suitable input signals 16
3-2 Block diagram of the total large-scale 3D printing system 17
3-3 Exploratory step response 18
3-4 Singular values belonging to the input-output data of heater one and zone one . 19
3-5 Singular values belonging to the input-output data of the MIMO experiment . . 19
3-6 Zone 1 temperature response and heater signal 20
3-7 Zone 2 temperature response and heater signal 21
3-8 Zone 3 temperature response and heater signal 21
3-9 Zone 4 temperature response and heater signal 21
3-10 Hose temperature response and heater signal 22
3-11 Nozzle temperature response and heater signal 22
3-12 Motor input signal 22
3-13 Zone 1 temperature response and heater signal of the validation dataset 24
3-14 Zone 2 temperature response and heater signal validation dataset 24
3-15 Zone 3 temperature response and heater signal validation dataset 24
3-16 Zone 4 temperature response and heater signal validation dataset 25
3-17 Hose temperature response and heater signal validation dataset 25
3-18 Nozzle temperature response and heater signal validation dataset 25
3-19 Linear SISO pressure model simulation and identification data 26

3-21 Singular values of the pressure identification data 27
3-20 Wiener model structure 27
3-22 Identification data simulation of the pressure at zone four 28
3-23 Strictly increasing piecewise linear function used as static output nonlinearity . . 28
3-24 Validation data simulation of the pressure at zone four 29

4-1 Fluid element free body diagram, adapted from [2] 32
4-2 Fully-developed laminar pipe flow for a newtonian fluid, adapted from [2] 34
4-3 Non-newtonian fluid laminar pipe flow velocity profile 35
4-4 Partial heated hose geometry with mesh 37
4-5 Flow velocity profile comparison of the analytical and CFD method for steady-state input pressures 38
4-6 Unsteady flow rate simulation 39
4-7 Singular values of the pressure-flow-data indicate a first-order model 40
4-8 Comparison of true volumetric output flow with linear SISO model simulation . . 40
4-9 Comparison of true volumetric output flow with the wiener model simulation . . 41

5-1 Receding horizon principle, adapted from [3] 44

6-1 Basic flow reference signal 50
6-2 Proposed flow rate reference signals 51
6-3 Block diagram describing the effect of the motor input on pressure and output flow rate 52
6-4 Linear Model predictive controller 53
6-5 Linear pressure ramp reference tracking simulation 54
6-6 Linear pressure exponential reference tracking simulation 54
6-7 Linear pressure sigmoid reference tracking simulation 55
6-8 Pressure reference tracking simulation results 56
6-9 Flow rate reference tracking simulation results 57
6-10 Motor disturbance signal used for simulation 58
6-12 Zoomed in temperature disturbance 59
6-11 Temperature simulation 59

List of Tables

1-1 Large-scale 3D printer specifications 2

2-1 Symbol clarification 9

3-1 Identification data model fit 23
3-2 Validation data model fit 23

4-1 List of symbols used in this section 33
4-2 Table showing the error between the analytical solution and CFD solution to the steady-state non-newtonian pipe flow problem for various pressures 37
4-3 VAF fit percentages for the linear model and wiener model 39

6-1 Tuning parameters for the synthesized model predictive controller 53
6-2 NRMSE values for each reference signal 57

Acknowledgements

I would like to thank my supervisor dr.ir. T.J.J. van den Boom for his assistance during the writing of this thesis. Many thanks go to dr.ir. E. de Vlugt for the supervision provided during the practical work and experiments.

Finally, I would like to thank everyone at HB|3D for providing me with the time and resources to do experiments on their large-scale 3D printer.

Delft, University of Technology
August 8, 2017

D. Gootjes

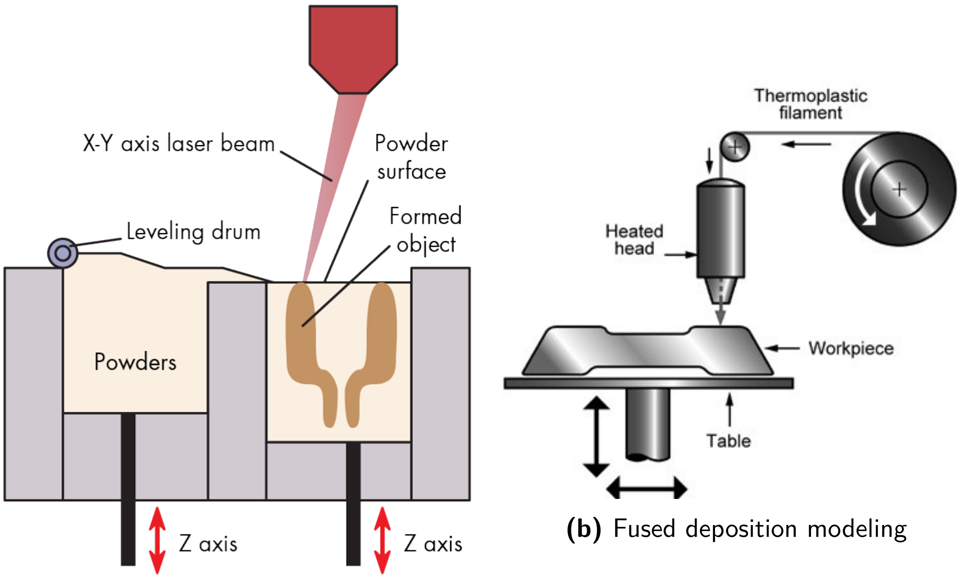
Chapter 1

Introduction

In the last three decades additive manufacturing (AM), also called 3D printing, has gained widespread attention in industry due to the possibility to revolutionize manufacturing [4]. AM creates an object by adding material, contrary to subtractive manufacturing processes such as milling and drilling, which shape an object by removing material. Some of the advantages of AM are that small batches are economically feasible, design changes can be easily and quickly implemented, and less material is needed compared to subtractive manufacturing methods.

AM refers to multiple techniques such as selective-laser-sintering, fused deposition modeling (FDM), and stereolithography. Selective-laser-sintering uses a powder bed as building material and a laser to sinter the material. After sintering one layer the powder bed is lowered and a layer of fresh powder is deposited and smoothed out. The fused deposition modeling process uses a thermoplast as the building material, and liquefies it to deposit layers of molten thermoplast on top of eachother. Stereolithography uses a photopolymer which solidifies after being exposed to ultra-violet light. The stereolithography process is schematically identical to selective-laser-sintering, but instead of a laser, ultraviolet light is used, and instead of powder a photopolymer. See figure 1-1.

The physical processes are different, but the idea is the same for all techniques: build a three-dimensional object by stacking contour slices on top of each other. Arguably the greatest benefit that AM brings to society is its customized use in the medical industry. For example inner ear hearing aids, knee replacements, skull parts, and hip replacements can be made using AM techniques [5] [6].



(a) Selective laser sintering

(b) Fused deposition modeling

Figure 1-1: Schematic of two additive manufacturing processes

The potential revolution of AM has motivated the start-up research and development company HB|3D to develop a large-scale 3D printer using a six degrees of freedom (DoF) industrial robot arm, a FANUC S-420iF, combined with a thermoplastic extrusion machine and a heated industrial hose. See table 1-1 for some specifications regarding the robot arm and heated industrial hose.

Table 1-1: Large-scale 3D printer specifications

FANUC S420-iF parameters	**Value**	**Heated hose parameters**	**Value**
Maximum reach	2.40 [m]	Maximum pressure	250 [bar]
Load capacity	120 [kg]	Maximum temperature	230 [°C]
Repeatability	±0.4 [mm]	Inner diameter	6 [cm]
Weight	1500 [kg]	Lenght	5 [m]

1-1 Problem statement

One of the most important requirements is to be able to create 3D printed objects with high dimensional accuracy. At the present time there are multiple dimensional error sources. For example, the robot arm sometimes behaves in an undesirable manner when the printing path contains a 90° angle. Near the turning point the robot arm decelerates, stops momentarily, and then accelerates in the new direction, resulting in excess material deposition near the corners. This decreases the dimensional accuracy of the printed object. The same issue emerges when the robot arm repositions to a new layer height.

Because the robot arm is occasionally converted to function as a milling machine (by removing the extrusion nozzle and mounting a milling head), for which the accelerating and decelerating behaviour does not pose a problem, combined with the fact that the robot arm has a low repeatability error serves as motivation to make no attempt to change and improve the robot path following behaviour. Avoiding excess material deposition can therefore only be achieved by adjusting the volumetric output flow. It is necessary that the robot and extruder exchange information in a way such that robot arm movement and extrusion volumetric output flow can be coupled to continually achieve a desired deposition rate. Currently, no communication between the robot arm and extruder is realized; both act as independent units. Designing and building an integrated process control unit is beyond the scope of this thesis. However, a first step can be made by controlling the volumetric output flow. Unfortunately, due to time constraints, no sensor is available to measure the volumetric output flow. Fortunately, the pressure at the hose entrance is available for measurement, which can be influenced by changing the extrusion motor speed. If the effect of the input pressure on the volumetric flow through the hose can be modeled, regulating the volumetric output flow will be possible without measuring it by controlling the input pressure.

The problem statement is formulated as follows:

"Create a model which completely describes the extrusion dynamics and the effect of pressure changes on the volumetric output flow rate of a pseudoplastic fluid flowing through the heated hose. Use the model to design a model predictive controller which allows the volumetric output flow to follow a feasible reference trajectory within 1% accuracy".

Practical considerations

Due to time constraints no flow measurement device could be installed; the volumetric flow rate dynamics are hence modeled using a commercially available computational-fluid-dynamics (CFD) software package namely ANSYS Fluent Academic Research, Release 18.0. Furthermore, due to limited time available and technical difficulties it was not possible to implement the model predictive controller on the real system.

1-2 Thesis outline

In chapter 2 the basic notions of the large-scale 3D printer are presented. First, a description of the printing process is given and the term high quality is defined. Then, the extrusion process is under scrutiny; what happens during extrusion and which actuators and sensors are available for control. Lastly, a first principle thermal modeling approach from literature is discussed. Chapter 3 presents some theory on two system identification techniques, and shows modeling results from experiments done in practice. The identified models describe the relation between the heater inputs and extrusion temperature, and between motor input and pressure at zone 4. Chapter 4 discusses the specifics of the CFD model and presents results regarding the estimated simplified model. Chapter 5 discusses the basics of model predictive control (MPC). In chapter 6 a simulation study is performed to assess the flow rate reference tracking effectiveness of the model predictive controller for three proposed reference signals. Furthermore, the temperature reference tracking behaviour of the model predictive controller is simulated, and it is discussed how the effect of the motor input disturbance can be removed by adding a feedforward signal. Finally, chapter 7 summarizes the conclusions and presents some recommendations.

Chapter 2

Large-scale 3D printing

The aim of this chapter is to develop a basic understanding of the working principles of the large-scale 3D printing process. First, the entire printing process from pre-processing to post-processing will be discussed. Then, the meaning of high quality is defined, and possible error sources are identified. After that, an assessment will be made on which type of error sources can be remedied using system control techniques. Next, the available inputs and outputs of the 3D printer will be presented. Finally, to develop an understanding of how the physical parameters of the large-scale 3D printer influence the temperature dynamics, a first principle model will be discussed.

2-1 Basics of thermoplast 3D printing

There are three basic steps in creating a product from start to finish using a 3D printer, the pre-processing step, the printing step, and the post-processing step. During the pre-processing step the designer creates a virtual 3D model by using a computer aided design (CAD) program. The 3D CAD model is then virtually sliced in multiple horizontal layers of equal height. The obtained contours are then used by a path-planning-algorithm which calculates the input signals for the nozzle positioning actuators. For industrial robots, such as the FANUC S-420iF, the trajectory generated is translated to g-code which the robot translates to joint motor actuation. This concludes the pre-processing step; see figure 2-1 for a flowchart.

The next step consists of actually printing the object: the nozzle traverses the pre-planned trajectory while liquid thermoplast is simultaneously deposited. Small-scale 3D printers push

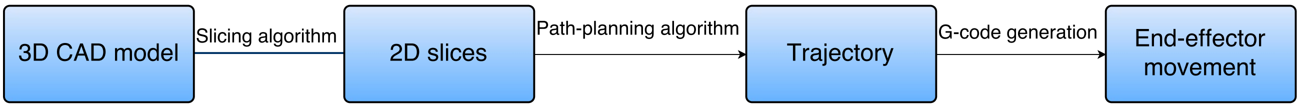


Figure 2-1: Additive manufacturing flowchart

a solid thermoplast filament through a heated nozzle, which then liquefies the thermoplast. The large-scale 3D printer uses an extruder in conjunction with a heated hose to supply the molten thermoplast. The contours are then stacked on top of eachother to create the object. See figure 2-2.

The final, optional, step is post-processing the printed object. This is done mostly to obtain a better surface finish. A multitude of options are available including, but not limited to:

- Sanding
- Painting
- Polishing
- Coating or gap filling

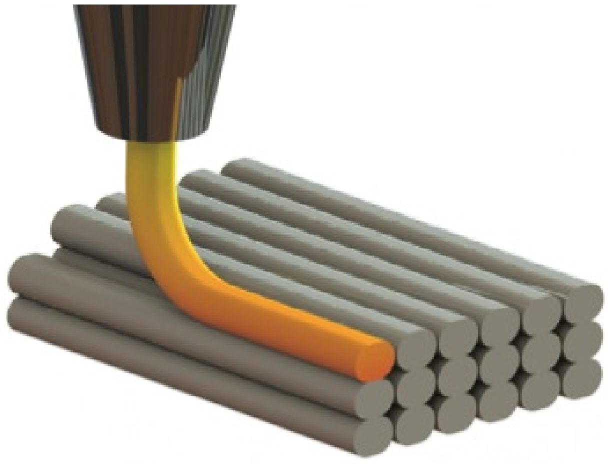


Figure 2-2: Layer building visualised.

This thesis will focus on improving the actual 3D printing step. Therefore discussion of pre-processing and post-processing will be omitted.

In the next section the notion of high quality is defined and possible error sources will be discussed. After the discussion a choice will be made on what sources of error can be mitigated using control engineering techniques.

2-1-1 High quality 3D printing

Because the term high quality is very broad and has different meanings depending on what the function of the object is, it needs to be further specified. According to the literature review [7] the main quality characteristics are:

- Surface roughness

- Mechanical properties
- Dimensional accuracy

A high quality product means it has low surface roughness, good mechanical properties, and a high dimensional accuracy. Each characteristic, along with methods on how to improve them, will be briefly discussed next.

Surface roughness

The surface roughness of a 3D printed product is determined by several factors, including but not limited to: layer thickness, layer width, and deposition speed. Low surface roughness is preferred as it is considered more aesthetically pleasing. In general the relationships hold that higher thermoplast temperatures and smaller layer thicknesses decrease the surface roughness, with the layer thickness being the dominant factor [8] [9] [10].

Mechanical properties

The mechanical properties of thermoplastic 3D printed product are mainly influenced by the layer to layer bonding and layer build orientation. In general a higher thermoplast temperature increases the bond strength between layers, which improves the mechanical properties of the printed product. The tensile strength is largest in the direction of the printed layers. Therefore, if it is known a priori in which direction a part will experience a force, the layers should be printed in the same direction [11] [12].

Dimensional accuracy

An analysis of dimensional accuracy is done by identifying possible error sources. The errors occur at three different steps of the printing process: before material depositioning, during material depositioning, and after material deposition. The errors occurring before the material depositioning are the result of incorrect nozzle trajectory generation or CAD model slicing height, which both are pre-processing steps and therefore will not be further discussed. Errors that occur after material deposition are the result of thermal shrinkage. To mitigate thermal shrinkage the thermoplast temperature can be lowered, or a material with a lower thermal expansion coefficient can be used [13] [14].

The errors that occur during the printing process can be divided in two distinct sources: nozzle positioning error, and volumetric flow errors. The nozzle positioning errors are the result of incorrect positioning by the FANUC s420-iF robot arm. Because the robot arm has a repeatability of ±0.4 [mm] it is deemed accurate enough. Therefore the effect of the robot arm positioning on dimensional accuracy will be disregarded for the remainder of this thesis. The steady-state volumetric flow rate of a nonnewtonian fluid (see chapter 4 for details) flowing through a circular nozzle is given by the following equation [15]:

$$Q = \pi \cdot \left(\frac{1}{2}\frac{\Delta P}{c_m(T_t)L}\right)^{\frac{1}{n}} \left(\frac{n}{3n+1}\right) R^{\left(\frac{3n+1}{n}\right)} \tag{2-1}$$

The only two non-static parameters influencing the flow rate Q are the consistency index c_m, which depends on the thermoplast temperature T_t, and the pressure difference ΔP. Furthermore, when the robot arm approaches a corner or is about to move to a new layer it decelerates to a full stop before accelerating in a new direction. If the volumetric flow rate is not decreased and increased during deceleration and acceleration respectively, an excess of material is deposited which decreases the dimensional accuracy. To eliminate both error sources the volumetric flow rate needs to be controlled. Which in turn means that the pressure and temperature of the extruder need to be controlled.

In summary the only quality characteristic which can be influenced by aplying techniques from control engineering is the dimensional accuracy; more specifically the dimensional errors generated during material depositioning. The next section discusses the basics of the extrusion process and identifies which inputs and outputs are available to control the extrusion pressure and temperature.

2-2 Extrusion basics

The extruder globally consists of five parts: the hopper, the screw, the barrel, the hose, and the nozzle. Small plastic pellets located at the hopper are fed to the screw, which pushes them through the barrel towards the hose. The hose is used to connect the extruder to the nozzle where the material deposition takes place. There are six heaters mounted to provide thermal energy to the process; four are located around the barrel zones, one is located across the entire length inside the hose, and the final one is located at the nozzle. An AC motor is used to drive the screw rotation speed. The pressure in the extruder is generated by the conic shape of the screw, and is the largest at zone four. Because the heated hose can only withstand pressures up to 250 [bar], the pressure is measured at zone four. As stated previously the pressure difference between zone four and the outside world determines the volumetric flow rate. See figure 2-3 for a complete picture of the extruder, heated hose, and nozzle. As stated previously the goal is to control the temperature and pressure of the extruder. The first step to control a process is by modeling it. Using the principle of conservation of energy the effect of the heaters on the temperature is modeled in the next section. Deriving a first principles model of the effect of motor input on temperature and pressure is omitted due to the complex nature. Instead, the effect of motor input on pressure and temperature is obtained using system identification techniques, and is discussed in chapter 3 [16].

2-2-1 First principle thermal modeling

A small finite element first principle model is described in [1]. Each zone is modeled as a thermal cell which has a heat capacity and exchanges heat with its surroundings and neighboring thermal cells. The model is obtained by applying the law of conservation of energy to each cell combined with the laws of conduction, radiation, and convection. Thermal energy is supplied to each cell by its corresponding heater element. See figure 2-4 and table 2-1.

This approach will be applied to the extruder with hose and nozzle, with the assumption that the hose and nozzle are isolated thermal cells. This is because there is only a very small path

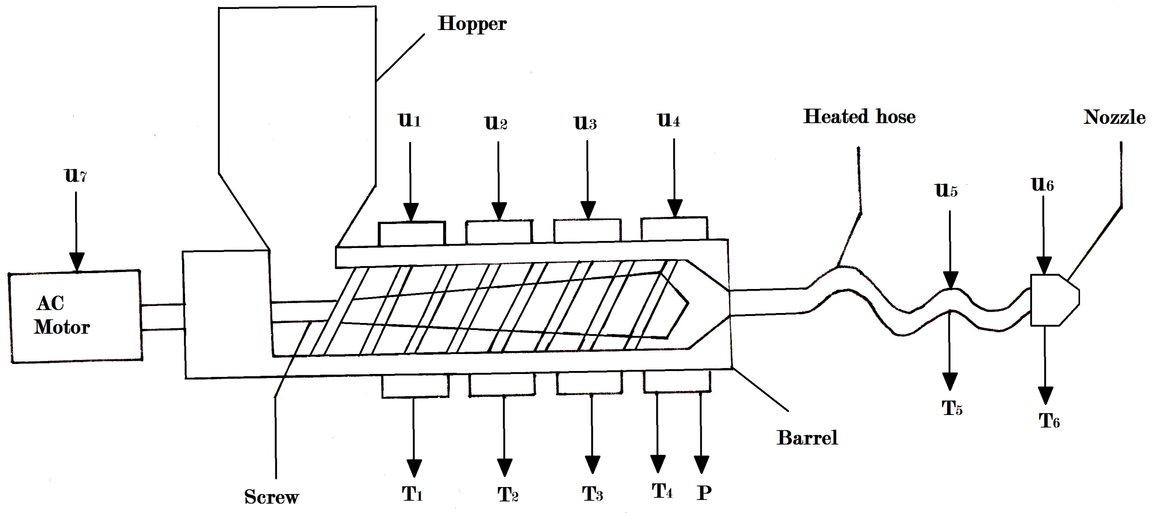


Figure 2-3: Extruder setup

Symbol	Units
T	Temperature [°C]
α	Heat radiation rate [W/°C]
β	Heat transfer rate [W/°C]
q	Heater input [W]
C	Heat capacity [J/°C]

Table 2-1: Symbol clarification

of thermal conduction between hose/nozzle and its neighbouring thermal cells.

The equations resulting from applying the principle of conservation of energy combined with the laws of conduction, radiation, and convection for the kth thermal cell with a total of n thermal cells can be written as:

For k = 1

$$\dot{T}_1 = -\frac{\alpha_1 + \beta_1}{C_1} T_1 + \frac{\beta_1}{C_1} T_2 + \frac{q_1}{C_1} \tag{2-2}$$

For (1 < k < n):

$$\dot{T}_k = \frac{\beta_{k-1}}{C_k} T_{k-1} - \frac{\alpha_k + \beta_{k-1} + \beta_k}{C_k} T_k + \frac{\beta_k}{C_k} T_{k+1} + \frac{q_k}{C_k} \tag{2-3}$$

And for k = n:

$$\dot{T}_n = \frac{\beta_{n-1}}{C_n} T_{n-1} - \frac{\alpha_n + \beta_{n-1}}{C_n} T_n + \frac{q_n}{C_n} \tag{2-4}$$

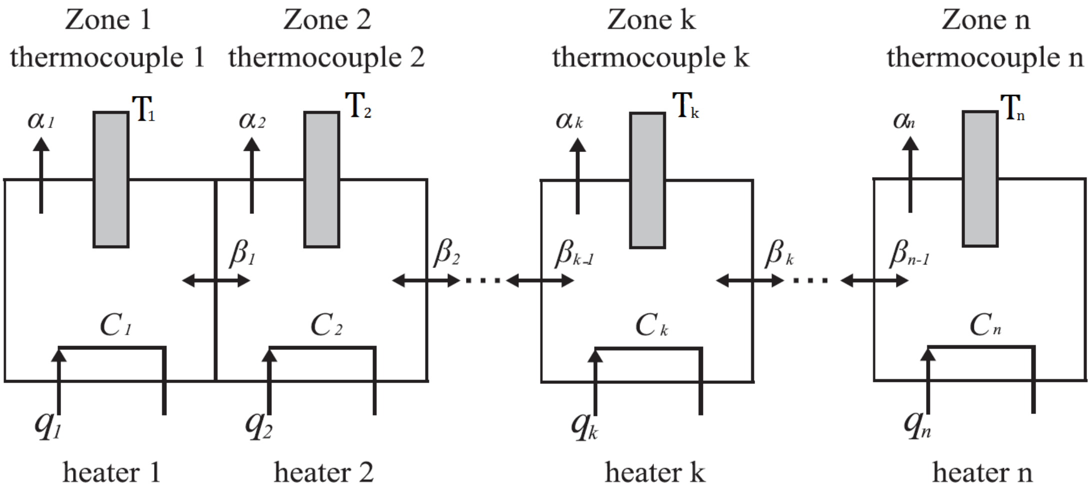


Figure 2-4: Thermal cell model. Adapted from [1]

Which can be written for the extruder with hose and nozzle as the following state-space equation:

$$
\begin{bmatrix} \dot{T}_1 \\ \dot{T}_2 \\ \dot{T}_3 \\ \dot{T}_4 \\ \dot{T}_5 \\ \dot{T}_6 \end{bmatrix} =
\begin{bmatrix}
-\frac{(\alpha_1+\beta_1)}{C_1} & \frac{\beta_1}{C_1} & 0 & 0 & 0 & 0 \\
\frac{\beta_1}{C_2} & -\frac{(\alpha_2+\beta_1+\beta_2)}{C_2} & \frac{\beta_2}{C_2} & 0 & 0 & 0 \\
0 & \frac{\beta_2}{C_3} & -\frac{(\alpha_3+\beta_2+\beta_3)}{C_3} & \frac{\beta_3}{C_3} & 0 & 0 \\
0 & 0 & \frac{\beta_3}{C_4} & -\frac{(\alpha_4+\beta_3+\beta_4)}{C_4} & 0 & 0 \\
0 & 0 & 0 & 0 & -\frac{\alpha_5}{C_5} & 0 \\
0 & 0 & 0 & 0 & 0 & -\frac{\alpha_6}{C_6}
\end{bmatrix}
\begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \end{bmatrix}
+
\begin{bmatrix}
\frac{1}{C_1} & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{1}{C_2} & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{1}{C_3} & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{1}{C_4} & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{1}{C_5} & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{1}{C_6}
\end{bmatrix}
\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{bmatrix}
\tag{2-5}
$$

2-3 Summary

In this chapter the basics of the large-scale 3D printing process were discussed. It emerged that the dimensional errors generated during material depositioning are most suitable to be influenced by using techniques from control engineering. A first principles model of the effect of the heaters on the temperature was derived in terms of unknown system parameters. Since these parameters determine the behaviour of the system, they need to be estimated. Furthermore, the effect of the motor input on the temperature is not modeled and no first principle model relating the motor input to the pressure was derived. This motivates the need for applying techniques from system identification, which will be discussed in the next chapter.

Chapter 3

System identification theory and practice

In order to control the outputs of a system it is needed to predict how the system outputs react to inputs. Something which describes the input-output behaviour of a system is generally called a model and can be obtained in various ways. For example, a model can be created by applying first principles such as conservation of energy, conservation of momentum, etc. However, this will only yield a useful model when all relevant parameters describing the system behaviour are known a priori with sufficient accuracy. This is rarely the case. Fortunately, there is another approach to obtain a system model. By exciting the system using its available inputs and recording the output responses a system model can be estimated. This approach is called system identification and will be the topic of this chapter.

Two linear system identification methods, namely the output-error method and the subspace method, will be discussed first. Choosing suitable input signals is discussed briefly, after which both system identification techniques are applied consecutively to selected input-output datasets describing the extruder, hose, and nozzle temperature response to heater and motor inputs. A model describing the relation between the pressure at zone four and the motor input signal will be identified as well. Obtaining a model which describes the volumetric output flow subject to the pressure at zone four involves using computational fluid dynamics (CFD) techniques and is covered in chapter 4.

3-1 Linear system identification

Both methods are restricted to the case of linear system identification. This means that it is assumed that the system can be described by a (discrete) linear time-invariant (LTI) state-space model, see equation 3-1. The model is assumed discrete because the 3D printer is driven by a microcontroller, which makes the system inherently sampled.

$$
\begin{aligned}
x(k+1) &= Ax(k) + Bu(k) \\
y(k) &= Cx(k) + Du(k)
\end{aligned}
\tag{3-1}
$$

The basic system identification idea is to deduce the system matrices (A, B, C, D), which completely specify the system's dynamical behaviour, from recorded inputs, u, and outputs, y. To increase the probability of correctly identifying a model, prior information regarding the system is needed; as discussed in [17] it is generally accepted that incorporating physical insights in the system identification procedure enhances the model quality. This is due to the fact that different models can be fit to input-output data, and restrictions on the possible model structures increases the likelyhood of obtaining a 'good' model. Here 'good' means that the model not only fits the estimation data, but validation data as well. The system identification procedure is therefore iterative; by doing preliminary experiments insights in the system behaviour are obtained which allows for the design of better future experiments. In the next two subsections the output-error method and the subspace method are explained.

3-1-1 Output-error parametric model estimation

The output-error parametric model estimation problem consists globally of the following four steps:

1. Parametrize the model
2. Formulate the estimation objective as an optimization problem
3. Numerically solve the optimization problem
4. Evaluate the accuracy of the obtained model

Parametrizing the model

To parametrize the model, assume that the entries of the system matrices (3-1) depend on a parameter vector θ. The objective is then to estimate this vector in such a way that the 'best' model is obtained. The parametrized model becomes

$$
\begin{aligned}
\hat{x}(k+1) &= A(\theta)\hat{x}(k,\theta) + B(\theta)u(k) \\
\hat{y}(k,\theta) &= C(\theta)\hat{x}(k,\theta) + D(\theta)u(k)
\end{aligned}
\tag{3-2}
$$

One particular way to parametrize the model is to apply the relevant laws of physics and estimate the unknown physical constants. For example, in section 2-2-1 a model parametrization derived from first principles is presented with unknown parameters. However, the model parametrization is not of particular importance as long as the input-output relation is adequately described. Therefore the main objective is to choose a parametrization which allows numerically stable estimation of the parameter vector. Discussion of certain parametrizations which have good numerical properties is beyond the scope of this thesis; knowing that they exist is deemed sufficient.

Formulate the estimation problem as an optimization problem

At this point the linear time-invariant model is parametrized. The next step is to estimate the parameter values. To do this the parameters are stacked in a vector θ, and the objective is to search for the vector θ that fits the input-output data in the 'best' way. One way to quantify the parametrized model 'goodness' is by using a quadratic cost function which penalizes the output error $(y - \hat{y})$. The output error is calculated for a specific choice of θ; see equation 3-3. The choice for the quadratic cost function is motivated by the fact that for each value of θ the cost is larger than or equal to zero, and that larger errors are penalized more heavily than small errors. The optimal parameter vector $\hat{\theta}$ minimizes the given cost function.

$$J_N(\theta) = \frac{1}{N} \sum_{k=0}^{N-1} ||y(k) - \hat{y}(k,\theta)||^2 \tag{3-3}$$

Numerically solving the optimization problem

The optimization problem can be solved by initializing a vector $\theta^{(i)}$ and updating it based on the computed Jacobian and Hessian functions. The cost function $J_N(\theta)$ is expanded as a Taylor series around $\theta^{(i)}$, where $\theta^{(i)}$ is the initial parameter estimation vector. The Taylor series expansion is given by:

$$\begin{aligned} J_N(\theta) =& J_N(\theta^{(i)}) + (J_N^{'}(\theta^{(i)}))^T(\theta - \theta^{(i)}) \\ &+ \frac{1}{2}(\theta - \theta^{(i)})^T J^{''}(\theta^{(i)})(\theta - \theta^{(i)}) + \text{higher order terms} \end{aligned} \tag{3-4}$$

Where $J_N^{'}(\theta^{(i)})$ is the Jacobian and $J_N^{''}(\theta^{(i)})$ is the Hessian. The higher order terms can be neglected if the initial parameter vector is close to the optimum. The necessary condition for minimizing the approximation of $J_N(\theta)$ then becomes

$$J_N^{'}(\theta^{(i)}) + J^{''}(\theta^{(i)})(\theta - \theta^{(i)}) = 0 \tag{3-5}$$

If the Hessian is invertible then the parameter vector can be updated as

$$\theta = \theta^{(i)} - J_N^{''}(\theta^{(i)})^{-1} J_N^{'}(\theta^{(i)}) \tag{3-6}$$

This method is called the Newton method [18]. The method does not always perform adequately because the inversion of the Hessian can become ill-conditioned. Other nonlinear optimization methods such as quasi-Newton methods and line-search methods can be used [19], but discussion of these methods is outside the scope of this thesis.

Model accuracy evaluation

Once an optimal parameter vector $\hat{\theta}$ is estimated it is needed to evaluate the correctness of the model. Evaluating the correctness, or accuracy, of the obtained model can be done by using a seperate validation dataset. The validation input signal is used to simulate the model's response and compared with the measured output. The criteria that will be used to quantify the 'goodness' of fit is the variance-accounted-for (VAF). It represents how much of the estimated signal variance is explained by the model. See equation 3-7.

$$fit_{VAF} = 100\left(1 - \frac{var(y-\hat{y})}{var(y)}\right) \tag{3-7}$$

This concludes the discussion on the output-error method. The motivation to explore the subspace identification method is that two of the main drawbacks of the output-error method are that the model order needs to be specified a priori, and that good results are only obtained if the initial parameter vector $\theta^{(i)}$ is close to the optimal value [18]. Subspace identification methods points to the model order in an elegant way, and can provide good initial estimates which can be refined using the output-error method.

3-1-2 Subspace identification

The second model estimation method to be discussed is the subspace identification method. The basic idea of subspace identification is that by storing input and output data in structured Hankel matrices (see equation 3-13) certain subspaces containing information related to the signal generating LTI state-space model (3-1) can be retrieved.

Suppose that an experiment provided an input $u(k)$ and output $y(k)$ data sequence. The input-output data can then be structured in a matrix equation as given by (3-8)

$$\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ \vdots \\ y(s-1) \end{bmatrix} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{s-1} \end{bmatrix} x(0) + \begin{bmatrix} D & 0 & 0 & \cdots & 0 \\ CB & D & 0 & \cdots & 0 \\ CAB & CB & D & & 0 \\ \vdots & & \ddots & \ddots & \\ CA^{s-2}B & CA^{s-3}B & \cdots & CB & D \end{bmatrix} \begin{bmatrix} u(0) \\ u(1) \\ u(2) \\ \vdots \\ u(s-1) \end{bmatrix} \tag{3-8}$$

Which can be written more compactly as:

$$Y_0 = \mathcal{O}_s x(0) + \mathcal{T}_s U_0 \tag{3-9}$$

It is possible to relate time-shifted versions of the state, input, and output vectors to the same matrices $\mathcal{O}_s$ and $\mathcal{T}_s$:

$$\begin{bmatrix} y(k) \\ y(k+1) \\ \vdots \\ y(k+s-1) \end{bmatrix} = \mathcal{O}_s x(k) + \mathcal{T}_s \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+s-1) \end{bmatrix} \tag{3-10}$$

Now, equations 3-9 and 3-10 can be combined, depending on the number of available input-output samples, to form the data equation:

$$Y_{0,s,N} = \mathcal{O}_s X_{0,N} + \mathcal{T}_s U_{0,s,N} \tag{3-11}$$

Where

$$X_{i,N} = \begin{bmatrix} x(i) & x(i+1) & \cdots & x(i+N-1) \end{bmatrix} \tag{3-12}$$

And the block hankel matrices $Y_{i,s,N}$ and $U_{i,s,N}$ are given by:

$$Y_{i,s,N} = \begin{bmatrix} y(i) & y(i+1) & \cdots & y(i+N-1) \\ y(i+1) & y(i+2) & \cdots & y(i+N) \\ \vdots & \vdots & & \vdots \\ y(i+s-1) & y(i+s) & \cdots & y(i+N+s-2) \end{bmatrix} \tag{3-13}$$

$$U_{i,s,N} = \begin{bmatrix} u(i) & u(i+1) & \cdots & u(i+N-1) \\ u(i+1) & u(i+2) & \cdots & u(i+N) \\ \vdots & \vdots & & \vdots \\ u(i+s-1) & u(i+s) & \cdots & u(i+N+s-2) \end{bmatrix} \tag{3-14}$$

By constructing a matrix orthogonal to $U_{0,s,N}$, denoted by $\Pi^{\perp}_{U_{0,s,N}}$, and right multiplying equation 3-11 with it results in:

$$Y_{0,s,N}\Pi^{\perp}_{U_{0,s,N}} = \mathcal{O}_s X_{0,N}\Pi^{\perp}_{U_{0,s,N}} \tag{3-15}$$

The column space of $Y_{0,s,N}\Pi^{\perp}_{U_{0,s,N}}$ is equal to the column space of $\mathcal{O}_s$. Therefore, if the column space of $Y_{0,s,N}\Pi^{\perp}_{U_{0,s,N}}$ is known, then the column space of matrix $\mathcal{O}_s$ is known, and estimates for C and A can be extracted. To find the column space of $Y_{0,s,N}\Pi^{\perp}_{U_{0,s,N}}$ compute a singular-value-decomposition of the matrix:

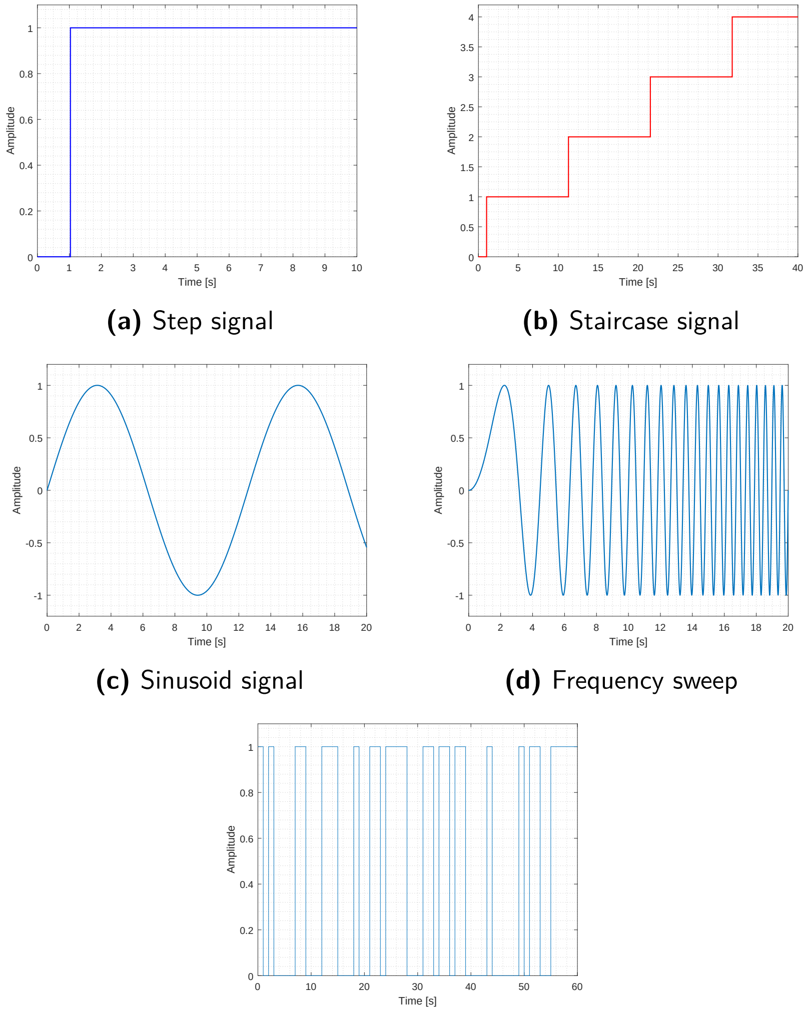
$$Y_{0,s,N}\Pi^{\perp}_{U_{0,s,N}} = U_n \Sigma_n V_n^T \tag{3-16}$$

The rank of Σ_n determines the rank of the system. Suppose that rank(Σ_n) = n, then U_n can be denoted as

$$U_n = \mathcal{O}_s T = \begin{bmatrix} CT \\ CT(T^{-1}AT) \\ \vdots \\ CT(T^{-1}AT)^{s-1} \end{bmatrix} = \begin{bmatrix} C_T \\ C_T A_T \\ \vdots \\ C_T A_T^{s-1} \end{bmatrix} \tag{3-17}$$

Where T is a similarity transformation matrix. Consequently, the estimated matrices C_T and A_T can be extracted from U_n. The magnitude of the singular values give an estimate of the rank of the system matrices, and hence of the model order. By comparing the magnitudes of the singular values and counting the number of singular values with significant magnitude, the model order needed to capture the system dynamics is uncovered. What remains is estimating B_T, D_T, and $x_T(0)$. A similar approach as outlined above can be used. Another way is to solve for B_T, D_T, and $x_T(0)$ by recasting the problem in a least-squares setting as a parametric output-error problem and solving for θ, as discussed in section 3-1-1.

In practice both system identification methods are used consecutively. First an initial model is estimated using subspace identification, since it elegantly provides the model order. Then, an output-error prediction method is used to improve the estimated model's predictive ability [18].



(a) Step signal

(b) Staircase signal

(c) Sinusoid signal

(d) Frequency sweep

(e) Generalized binary noise signal

Figure 3-1: A selection of suitable input signals

3-1-3 Input signal selection

One might ask what type of input signals should be used to excite the system in order to capture relevant dynamics. In practice different types of input signals are used to gather information about the system. Examples are: a step signal, staircase signal, sinusoid, frequency sweep or generalized-binary-noise (GBN) sequence; see figure 3-1. In order to succsesfully identify a system model, the inputs used to excite the system should be persistently exciting. This means that the input Hankel matrix $U_{0,n,N}$ (3-18) has full row rank n.

$$U_{0,n,N} = \begin{bmatrix} u(0) & u(1) & \cdots & u(N-1) \\ u(1) & u(2) & \cdots & u(N) \\ \vdots & \vdots & & \vdots \\ u(n-1) & u(n) & \cdots & u(N+n-2) \end{bmatrix} \tag{3-18}$$

Essentially it does not matter what type of signal is used, as long as this condition is satisfied. One approach is to generate an input signal and check a posteriori if it is persistently exciting. Since the system is sampled and the heater input signals can only take discrete values between 0 and 1 it is natural to use a GBN signal with varying amplitude to excite the system.

3-2 Large-scale 3D printer modeling and system identification

Two system identification techniques and input signal selection have been discussed; so now it is necessary to choose which input-output pairs are to be modeled. As discussed previously the total system model to be identified is the large-scale 3D printer. It consists of a robot arm for nozzle positioning and an extruder combined with a heated hose to supply molten thermoplast material. As mentioned previously modeling and controlling the robot arm will be omitted due to the fact that the positioning is already highly accurate (it has a repeatability ±4 [mm]). To accurately print large-scale products both the temperature and the volumetric output flow must be modeled and controlled. This motivates the identification of the following input-output models (see figure 3-2):

1. Barrel, hose, and nozzle temperature responce to heater power and motor speed
2. Pressure at zone four subject to the motor speed
3. Hose volumetric output flow subject to the pressure at zone four

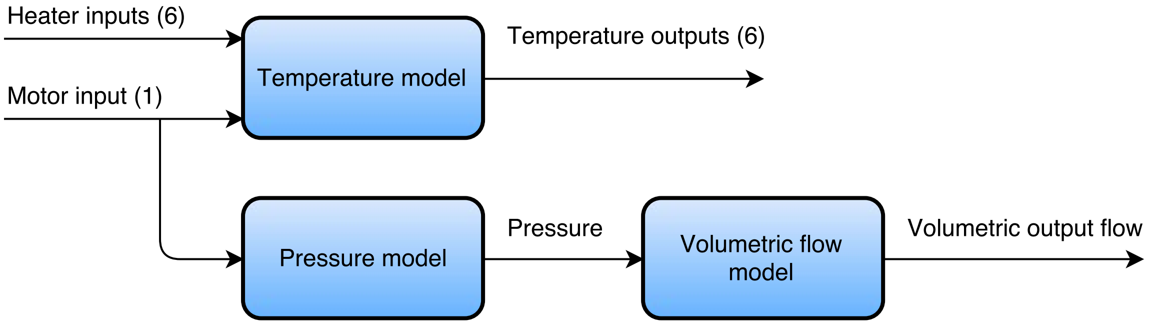


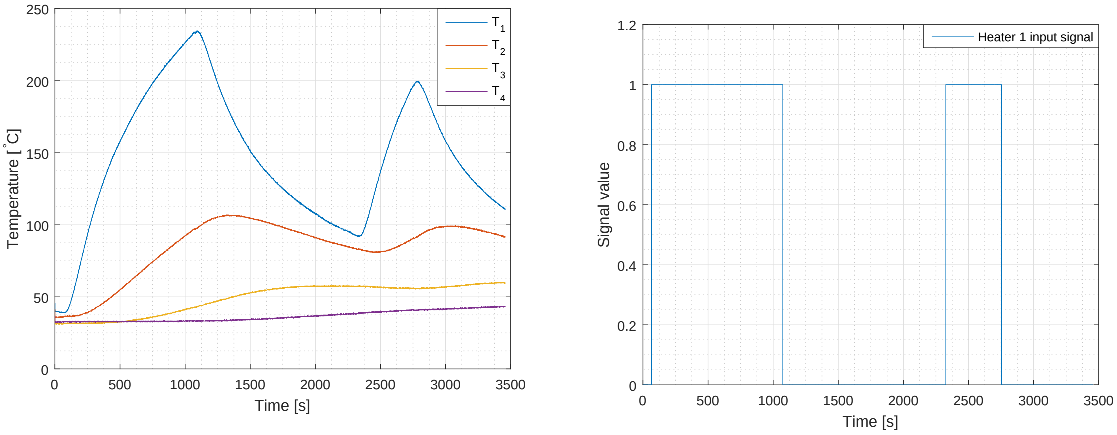
Figure 3-2: Block diagram of the total large-scale 3D printing system

In chapter 2 it is suggested that there is some coupling between the temperature responses of adjacent thermal cells. Since the barrel consists of one metal pipe it seems likely that the temperature responses are coupled. This assumption is tested by applying a step response on the first heater and recording the temperatures of the barrel (zone one through four). Next, a multiple-input-multiple-output (MIMO) experiment is performed where each heater is driven by a different variable-amplitude GBN signal.

3-2-1 Exploratory step response experiment

In figure 3-3a the temperature response of zones one through four to the input (see figure 3-3b) can be seen. Clearly, heater one has the largest effect on the temperature of zone

one. Through conduction the temperature of zone two is raised, which in effect raises the temperature of zone three, which in turn increases the temperature of zone four.



(a) Thermal response of zone one, two, three, and four

(b) Heater one step input signal

Figure 3-3: Exploratory step response

By looking at the graph it can be concluded that there is coupling between adjacent zones; the assumption made in chapter 2 is confirmed. It is also interesting to look at what model order describes one zone sufficiently. As described in section 3-1-2 the singular values of the output hankel matrix right multiplied with a matrix orthogonal to the input hankel matrix indicate the model order. See figure 3-4 for the singular values corresponding to the input-output data of heater one and zone one. The singular values indicate that a first-order model will adequately capture the dynamics. Therefore the MIMO model structure will be sixth order with coupling terms between the adjacent temperature zones for the barrel, just as described by the first principles model in chapter 2.

3-2-2 MIMO thermal model identification

The MIMO experiment consists of two parts. In the first part the heaters are driven by GBN signals. Once the zone temperatures are high enough the motor speed is varied; this is the second part. It is necessary to preheat the extruder before starting the motor because extrusion at low temperature is not possible. To prevent technical difficulties such as memory problems regarding storing signal values and unexpected computer crashes, the motor input signal is manipulated by an operator and is therefore not a GBN signal. Inspection of each heater signal and motor signal shows that they are all persistently exciting.

After the data is recorded from the experiment a model can be estimated. The singular values of the MIMO data can be seen in figure 3-5; they indicate, as expected, that a sixth order model suffices. The first principle model described in section 2-2-1 is used as parametrization for the system identification procedure, with an extra input signal added to capture the influence of the motor input. The estimated sixth order state-space model (recall that the general discrete linear time-invariant model description is given by equation 3-1) is described

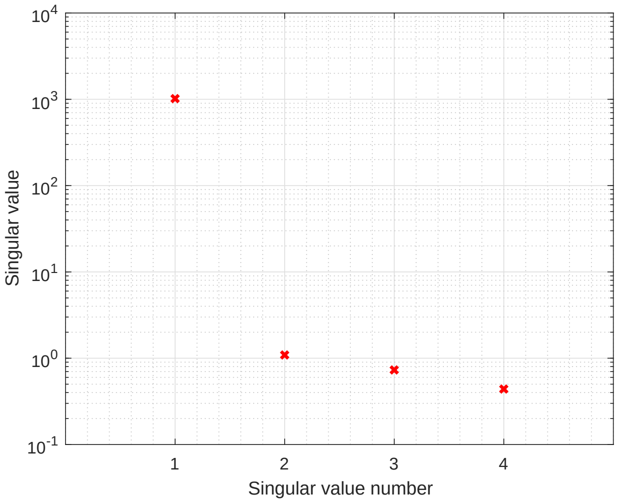


Figure 3-4: Singular values belonging to the input-output data of heater one and zone one

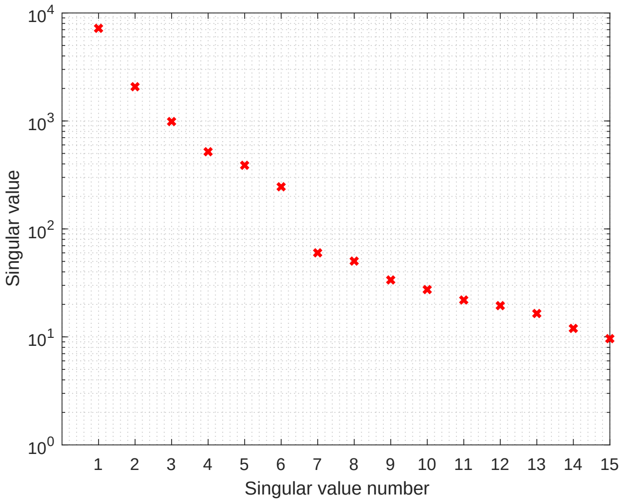


Figure 3-5: Singular values belonging to the input-output data of the MIMO experiment

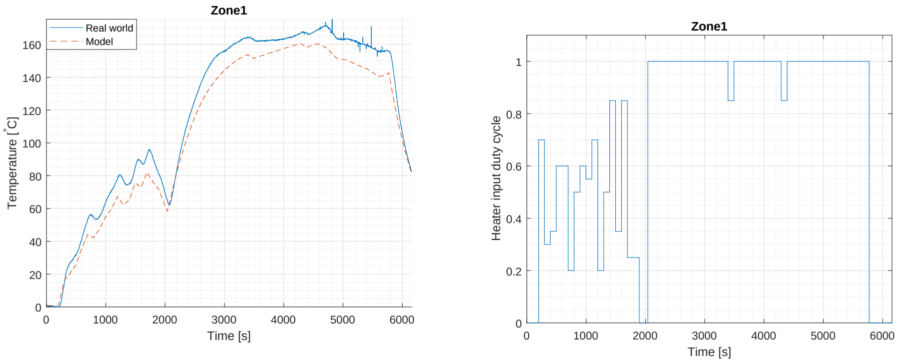
by the following matrices:

$$A = \begin{bmatrix} 0.992 & 0.0018 & 0 & 0 & 0 & 0 \\ 0.0023 & 0.9919 & 0.0043 & 0 & 0 & 0 \\ 0 & -0.0042 & 1.0009 & 0.0024 & 0 & 0 \\ 0 & 0 & 0.0013 & 0.9979 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.9972 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.9953 \end{bmatrix}$$

$$B = \begin{bmatrix} 1.0033 & & 0 & 0 & 0 & 00 & -0.2175 \\ 0 & 1.0460 & 0 & 0 & 0 & 0 & -0.0788 \\ 0 & 0 & 1.0326 & 0 & 0 & 0 & -0.0020 \\ 0 & 0 & 0 & 0.4798 & 0 & 0 & -0.0669 \\ 0 & 0 & 0 & 0 & 0.8882 & 0 & 0.1273 \\ 0 & 0 & 0 & 0 & 0 & 1.1699 & -0.1792 \end{bmatrix}$$

$$\begin{aligned} C &= I_6 \\ D &= 0 \end{aligned} \tag{3-19}$$

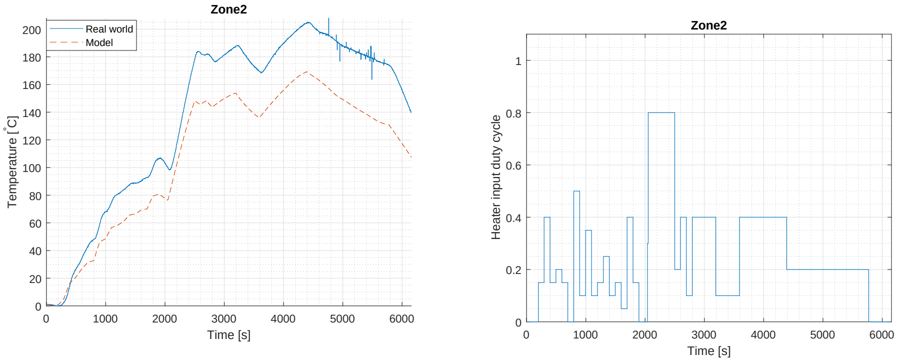
With I_6 representing the 6×6 identity matrix. The columns of B represent the effect of the heaters, with column one corresponding to zone one, column two to zone two, etc. The last column represents the effect of the motor input on the temperature of each zone. See figures 3-6, 3-7, 3-8, 3-9, 3-10, 3-11, and 3-12 for the simulated and measured temperature responses, and input signals.



(a) Zone 1 measured and modeled temperature response

(b) Zone 1 heater input

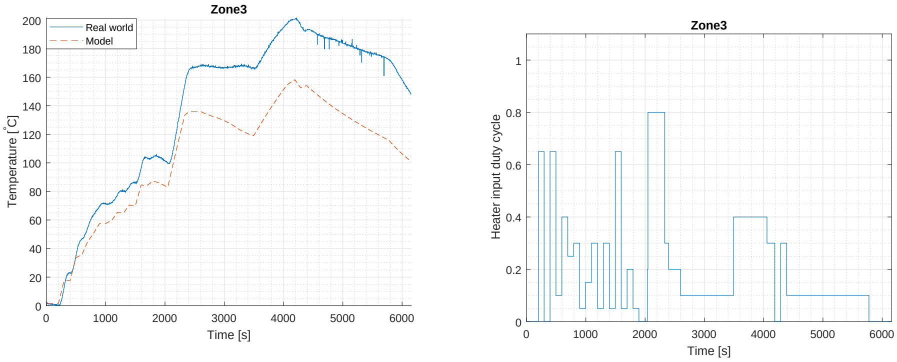
Figure 3-6: Zone 1 temperature response and heater signal



(a) Zone 2 measured and modeled temperature response

(b) Zone 2 heater input

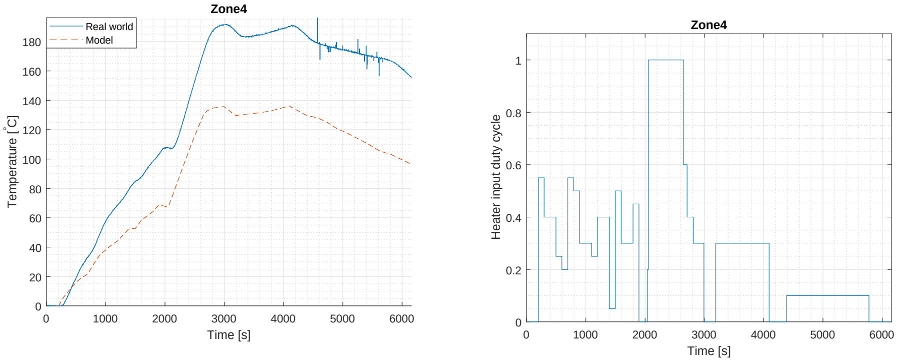
Figure 3-7: Zone 2 temperature response and heater signal



(a) Zone 3 measured and modeled temperature response

(b) Zone 3 heater input

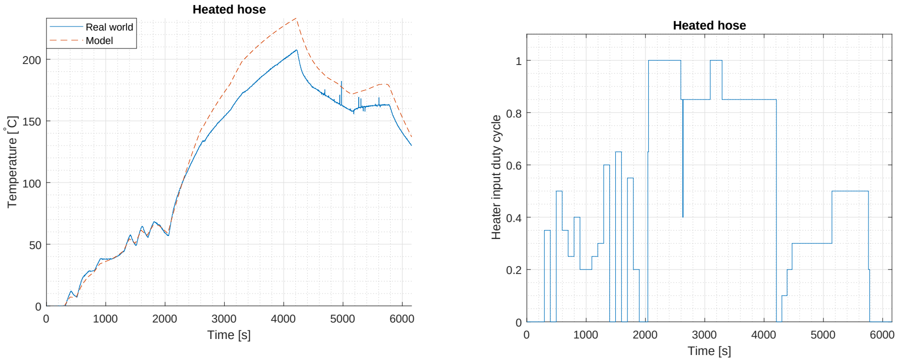
Figure 3-8: Zone 3 temperature response and heater signal



(a) Zone 4 measured and modeled temperature response

(b) Zone 4 heater input

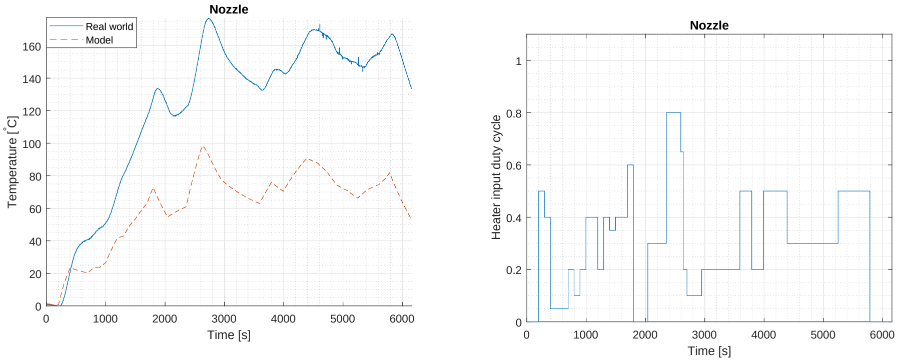
Figure 3-9: Zone 4 temperature response and heater signal



(a) Heated hose measured and modeled temperature response

(b) Heated hose heater input

Figure 3-10: Hose temperature response and heater signal



(a) Nozzle measured and modeled temperature response

(b) Nozzle heater input

Figure 3-11: Nozzle temperature response and heater signal

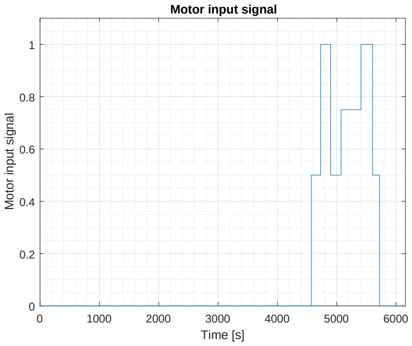


Figure 3-12: Motor input signal

Now that a MIMO model is identified the accuracy of the model needs to be assessed. From a visual inspection it can be seen that the system dynamics are captured in the model, but that as the simulation time increases the simulated temperature drifts away from the measured temperature. This indicates the possibility of unmodeled dynamics. A quantitative description of model accuracy is given, as indicated in section 3-1-1, by the variance-accounted-for (VAF) fit percent. The VAF for each zone is presented in table 3-1.

Table 3-1: Identification data model fit

Partition	VAF
Zone 1	99.3%
Zone 2	95.9%
Zone 3	93.9%
Zone 4	90.3%
Heated hose	98.6%
Nozzle	74.8%

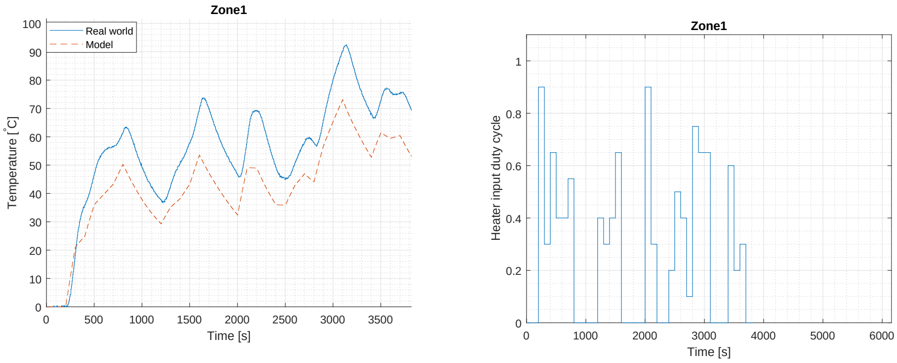
The VAF values are quite high for each zone, except for the nozzle. This might be caused by the fact that the nozzle has a very small thermal capacitance due to its small size, which makes it susceptible to disturbances such as drafts.

Before any final conclusions on the quality of the model can be drawn a model simulation with validation data for comparison is needed. Due to technical difficulties no validation dataset is available where the motor input is used to excite the system. Therefore, the used validation dataset can only validate the thermal response due to the heaters. See figures 3-13, 3-14, 3-15, 3-16, 3-17, and 3-18 for the validation data and model simulation. By visually inspecting the aforementioned figures it can be seen that the model captures the dynamics, but that the true temperature drifts away from the modeled response as the simulation time increases; which indicates, just as the identification data did previously, unmodeled dynamics. Nevertheless a quantitative description of the model accuracy is needed; see table 3-2 for the VAF fit percent of each partition.

From the VAF values it can be seen that the model fit of the nozzle is the worst; this could be due to the sensisitivity to disturbances, as mentioned previously. Since the rest of the model VAF's are approximately equal or greater than 80% the model is deemed good enough for control.

Table 3-2: Validation data model fit

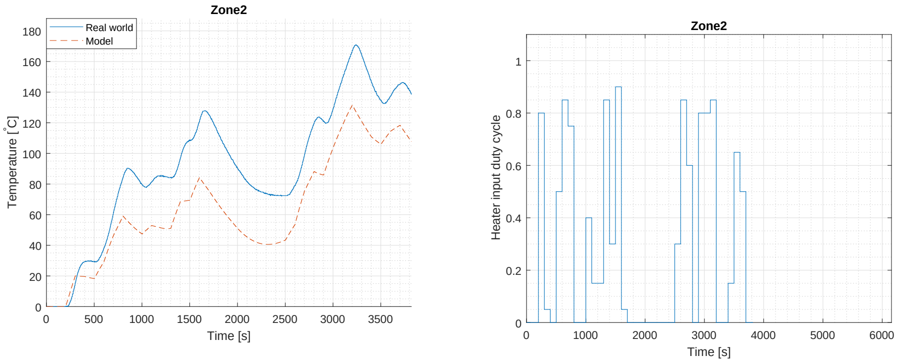
Partition	VAF
Zone 1	94.4%
Zone 2	90.8%
Zone 3	82.0%
Zone 4	79.6%
Heated hose	99.8%
Nozzle	68.5%



(a) Zone 1 measured and modeled temperature response of the validation data

(b) Zone 1 heater input validation data

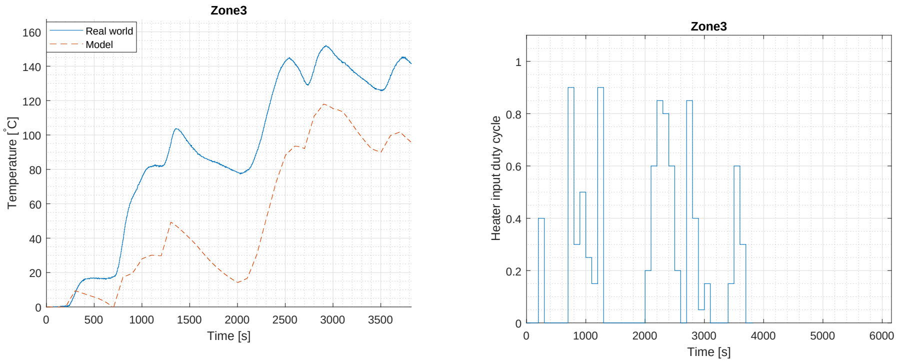
Figure 3-13: Zone 1 temperature response and heater signal of the validation dataset



(a) Zone 2 measured and modeled temperature response validation data

(b) Zone 2 heater input validation data

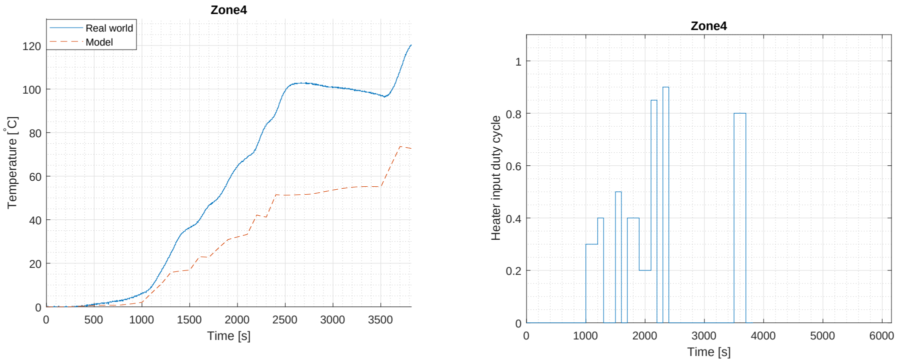
Figure 3-14: Zone 2 temperature response and heater signal validation dataset



(a) Zone 3 measured and modeled temperature response validation data

(b) Zone 3 heater input validation data

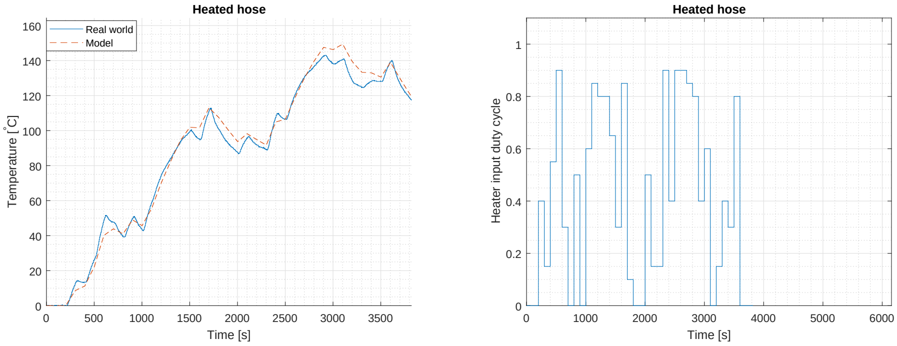
Figure 3-15: Zone 3 temperature response and heater signal validation dataset



(a) Zone 4 measured and modeled temperature response validation data

(b) Zone 4 heater input validation data

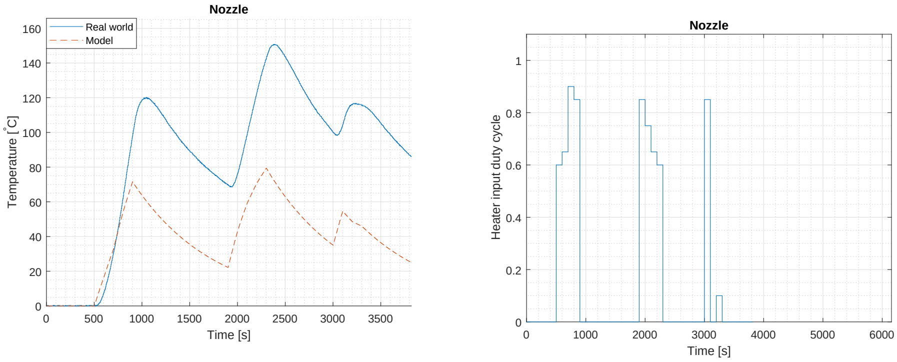
Figure 3-16: Zone 4 temperature response and heater signal validation dataset



(a) Heated hose measured and modeled temperature response validation data

(b) Heated hose heater input validation data

Figure 3-17: Hose temperature response and heater signal validation dataset



(a) Nozzle measured and modeled temperature response validation data

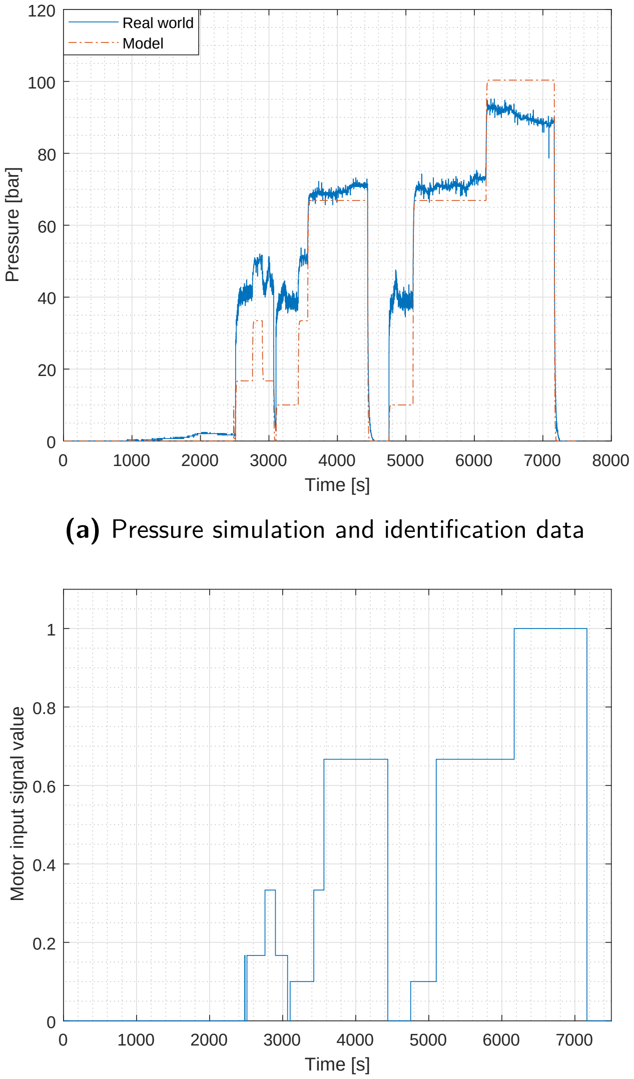
(b) Nozzle heater input validation data

Figure 3-18: Nozzle temperature response and heater signal validation dataset

3-3 Extrusion pressure model identification

Now that the temperature response is modeled it is time to model the pressure response to the motor input. Due to the robustness and stability issues of the data acquisition hardware and software system, the motor input does not consist of a pseudo-random-binary-signal. Instead, an operator chooses motor input values based on the observed pressure values.

The singular values of the motor-pressure identification data, see figure 3-21, suggest that a first order model can be identified to describe the system behaviour adequately. The identified first-order linear model is used to simulate the pressure response due to the motor input; see figure 3-19.



(a) Pressure simulation and identification data

(b) Motor input signal

Figure 3-19: Linear SISO pressure model simulation and identification data

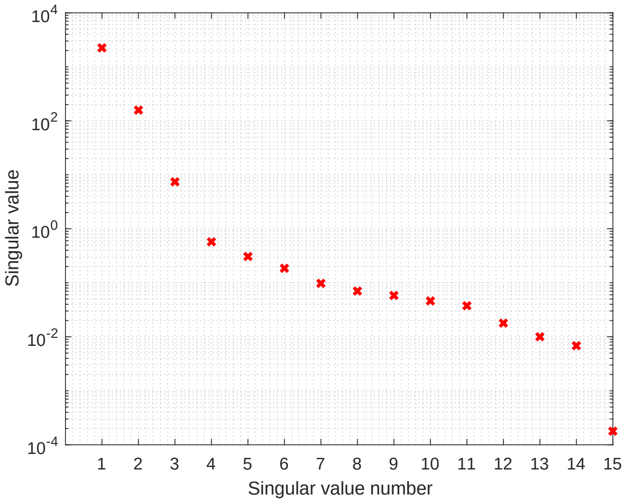


Figure 3-21: Singular values of the pressure identification data

The pressure response seems to behave globally in a nonlinear fashion, and locally as a first-order system. One way to describe the nonlinear dynamics is via a wiener model. A wiener model is a block-structured model consisting of a linear time-invariant (LTI) system block followed by a static nonlinear output block. See figure 3-20.

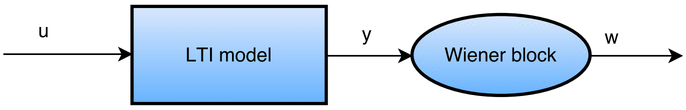


Figure 3-20: Wiener model structure

The static output block applies a nonlinear mapping from the LTI output to the total system output. These mappings can be represented by various functions such as piecewise-linear, saturation, dead-zone, or polynomial functions. The main advantage of wiener models is that, if the output static nonlinear blocks is invertible, a linear system can be obtained by post-multiplying the wiener model with the inverse function. Once the model is linear all linear analysis and controller design techniques such as controllability, observability, and stability analysis can be performed.

Using the identified linear model as initial estimate for the LTI system block, a wiener model is identified. As static nonlinearity block a piecewise linear function is chosen because it is easily inverted (the inverse equals the original piecewise linear function with x- and y-coordinate switched). Not every piecewise linear function can be inverted; the inversion can only be performed if the value of the piecewise linear function is strictly increasing or decreasing over its entire domain. The amount of breakpoints of the piecewise linear function is changed until a strictly increasing or decreasing function is identified, which still describes the system behaviour adequately. See figure 3-23 for the strictly increasing piecewise linear function.

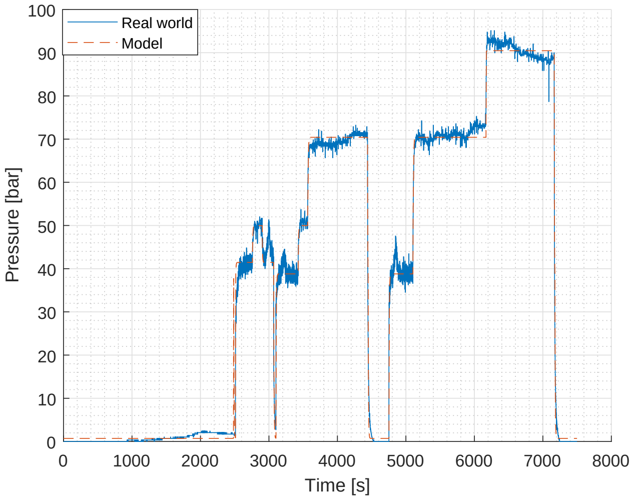


Figure 3-22: Identification data simulation of the pressure at zone four

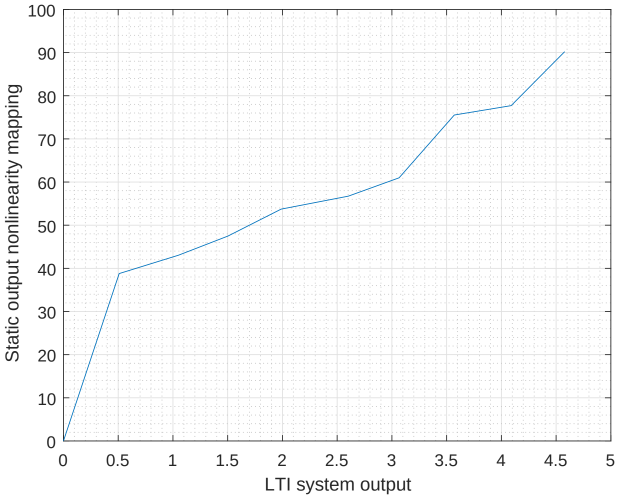
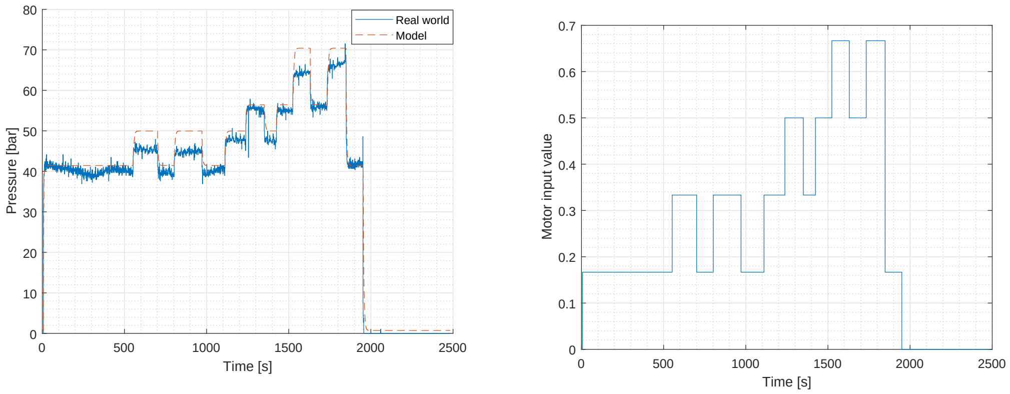


Figure 3-23: Strictly increasing piecewise linear function used as static output nonlinearity

The identified wiener model is used to simulate the response to a validation input signal. A comparison between the validation data and the simulated pressure response can be found in figure 3-24. Visual inspection indicates a good model fit; this is acknowledged by the VAF fit percent value of 99.5%. The identified nonlinear wiener model is good enough for control.



(a) Pressure response model simulation validation data

(b) Motor input validation data

Figure 3-24: Validation data simulation of the pressure at zone four

3-4 Summary

In this chapter two linear system identification methods were discussed. They were used to identify two models: one describing the temperature response of the large-scale 3D printer to heater and motor inputs, and one describing the pressure response at zone four subject to the motor input. The temperature model is a sixth order linear time-invariant state-space model, and the pressure model is a wiener model with, where the static output nonlinearity is represented with a piecewise linear function. The variance-accounted-for (VAF) of the temperature model is greater or equal than 80% for each zone, except for the nozzle where the VAF is 68.5%. The VAF of the pressure model is 99.5%. Both models are deemed good enough for control.

The only undescribed system dynamics involve the effect of the pressure at zone four on the volumetric output flow. Since the volumetric output flow is not measured, a computational fluid dynamics approach is chosen to model the dynamics. This will be the topic of the next chapter.

Chapter 4

Modeling the volumetric output flow rate

The aim of this chapter is to model the volumetric output flow rate response to pressure changes. In the previous chapter a temperature and pressure model of the extruder were obtained by applying system identification techniques to input-output data obtained from real-life experiments. Unfortunately, the flow rate is not measured and a flow rate measurement device could not be installed in a timely manner. The approach to overcome this problem is by performing an accurate simulation using a commercially availabe computational fluid dynamics (CFD) package; more specifically ANSYS Fluent for Academic Research, version 18.0. Since the goal is to use the flow rate model for real-time control, and CFD model simulations are computationally intensive, a wiener model will be fit to the CFD simulation data for simplification purposes.

First, an introduction to fluid dynamics is made by introducing the equations governing general fluid flow. These equations are then used to analytically solve a steady-state pipe flow problem of a newtonian fluid. An extension to non-newtonian steady-state pipe flow will be made which will be solved analytically as well. Then, the partial differential equations describing the unsteady non-newtonian laminar pipe flow problem are presented, for which no analytical solution is presently known. After that it is presented how the unsteady pipe-flow problem is modeled and solved in ANSYS Fluent. The obtained numerical solution is compared with the derived analytical solution for steady-state non-newtonian pipe flow to check if the CFD simulation is valid. After validating the steady-state simulation results an unsteady flow simulation is conducted. The obtained unsteady pressure and volumetric output flow data is subsequently used to identify a wiener model.

4-1 Fluid dynamics introduction

A fluid is distinctive from a solid in the way it reacts to a shearing force. Solids can resist shear stress by undergoing a static (elastic) deflection, and a fluid cannot. Fluids move when subjected to a shear stress. To quantify a fluid's resistance to flow the term viscosity is used, which describes a relation between applied shear stress and strain rate, see figure 4-1. If a fluid shows a linear relation between applied shear stress and resuling strain it is called a newtonian fluid (4-1). For clarification on the meaning of symbols used in this chapter, see table 4-1.

Another distinction between fluids and solids is that fluid flows that are bounded by a solid surface seek momentum and energy equilibrium with that surface. All fluids at a point of contact take on the velocity of that surface. This is called the no-slip condition, and provides a set of boundary values for the (partial) differential equations to be solved later in the chapter.

$$\tau = \mu \left(\frac{du}{dy} \right) \tag{4-1}$$

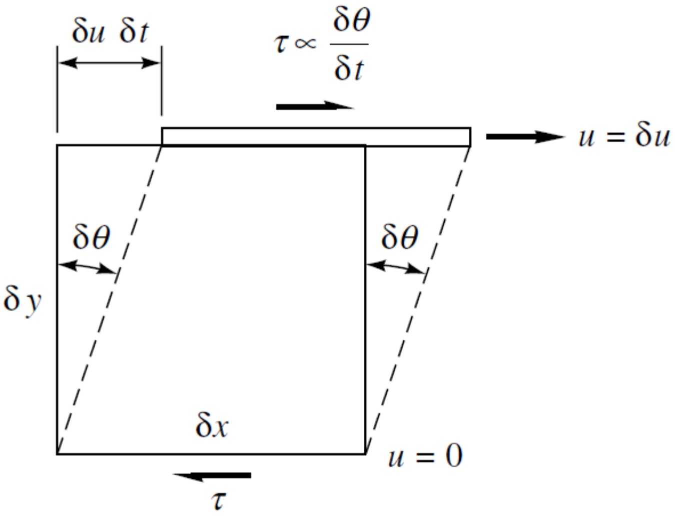


Figure 4-1: Fluid element free body diagram, adapted from [2]

Now that the distinction between fluids and solids is made clear, it is time to analyse the flow of fluids by applying some of the fundamental laws of mechanics to an infinitesimal control volume element. Applying the principle of conservation of mass to an infinitesimal control volume element results in the so-called continuity equation (4-2).

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho u) + \frac{\partial}{\partial y}(\rho v) + \frac{\partial}{\partial z}(\rho w) = 0 \tag{4-2}$$

And applying the law of conservation of momentum to an infinitesimal control volume element results in the linear momentum equations (4-3) (written in the cartesian coordinate system).

$$\begin{aligned}
\rho g_x - \frac{\partial p}{\partial x} + \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} &= \rho \left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z} \right) \\
\rho g_y - \frac{\partial p}{\partial y} + \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \tau_{yy}}{\partial y} + \frac{\partial \tau_{zy}}{\partial z} &= \rho \left(\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z} \right) \\
\rho g_z - \frac{\partial p}{\partial z} + \frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial \tau_{zz}}{\partial z} &= \rho \left(\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z} \right)
\end{aligned} \tag{4-3}$$

These equations are valid for any fluid under any general motion. By making some assumptions on the fluid flow under consideration, and combining that information with equations 4-2 and 4-3, a variety of flow problems can be solved. The next section will show how to analytically solve the laminar steady-state pipe flow of a newtonian fluid.

Table 4-1: List of symbols used in this section

Symbol	Meaning
ρ	Density [kg/m^3]
u	Flow velocity in x-direction [m/s]
v	Flow velocity in y-direction [m/s]
w	Flow velocity in z-direction [m/s]
g	Gravitational acceleration [m/s^2]
τ	Shear stress [N/m^2]
p	Pressure [Pa]
μ	Viscosity [kg/ms]
R	Hose radius [m]
x	Axial direction
r	Radial direction
θ	Strain
L	Pipe length [m]
a	Speed of sound [m/s]

4-1-1 Laminar steady-state pipe flow of a newtonian fluid

Consider a circular pipe where a fluid driven by a pressure gradient flows through, see figure 4-2. Further assumptions are that $v = w = 0$[m/s] and that $dp/dx = constant < 0$. Thus fluid only flows in the axial direction of decreasing pressure. Combining the x-momentum equation with the newtonian fluid viscosity equation (4-1) and the above assumptions results in the following momentum equation:

$$\rho \frac{\partial u}{\partial t} = -\frac{\partial p}{\partial x} + \mu \frac{\partial^2 u}{\partial y^2} \tag{4-4}$$

It reduces to an ordinary differential equation if it is solved for the steady-state condition by setting $\frac{\partial u}{\partial t} = 0$. Solving it by double integration and applying the no-slip boundary conditions $u(h) = u(-h) = 0$ gives the flow velocity profile, see equation (4-5). In conclusion, fully developed newtonian pipe flow results in a quadratic flow velocity profile. The next section extends the obtained steady-state solution to non-newtonian fluids.

$$u(y) = -\frac{dp}{dx}\frac{h^2}{2\mu}\left(1 - \frac{y^2}{h^2}\right) \tag{4-5}$$

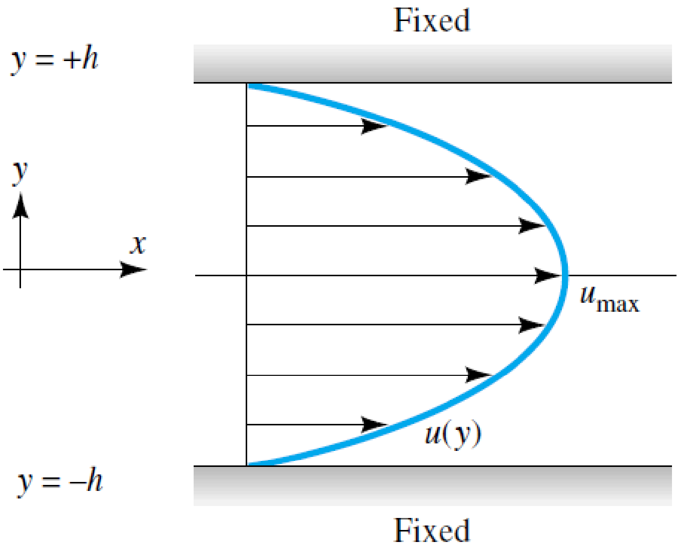


Figure 4-2: Fully-developed laminar pipe flow for a newtonian fluid, adapted from [2]

4-1-2 Laminar steady-state pipe flow of a non-newtonian fluid

The viscosity of a non-newtonian fluid viscosity can be described via a power-law (4-6). The term c_m is called the consistency index, and n is the power-law coefficient. For $0 < n < 1$ the fluid is called shear-thinning, for which a typical example is quicksand or ketchup; it gets runnier the more it is strained. If $n > 1$ the fluid is dilatant which means that the fluid behaves more like a solid the faster it is strained; as an example consider oobleck (a suspension of cornstarch in water).

$$\tau = c_m \left(\frac{\partial u}{\partial y}\right)^n \tag{4-6}$$

Taking advantage of the symmetry of the hose the momentum equations can be written in cylindrical coordinates. The fluid problem assumptions are that flow in the radial and azimuth direction are equal to zero, and that the pressure only varies in the axial direction ($\partial P/\partial r = 0$). Therefore only the axial momentum equation (4-7) needs to be considered.

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = -\frac{1}{\rho}\frac{\partial P}{\partial x} + \frac{1}{\rho r}\frac{\partial(r\tau_{rx})}{\partial r} \tag{4-7}$$

The continuity equation can now be written as:

$$\frac{\partial P}{\partial t} + u\frac{\partial P}{\partial x} + \rho a^2\frac{\partial u}{\partial x} = 0 \tag{4-8}$$

For the steady-state solution it must hold that $\partial u/\partial t = 0$. Solving the axial momentum equation by applying the boundary conditions results in the flow velocity profile, as written in equation 4-9 [15] [2].

$$u(r) = \left(-\frac{1}{2c_m}\frac{dP}{dx}\right)^{1/n} \cdot \left(\frac{n}{n+1}\right) \cdot \left(R^{\frac{n+1}{n}} - r^{\frac{n+1}{n}}\right) \tag{4-9}$$

See figure 4-3 for a visual representation of the simulated steady-state flow velocity profile of the hose output for the selected material of PPG ($n = 0.38$, $c_m = 7500$, $\frac{dP}{dx} = 7.8$ [bar/m]).

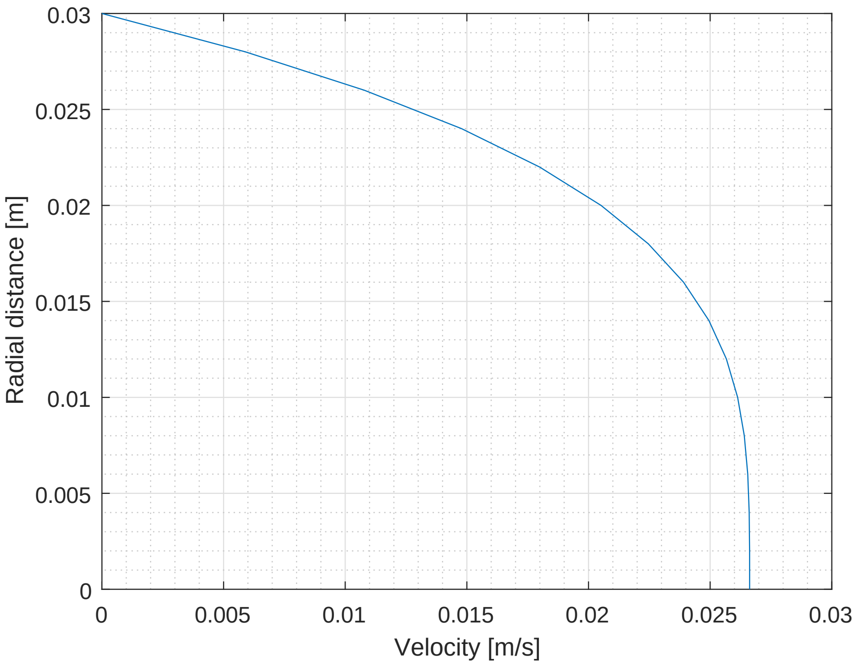


Figure 4-3: Non-newtonian fluid laminar pipe flow velocity profile

Solving the partial differential equations (4-2) (4-3) for the unsteady pipe flow case is more challenging and usually resolved using techniques from computational fluid dynamics [2]. This will be discussed in the next section.

Parameter	**Value**
n	0.38
c_m	7500
R	0.03
dP/dx	7.8

4-2 Computational fluid dynamics

The philosophy of CFD is to quantatively describe the solution to a flow problem. Overly simplified, this is done by setting up the governing equations, discretizing them, applying boundary conditions, and solving the resulting system of equations simultaneously. The continuity (4-8) and momentum (4-7) equations presented in the previous section combined with the non-newtonian viscosity equation (4-6) completely specify the flow behaviour of a non-newtonian fluid. To discretize the partial differential equations a finite difference approximation is commonly used. For example define $u_{i,j}$ as the x-component of the velocity at a point (i, j), then the velocity $u_{i+1,j}$ at point $(i+1, j)$ can be denoted in terms of a Taylor's series around the point (i, j):

$$u_{i+1,j} = u_{i,j} + \left(\frac{\partial u}{\partial x}\right)_{i,j} \Delta x + \left(\frac{\partial^2 u}{\partial x^2}\right)_{i,j} \frac{(\Delta x)^2}{2} + \left(\frac{\partial^3 u}{\partial x^3}\right)_{i,j} \frac{(\Delta x)^3}{6} + \cdots \tag{4-10}$$

This infinite sum is truncated since it is impossible to carry an infinite number of terms. If Δx is chosen appropriately (very small), then $(\Delta x)^2$ is negligble compared to it. The approximate form can then be written as:

$$u_{i+1,j} \approx u_{i,j} + \left(\frac{\partial u}{\partial x}\right)_{i,j} \Delta x \tag{4-11}$$

Or more clearly as:

$$\left(\frac{\partial u}{\partial x}\right)_{i,j} = \frac{u_{i+1,j} - u_{i,j}}{\Delta x} \tag{4-12}$$

This equation is called the first-order forward difference. The approximation error is:

$$\mathcal{O}(\Delta x) = \left(\frac{\partial^2 u}{\partial x^2}\right)_{i,j} \frac{(\Delta x)}{2} + \left(\frac{\partial^3 u}{\partial x^2}\right)_{i,j} \frac{(\Delta x)^2}{6} + \cdots \tag{4-13}$$

In a similar fashion the backward difference (4-14) and central difference derivative approximations can be derived.

$$\left(\frac{\partial u}{\partial x}\right)_{i,j} = \frac{u_{i,j} - u_{i-1,j}}{\Delta x} + \mathcal{O}(\Delta x) \tag{4-14}$$

$$\left(\frac{\partial u}{\partial x}\right)_{i,j} = \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} + \mathcal{O}(\Delta x)^2 \tag{4-15}$$

The advantage of using the central difference approximation is that the error term is smaller, as denoted by $\mathcal{O}(\Delta x)^2$. By applying the finite difference approximation to the whole grid a (large) system of equations is formed which are then solved simultaneously; this is called the implicit method. A multitude of practical problems arises, however delving further into it is beyond the scope of this thesis. It suffices to say that topics such as numerical stability, finite difference approximation error, and computation time are heavily interconnected and actively being researched in the field of computational fluid dynamics [20]. The next section discusses how the unsteady non-newtonian pipe flow is modeled in ANSYS Fluent, and how the obtained results are validated.

4-2-1 ANSYS Fluent Academic Research version 18.0 simulation

The heated hose has a radius of 0.03 meter and a length of 5 meter. It is assumed to be in a horizontal position, while in reality the hose is curved around the robot arm. This will make almost no difference to the obtained solution; if the hose is pointed entirely vertical the pressure difference would be $900 \cdot 9.81 \cdot 5 \approx 0.44$ bar. Typical extrusion pressures are in the range of 40 to 100 bar (see chapter 3), which means that the pressure erorr is approximately 1%. This is deemed acceptable.

Since no complicated flows are expected the heated hose is meshed using basic rectangular elements; 10 in the vertical direction spanning the radius of 0.03 [m] and 100 in the horizontal direction spanning the hose length of 5 [m]. See figure 4-4 for the discretized heated hose geometry.

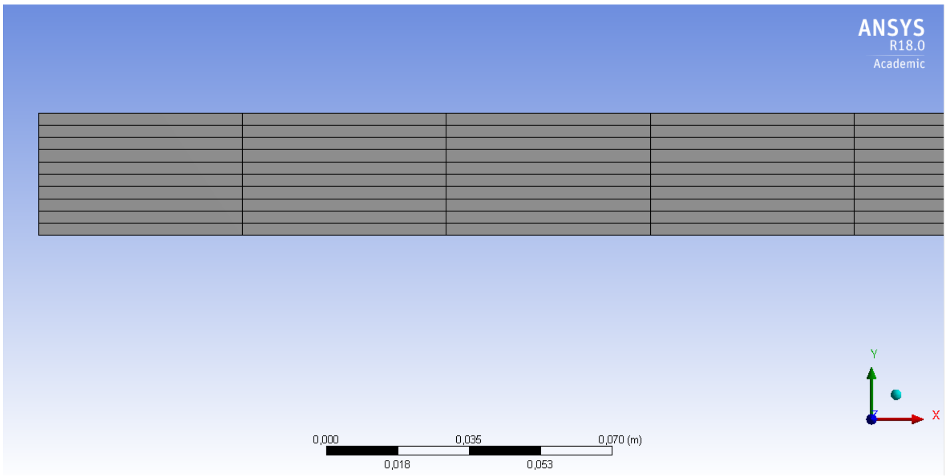


Figure 4-4: Partial heated hose geometry with mesh

Fortunately the CFD software provides checks on the convergence of the solution; however the validity should be checked by the user. The validity of the obtained solution is checked by comparing the steady-state simulated flow velocity profile with the analytically computed flow velocity profile. See figure 4-5 for a visual comparison of the velocity profiles and table 4-2 for a comparison of the volumetric flows.

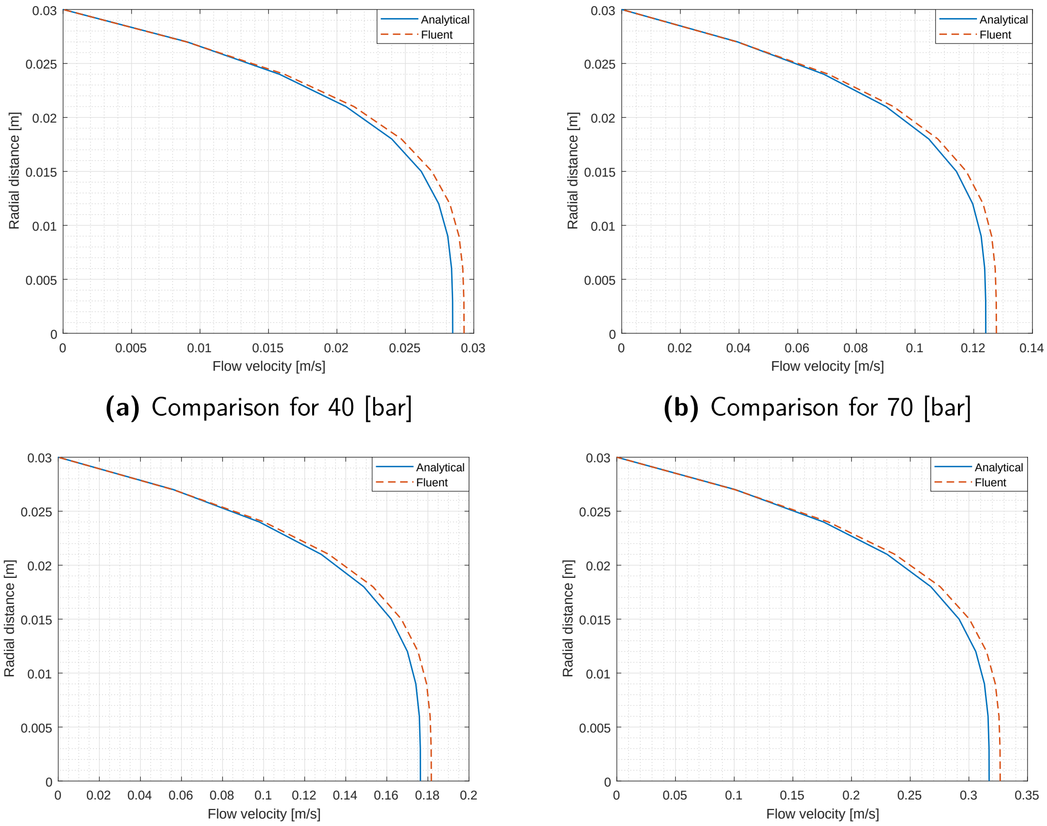
Table 4-2: Table showing the error between the analytical solution and CFD solution to the steady-state non-newtonian pipe flow problem for various pressures

Pressure [bar]	**Analytical flow** ($\times 10^{-5}[m^3/s]$)	**CFD** ($\times 10^{-5}[m^3/s]$)	**error**[%]
40	5.17	5.30	2.5
70	22.53	23.10	2.5
80	32.02	32.85	2.5
100	57.60	59.08	2.5

The steady-state volumetric output flow errors of 2.5% are deemed low enough for the CFD model to be valid. The obtained input-output data can be used for system identification, as will be done in the next section.

4-2-2 Flow rate modeling using system identification

First, some input-output data must be gathered before the system identification procedure can start. The unsteady input pressure can be seen in figure 4-6a. The corresponding volu-



(a) Comparison for 40 [bar]

(b) Comparison for 70 [bar]

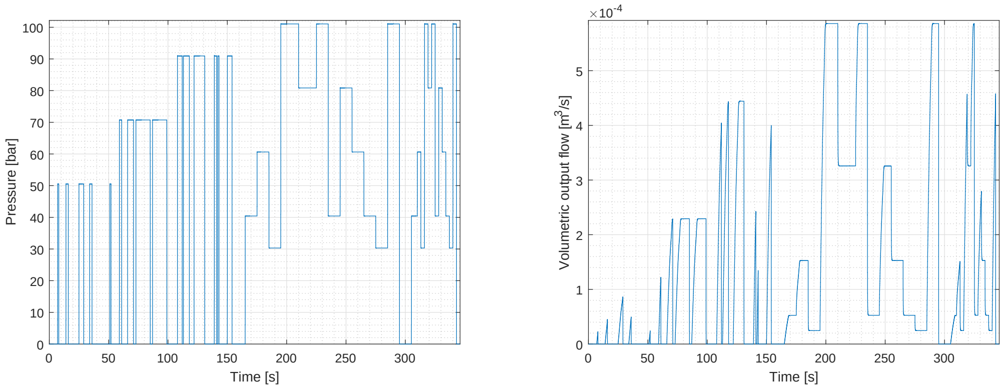
(c) Comparison for 80 [bar]

(d) Comparison for 100 [bar]

Figure 4-5: Flow velocity profile comparison of the analytical and CFD method for steady-state input pressures

metric output flow rate, see figure 4-6b, is obtained by numerically integrating the velocity profiles obtained using the trapezoidal rule.

As a first attempt at modeling the system, a linear model is identified. The accompanying singular values, see figure 4-7, suggest that a first-order model should fit the data. By visually inspecting the linear model simulation, see figure 4-8, one can see that the model fits the data quite poorly. The next step is to try and identify a wiener model where the output nonlinearity is a strictly increasing piecewise linear function. The model fit in terms of variance-accounted-for (VAF) can be found in table 4-3. The VAF fit percentages are 81.5% and 97.8% for the linear and wiener model respectively, indicating a substantial improvement. See figure 4-9 for a visual comparison between identification data and wiener model simulation. Since the model is identified by using data from a CFD simulation no validation data is needed; the CFD simulation will produce the same results upon repeated simulation.



(a) Simulated pressure input signal **(b)** Simulated olumetric output flow rate signal

Figure 4-6: Unsteady flow rate simulation

Table 4-3: VAF fit percentages for the linear model and wiener model

Model	**VAF**
Linear	81.5%
Wiener	97.8%

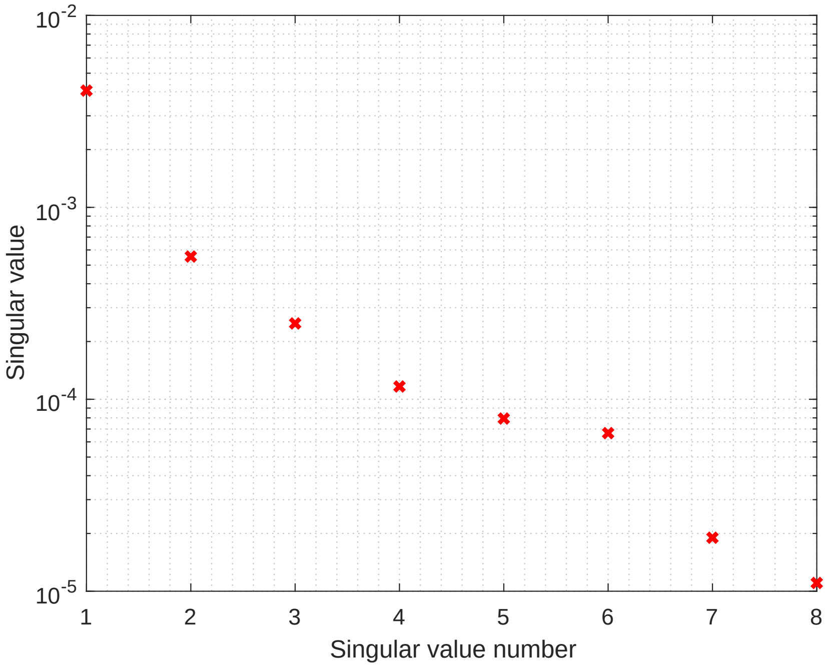


Figure 4-7: Singular values of the pressure-flow-data indicate a first-order model

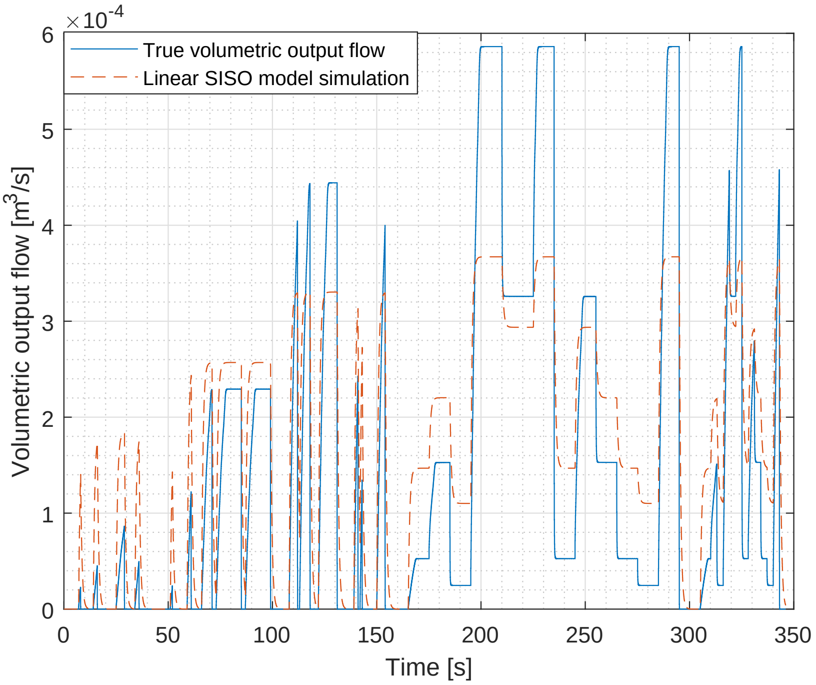


Figure 4-8: Comparison of true volumetric output flow with linear SISO model simulation

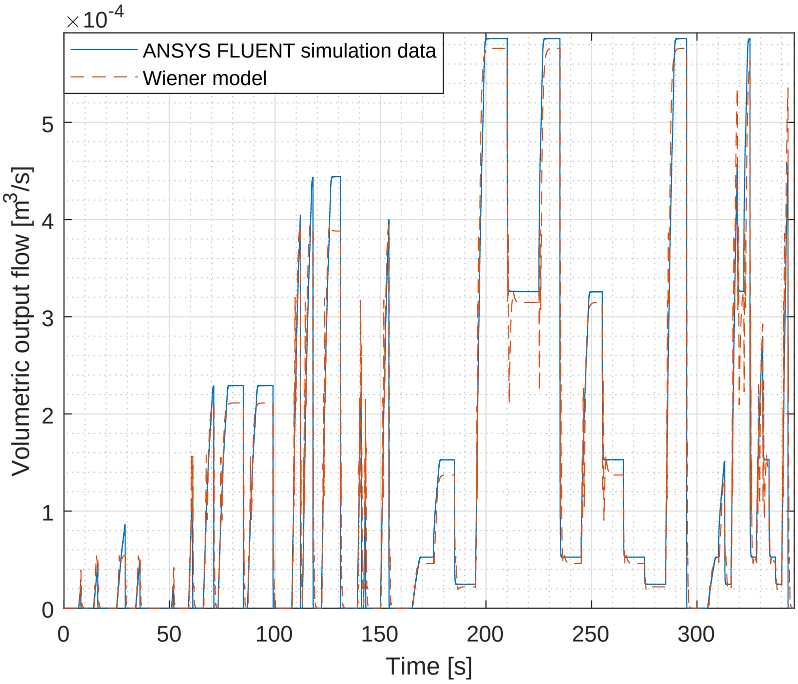


Figure 4-9: Comparison of true volumetric output flow with the wiener model simulation

4-3 Summary

In this chapter the basics of fluid mechanics was presented and used to find solutions to the steady-state pipe flow problem for newtonian and non-newtonian fluids. To solve the unsteady case a commercially available computational fluid dynamics (CFD) software package, ANSYS Fluent for Academic Research version 18.0, was used. A brief description of the general idea behind CFD is then presented, after which the flow problem is set up. Next, the validity of the created CFD model was confirmed by comparing the CFD solution for a steady-state non-newtonian pipe flow problem to the analytical solution and noting that the predicted volumetric flow rates differed by no more than 2.5%.

The CFD model was then used to generate unsteady-state volumetric flow data by exciting the model with a varying inlet pressure. This data was then used to identify a wiener model, which had a variance-accounted-for (VAF) fit percent of 97.8%. In conclusion, the wiener model fits the obtained pressure-flow data very well and will be used for the simulation study in chapter 6.

Now that all relevant models are identified, it is time to discuss the model predictive control strategy. This will be the topic of the next chapter.

Chapter 5

Model predictive control

The advantages of MPC is that it can deal with constraints on input and output signals in a straightforward way, it can work with both linear and non-linear models (especially if the non-linear model is a hammerstein-wiener model, see [21]), and has been succesfully applied in various processes such as chemical reactors, a combustion engine and a plastic injection moulding process [22], [23]. In [24] a MPC scheme has been succesfully implemented for extrusion melt pressure and barrel temperature control. Two seperate control loops were used, one SISO to control the melt pressure with screw speed as input, and a MIMO loop for controlling the barrel temperatures. Since MPC can deal with constraints, is well-suited for processes with slow dynamics, and has succesfully outperformed traditional (PID) control methods [24] it is chosen as the preferred control strategy. The MPC methodology needs the following items:

- Model of the system
- Performance index (cost function)
- Constraints
- Optimization algorithm
- Receding horizon principle

MPC determines the optimal control input sequence by minimizing a performance index subject to constraints. Only the first control input sample will be implemented, after which the optimization step is again performed. This is called the receding horizon principle, see figure 5-1.

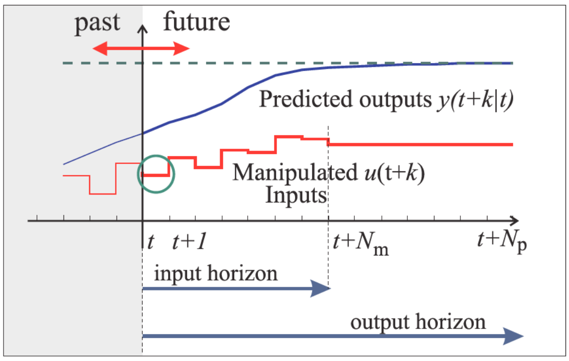


Figure 5-1: Receding horizon principle, adapted from [3]

5-1 Performance indices

The performance index, or cost function, is a major influence on what the input sequence will be. Each performance index containes a (weighted) sum of squared characteristic signals of the system. These characteristic signals always include the control signal and are complemented with the state signal, output signal, or some other signal such as a measure of how close the system output is to a specified boundary value. Most commonly used are:

- Generalized Predictive Control (GPC) performance index
- Linear Quadratic Predictive Control (LQPC) performance index
- Zone performance index

Generalized Predictive Control performance index

The GPC performance index is based on control and output signals and is given by:

$$J(k) = \sum_{j=N_m}^{N} \left(\hat{y}_p(k+j|k) - r(k+j)\right)^T \left(\hat{y}_p(k+j|k) - r(k+j)\right) + \lambda^2 \sum_{j=1}^{N} \Delta u^T(k+j-1|k)\Delta u(k+j-1|k) \tag{5-1}$$

$\hat{y}_p(k)$ $= P(q)\hat{y}(k)$ and is the weighted process output signal
$r(k)$ is the reference signal
$\hat{y}(k)$ is the process output signal
$\Delta u(k)$ is the control increment signal
N_m is the minimum cost-horizon
N is the prediction horizon
N_c is the control horizon
λ is the control signal weight
$P(q^{-1})$ $= 1 + p_1 q^{-1} + \cdots + p_n q^{-n}$ is a polynomial with desired closed-loop poles
q^{-1} is the inverse-shift operator

The polynomial $P(q^{-1})$ can be created by a designer to penalize overshoot and can be seen (in the noise-free case and when $\lambda = 0$) to place the n closed-loop poles at the roots of $P^{-1}(q^{-1})$.

Linear Quadratic Predictive Control performance index

The LQPC performance index penalizes the state and control signals, and is written as:

$$J(k) = \sum_{j=N_m}^{N} \hat{x}^T(k+j|k)Q\hat{x}(k+j|k) + \sum_{j=1}^{N} u^T(k+j-1|k)Ru(k+j-1|k) \tag{5-2}$$

$\hat{x}$	is the predicted state vector
u	is the control signal
Q	is the state weighing matrix
R	is the control signal weighing matrix

The weighing matrix Q largely determines the response by assigning weights to the different state variables. If the state-space model is obtained through system identification, and therefore the states have no known real-world physical meaning, tuning Q to obtain a desired response can be challenging.

Zone performance index

The zone performance index penalizes the output tracking error and the control increment signal. The output tracking error is only penalizes when the absolute value exceeds a certain threshold $\delta_{max,i}$. The performance index is given by:

$$J(k) = \sum_{j=N_m}^{N} \hat{\epsilon}^T(k+j|k)\hat{\epsilon}(k+j|k) + \lambda^2 \sum_{j=1}^{N} \Delta u^T(k+j-1|k)\Delta u(k+j-1|k)$$

Where:

$$\epsilon_i = \begin{cases} 0, & \text{for } |y_i(k) - r_i(k)| \leq \delta_{max,i} \\ y_i(k) - r_i(k) - \delta_{max,i}, & \text{for } y_i(k) - r_i(k) \geq \delta_{max,i} \\ y_i(k) - r_i(k) + \delta_{max,i}, & \text{for } y_i(k) - r_i(k) \leq -\delta_{max,i} \end{cases}$$

From here, a wide range of MPC-related topics open up, such as how to deal with non-linear models and/or non-linear constraints [21] [22] [23] [25] [26]. These topics are outside the scope of this thesis and will not be further discussed.

5-2 Constraints

Every system is in practice subject to physical constraints. Examples are actuator limitations, power limitations, robotic arms which are only allowed to reach a certain part of the workspace, environmental regulations, safety, maximum temperature allowed, maximum pressure allowed, etcetera. In most cases these constraints can be transformed into bounds on the control, output, or state signals. These bounds are of the inequality constraint form, using the input signal as an example:

$$u_{min} \leq u(k) \leq u_{max} \quad \forall k \in \mathbb{Z} \tag{5-3}$$

Some constraints arise in increasing the robustness of the algorithm used to solve the problem; for example the prediction horizon N_p and the control horizon N_c. After the prediction time-step exceeds the control horizon the control action should remain constant:

$$\Delta u(k+j|k) = 0 \quad \text{for } j \leq N_c \tag{5-4}$$

Another constraint typically used to force stability is the so-called state end-point constraint:

$$\hat{x}(k+N|k) = x_{ss} \tag{5-5}$$

Where x_{ss} is the steady-state value of the state-signal which forces the state of the system at the end of the prediction horizon to x_{ss}. The same approach can force the output of the system to a steady-state value. However, if it is not possible to reach the desired state or output in the prediction horizon interval it can lead to infeasibility [27].

5-3 Optimization

The performance indices presented in the previous section belong to a class called quadratic programming problems. Generally they can be written as:

$$\begin{aligned} &minimize \quad J(x) = \frac{1}{2}x^T Q x + c^T x \\ &subject \quad to \quad Ax \leq b \end{aligned} \tag{5-6}$$

The matrix Q is assumed positive-semidefinite and symmetric. This means that for any vector x the quantity $x^T Q x \geq 0$. If the original matrix Q is not symmetric the following trick can be used to create a symmetric matrix whilst keeping the same objective function:

$$Q_{new} = \frac{1}{2}\left(Q + Q^T\right)$$

For the unconstrained case the objective function has an extremum when the gradient is equal to zero:

$$\nabla J(x) = Qx + c = 0 \tag{5-7}$$

If Q is invertible the solution vector $\hat{x} = -Q^{-1}c$ attains the extremum. Because Q is positive-semidefinite the extremum will be a minimum.
Note that equality or inequality constraints might make the solution vector $\hat{x}$ infeasible. Some algorithms used to solve the quadratic programming problem with constraints are the:

- Modified simplex algorithm
- Interior point method

For optimization problems with a large amount of variables or constraints (> 1000) the interior point method is more efficient than the simplex method [19].

5-4 Tuning a model predictive controller

The purpose of tuning is to provide good tracking, disturbance rejection, and robustness against model mismatch. The tuning parameters for the GPC performance index are:

N_m	the minimum cost-horizon
N	the prediction horizon
N_c	the control horizon
λ	the control signal weighing factor
$P(q^{-1})$	$= 1 + p_1 q^{-1} + \cdots + p_n q^{-n}$ is a polynomial with desired closed-loop poles

The tuning process is iterative and usually starts by using rules of thumb to determine the initial settings. As a rule of thumb $N_m = 1$ is generally the most appropriate, unless there is dead-time or an inverse response in which case N_m should be increased. The prediction horizon, N, should be greater than the settling time of the output of the process to $\Delta u(k)$. This way all crucial dynamics are taken into account during the optimization step. The control horizon, N_c, determines after how many sampling instants the control signal becomes constant. A small control horizon ($N_c \ll N$) smoothes the control signal and forces its value to its steady-state value. Another effect of decrasing N_c is that it reduces the computational effort of the optimization step. As a rule of thumb $n \leq N_c \leq N/2$, where n is the model order. The value of λ should be chosen as small as possible, possible even zero. The optimization algorithm then only penalizes the output error while keeping the control input in the bounds of the predefined inequality constraint. Finally, the polynomial $P(q^{-1})$ should be chosen such that the roots of the polynomial are the desired poles of the closed loop [27].

The tuning of a model predictive controller can be formulated as an optimization problem. Discussing this is outside the scope of this thesis.

The model predictive controller updates its estimation each sampling instant by measuring the output. This measured output is in real systems always corrupted with noise, leading to incorrect output values. The reconstruction of the optimal 'true' output from measurements is done by implementing a kalman filter. This will be the topic of the next section.

5-5 Kalman filtering

Feedback is always applied based on measurements of the output(s). These measured outputs are often corrupted with noise. If the system is LTI then the output is a linear combination of the states of the system. Furthermore, an LQ feedback controller needs full information on the state of the system, even if not all the states are measured. This motivates the need for state reconstruction as is done by an observer. An observer tries to estimate the state (denoted by $\hat{x}$) by combining measurements of the system output, and system inputs, see equation 5-8

$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + L(y - \hat{y}) \tag{5-8}$$

See equation 5-9 to see that the estimated state converges to the true state if $A - LC$ is hurwitz.

$$\begin{aligned} x_e(k+1) &= \hat{x}(k+1) - x(k+1) \\ &= A\hat{x}(k) + Bu(k) - Ax(k) - Bu(k) + L(y(k) - \hat{y}(k)) \\ &= A\left(\hat{x}(k) - x(k)\right) - LC(\hat{x}(k) - x(k)) \\ &= (A - LC)\, x_e(k) \end{aligned} \tag{5-9}$$

This type of observer is called a Luenberger observer. If the output measurements are corrupted with noise a kalman filter needs to be used. The kalman filter presents an optimal way of retrieving the 'true' output if the characteristic of the noise (the covariance matrix) is known. The LTI system description corrupted with noise is given by equation 5-10

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + w(k) \\ y(k) &= Cx(k) + v(k) \end{aligned} \tag{5-10}$$

The covariance matrix of the noise is given by

$$Cov(v,w) = E\left[\begin{bmatrix} v(k) \\ w(k) \end{bmatrix} \begin{bmatrix} v(j)^T w(j)^T \end{bmatrix}\right] = \begin{bmatrix} R(k) & S(k)^T \\ S(k) & Q(k) \end{bmatrix} \tag{5-11}$$

The kalman filter tackles the problem of finding the optimal value of the feedback gain K by forcing the estimated state to be unbiased and minimum variance. The kalman gain is calculated as:

$$L(k) = \left(S(k) + AP(k)C^T\right)(R(k) + CP(k)C^T)^{-1} \tag{5-12}$$

Where $P(k)$ is found by solving the riccati equation

$$P(k+1) = AP(k)A^T + Q - \left(S + AP(k)C^T\right)\left(R + CP(k)C^T\right)^{-1}\left(S + AP(k)C^T)^T\right) \tag{5-13}$$

[18].

Chapter 6

Simulation study

The aim of this chapter is to present a simulation study where the models obtained in previous chapters are used to control the volumetric flow rate. As stated previously, the goal is to minimize the dimensional errors that occur during the printing process. There are two error sources which will be investigated. The first error source occurs due to the fact that the robot arm does not move at a constant speed. When the robot arm is about to turn a corner, or start a new layer, its speed decreases. This is followed by a full stop, and then by the robot arm accelerating in a new direction. Not changing the volumetric output flow rate during this speed transition will cause an excess of material deposition. If the robot arm speed trajectory is known a priori it is possible to use a model predictive controller to change the volumetric flow rate before errors occur. The second error source occurs because the motor input has an undesirable effect on the temperature of the machine. To keep the steady-state flow rate constant, the temperature must be constant since changes in temperature can cause undesirable changes in viscosity, as was discussed in chapters 1 and 2.

In the first section the flow tracking problem is reformulated as a pressure tracking problem because the flow rate is not measured. In order to use a linear model predctive control aglorithm a linear model is needed. This motivates the need of transforming the pressure reference tracking as a linear pressure reference tracking problem. Next, three plausible flow reference trajectories are presented because the true desired flow trajectory is unknown. Then, some simulations are performed to see whether the proposed model predictive controller is able to track the reference linear pressure trajectories. The simulated linear pressure responses are then to the corresponding resulting flow rate trajectories, which will be compared to the original desired flow rate trajectories. Finally, the effect of the motor input on the temperature response will be simulated and discussed. The chapter ends with a discussion of the results.

6-1 The flow rate reference tracking problem

As discussed before, the goal is to make the flow rate track a given reference trajectory. For dimensionally accurate 3D printing it is imperative that the layer thickness is constant. This is achieved by keeping the quotient of the robot arm speed and volumetric flow rate constant. When the robot arm approaches a corner or is about to start a new layer, it decelerates, comes to a full stop, and then accelerates in a new direction. This means that the flow rate should be decreased and increased at the same rate as the robot arm speed to print layers with constant thickness. The true shape of the robot arm speed trajectory is unknown, therefore a few basic assumptions regarding the shape are made. It is assumed that the acceleration and deceleration times are constant and have a duration of 5 [s]. Furthermore, the robot arm, and therefore the flow rate, is assumed to be at rest at $t = 0$ seconds. At $t = 10$ seconds the robot arm accelerates and reaches a steady-state at $t = 15$ seconds. After a while a corner is approached, causing the robot arm to decelerate at $t = 40$ seconds. At $t = 45$ seconds the robot comes to a full stop momentarily before accelerating in a new direction. The steady-state speed is achieved at $t = 50$ seconds and kept constant until the end of the simulation at $t = 80$ seconds. The basic flow reference signal can now be constructed as the one in figure 6-1.

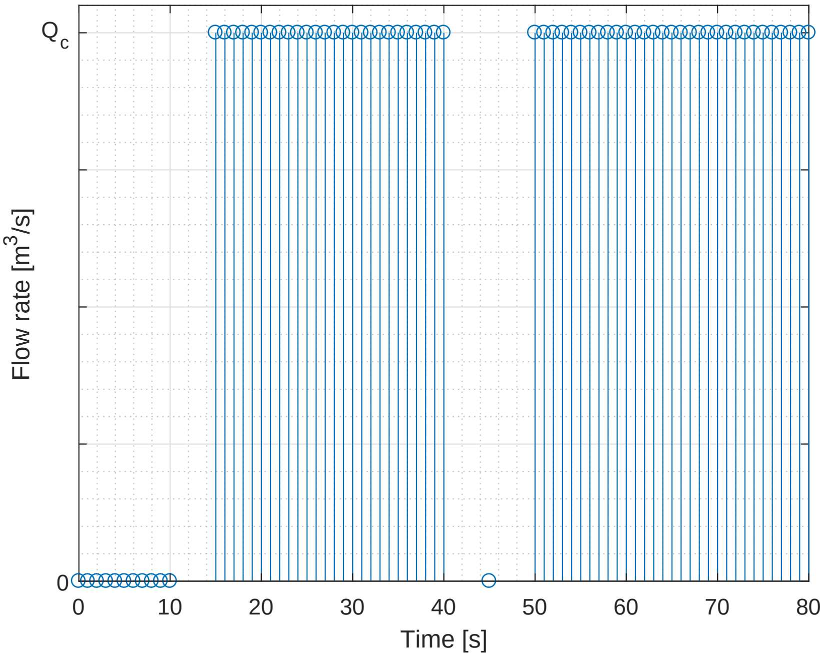


Figure 6-1: Basic flow reference signal

The shape of the function connecting the zero flow rate values to the constant flow rate Q_c is not known. Recall that these segments of the flow reference trajectory need to be tracked accurately to minimize the excess material depositioning. Since the true shape is unknown three plausible shapes are chosen to be used for simulation: a ramp function, an exponential function, and a sigmoid function. In figure 6-2 the shapes of the three proposed

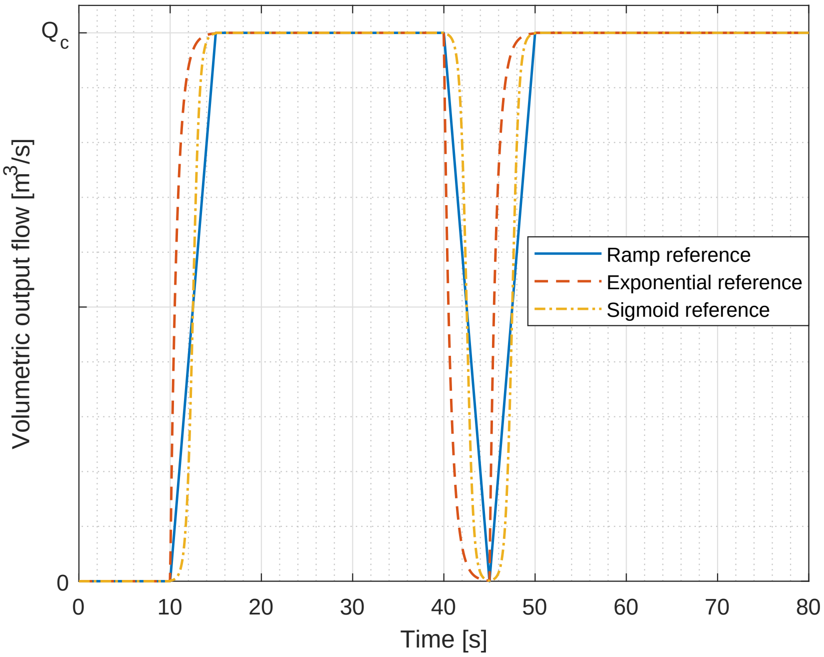


Figure 6-2: Proposed flow rate reference signals

flow rate reference signals can be seen. For simplicity, the three reference signals will be referred to as the ramp, exponential, and sigmoid reference signals, even though they each have a steady-state section. There is one problem: the volumetric flow rate is not measured, which makes tracking a reference signal impossible. Fortunately, the pressure is measured. Therefore, if the volumetric flow rate reference trajectory can be translated to a corresponding pressure reference trajectory then tracking will be possible. This approach essentially uses the pressure to track a predefined feedforward signal. Another problem is that the pressure model is nonlinear, while the linear model predictive control algorithm uses a linear model. Both problems and their solution will be discussed in the next section.

6-1-1 Translating the volumetric reference flow rate to its corresponding pressure reference

The effect of the motor input on the pressure, and the pressure on the volumetric flow rate, can be described by a block diagram as the one in figure 6-3. The rectangular blocks G_P and G_Q represent the SISO LTI models; the elipsoidal blocks W_P and W_Q represent the static output nonlinearities. The relation between the volumetric reference flow rate, which from now on will be denoted as Q_r, to the corresponding pressure reference signal P_r is given by the volumetric flow rate wiener model identified in chapter 4. The relation can be written as

$$Q_r = W_Q(G_Q P_r) \tag{6-1}$$

The signal P_r can be deduced by subsequently applying the inverse of the wiener block, W_Q^{-1}, and the inverse dynamical model, G_Q^{-1} to the known signal Q_r. The wiener block W_Q is

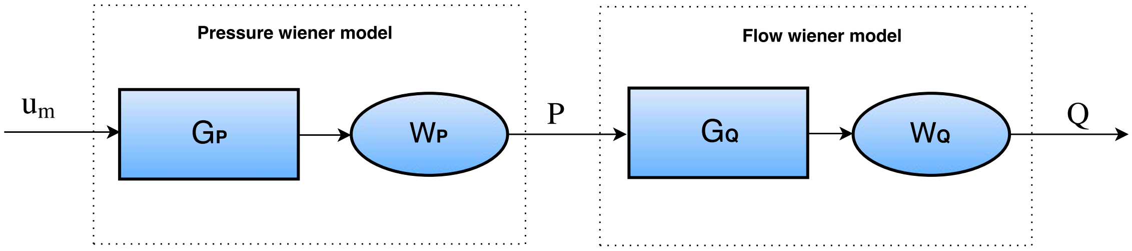


Figure 6-3: Block diagram describing the effect of the motor input on pressure and output flow rate

guaranteed invertible because its a strictly increasing function consisting of piecewise linear functions. The linear time-invariant block G_Q is described by the following discrete-time transfer function (with sample time $h_Q = 0.05$ seconds):

$$G_Q = \frac{\bar{Q}(k)}{P(k)} = \frac{1}{z - 0.9225} \tag{6-2}$$

Where $\bar{Q}$ is the output of the block G_Q, which will be called the linear flow rate for simplicity. For SISO LTI models the inverse is obtained by switching the numerator and denominator of the transfer function:

$$G_Q^{-1} = \frac{P(k)}{\bar{Q}(k)} = \frac{z - 0.9225}{1} \tag{6-3}$$

Which can be rewritten as a linear difference equation:

$$P(k) = \bar{Q}(k+1) - 0.9225\bar{Q}(k)$$

The inverse model is non-causal because the output $P(k)$ depends on future input values $\bar{Q}(k+1)$. Non-causal systems are not realizable in real life since the world is causal. Fortunately, however, because all future values of $\bar{Q}$ are known it is possible to construct the signal $P(k)$ for every instance k using the non-causal inverse model G_Q^{-1}. Therefore, given a flow rate reference Q_r it is possible to deduce its corresponding pressure sequence P_r by applying W_P^{-1} and G_Q^{-1} consecutively.

The second problem, transforming the pressure reference P_r to its linear pressure reference $\bar{P}_r$, can be solved by simply applying the inverse function of the wiener block of the pressure model, W_P^{-1} to P_r. Again, the function represented by the wiener block is strictly increasing and consists of piecewise linear functions, making it invertible.

Now that everything that is needed for the linear model predictive control algorithm is available, it is time to initialize the controller parameters, which will be done in the next section.

6-2 Model predictive controller synthesis

As state previously, a model predictive controller seems like the best choice for tracking an a priori known reference signal because the controller uses that knowledge in its control law.

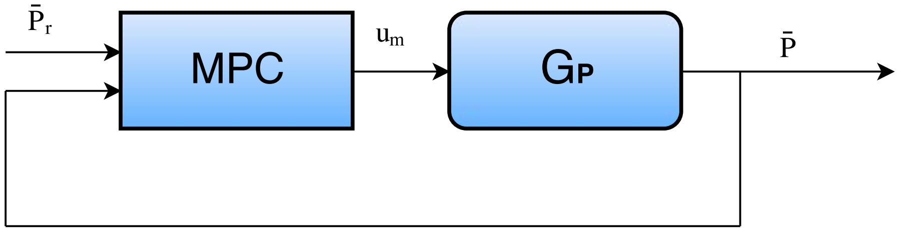


Figure 6-4: Linear Model predictive controller

Table 6-1: Tuning parameters for the synthesized model predictive controller

Parameter	**Value**
N_m	1
N	18
N_c	9
λ	0
$P(q^{-1})$	1

Another advantage is that the constraint on the motor input value is easily added to the controller. For a more detailed discussion on what MPC is and what its drawbacks and advantages are, see chapter 5.

Generalized-predictive-control (GPC) is chosen as a cost function, because the goal is to let the output track a reference signal. An inequality constraint on the motor input is added such that $0 \leq u_m \leq 1$. Now, only the tuning parameters need to be specified. They are chosen by following the rules of thumb presented in chapter 5, see table 6-1 for the numerical values. Obviously, the linear pressure reference $\bar{P}_r$ needs to be tracked, and therefore the linear pressure model G_P is used as prediction model. Now that all relevant MPC parameters are defined its to evaluate its reference tracking performance.

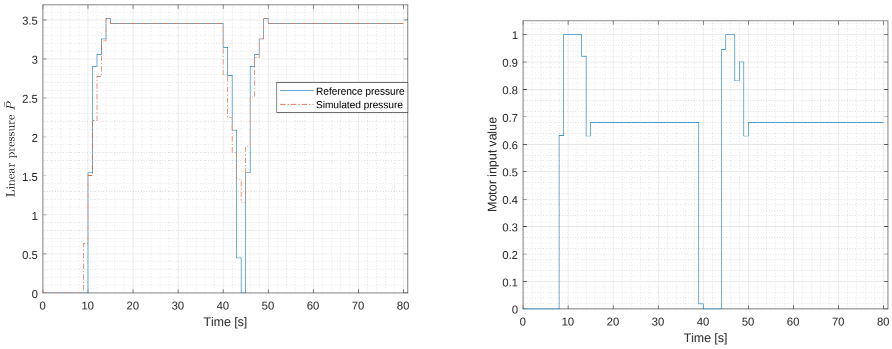
6-3 Reference tracking simulation

As previously discussed, the model predictive controller will try to track three different reference signals; namely a ramp, exponential, and sigmoid signal. The case where the desired constant volumetric flow rate Q_c equals 50% of the maximum achievable flow rate Q_{max} will be simulated. Before the simulations are performed it is needed to quantify the tracking error. For that purpose the normalized-root-mean-squared (NRMSE) fit percent is used since it provides a measure for how much two signals are alike. It is calculated as:

$$fit_{NRMSE} = 100 \cdot (1 - \frac{||y - \hat{y}||}{||y - mean(y)||}) \tag{6-4}$$

Following the steps described in the previous sections, the volumetric flow rate reference signal Q_r is reformulated as its corresponding linear pressure reference signal $\bar{P}_r$. Some comments have to be made regarding the loss of precision due to sampling. The flow rate reference signal is specified in continuous-time, but since the control system is digital and sampled at a frequency of one hertz, the linear pressure reference $\bar{P}_r$ only has values specified each second. This will cause some tracking error, since the flow rate changes continuously and the motor input values can only be changed at discrete time instants. The only way to decrease this error is by increasing the sampling rate. However, the sampling time cannot be smaller than the time it takes for the model predictive control algorithm to perform the optimization step. The pressure sampling time will be fixed at one second.

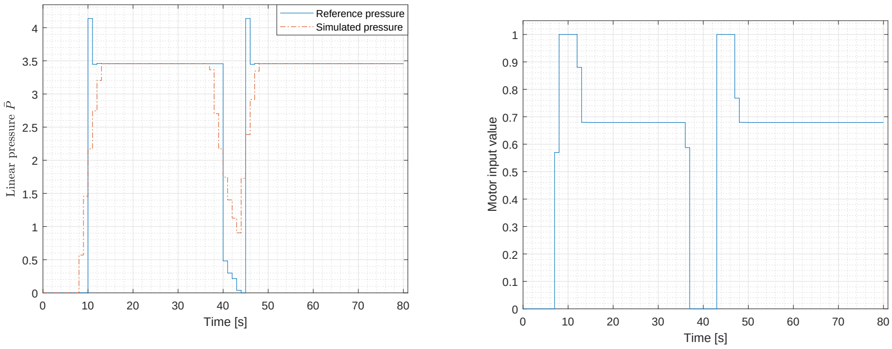
Linear pressure reference tracking



(a) Linear pressure reference and MPC simulation

(b) Corresponding motor input values

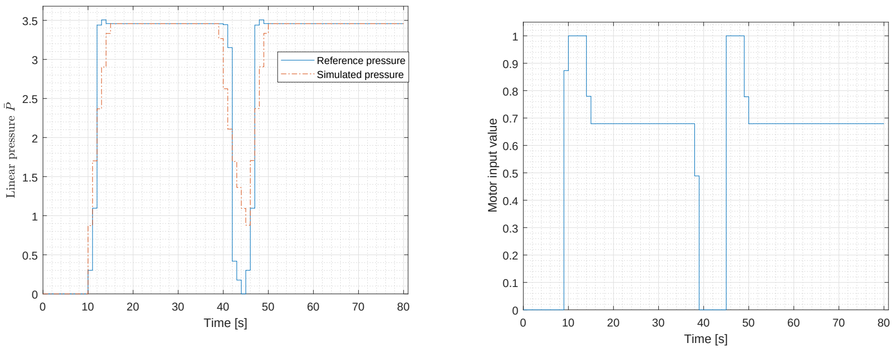
Figure 6-5: Linear pressure ramp reference tracking simulation



(a) Linear pressure reference and MPC simulation

(b) Corresponding motor input values

Figure 6-6: Linear pressure exponential reference tracking simulation



(a) Linear pressure reference and MPC simulation

(b) Corresponding motor input values

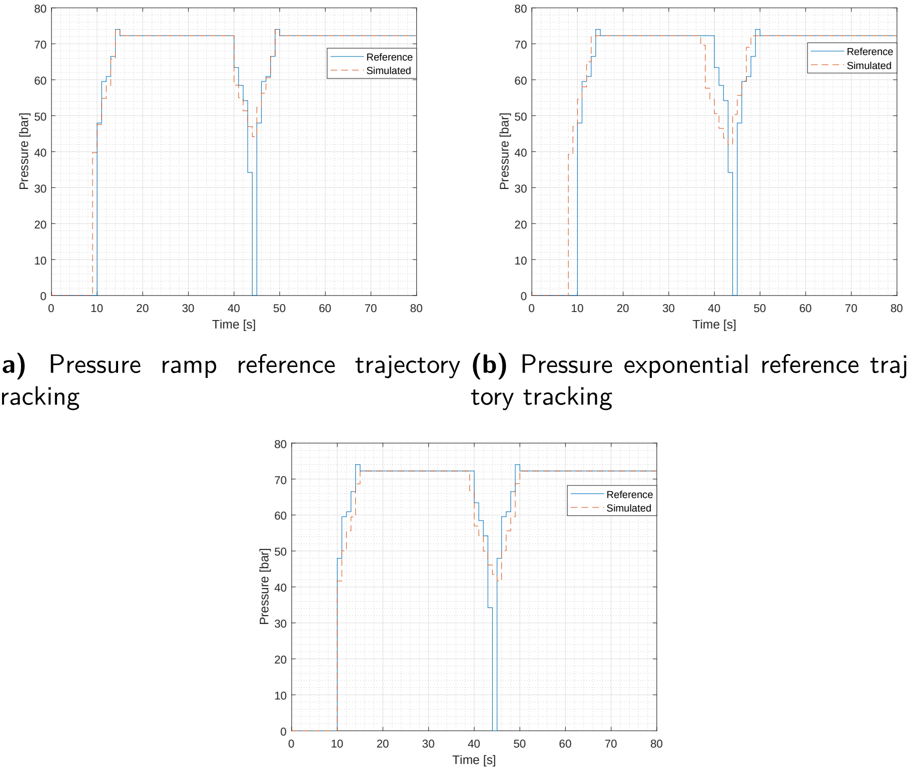
Figure 6-7: Linear pressure sigmoid reference tracking simulation

From the linear pressure reference tracking simulations (see figures 6-5, 6-6, and 6-7) it can be seen that the controller tries to perfectly track the signal but is unable to do so due to actuator saturation. See table 6-2 for the NRMSE fit percentages. Also, note how the MPC algorithm does not try to apply inputs larger than the constraints allow it to. The difficulty in tracking the reference lies in the large startup slopes and braking slopes at $t = 10$ seconds, $t = 40$ seconds, and $t = 45$ seconds. Due to the input constraints the system is not able to reduce the pressure faster than its free response. This results in the fact that none of three simulated linear pressure signals reach the zero point at $t = 45$ seconds. Note how the controller quickly reaches the steady-state sections without overshoot, and tracks them without error.

In the next section the obtained linear pressure simulation is transformed to its corresponding pressure value by applying the wiener block function W_P to $\bar{P}$ in order to investigate the effect of the wiener block of the pressure model on the tracking error.

Pressure reference tracking

By transforming the linear pressure reference tracking simulation results using the wiener pressure block W_p the signals are transformed to the pressure values that the system will achieve (see figure 6-8). The only effect that the wiener block can have is in the fact that the errors are transformed in a nonlinear way. And it seems that the tracking error of the exponential and sigmoid signals have increased more than the error of the ramp signal. Again, see table 6-2 for the NRMSE fit percentages. Obviously, the steady-state behaviour is obviously identical to the linear pressure situation since the wiener block is a static nonlinear function.



(a) Pressure ramp reference trajectory tracking

(b) Pressure exponential reference trajectory tracking

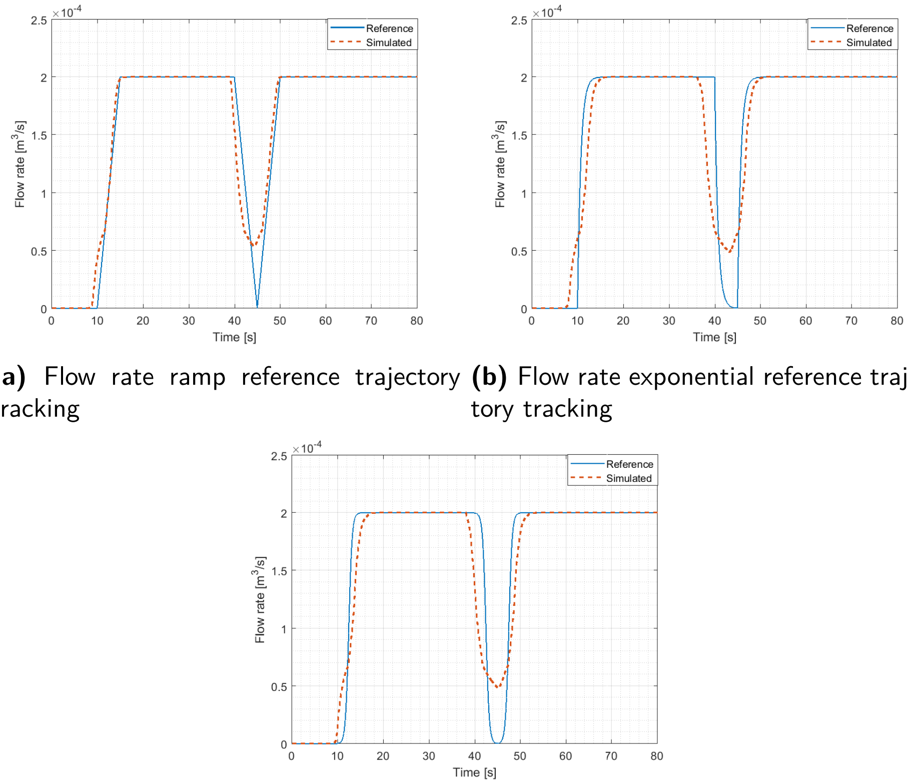
(c) Pressure sigmoid reference trajectory tracking

Figure 6-8: Pressure reference tracking simulation results

Table 6-2: NRMSE values for each reference signal

	Linear pressure fit	**Pressure fit**	**Flow rate fit**
Ramp reference signal	81.6%	72.3%	80.8%
Exponential reference signal	62.8%	60.9%	65.9%
Sigmoid reference signal	73.1%	77.0%	68.2%

Flow rate reference tracking



(a) Flow rate ramp reference trajectory tracking

(b) Flow rate exponential reference trajectory tracking

(c) Flow rate sigmoid reference trajectory tracking

Figure 6-9: Flow rate reference tracking simulation results

Finally, the most important question to ask is how well the flow rate reference is tracked; see figure 6-9 for a visual inspection, and again table 6-2 for the NRMSE fit percentages. Notice how the ramp flow reference signal is tracked best. This can be explained by the fact that the constant acceleration and deceleration slope are easiest to maintain if it is attainable by the system inputs. The exponential and sigmoid function both have segments which are less steep than the other segments of the respective function. To reach the same steady-state value, this lack of steepness has to be compensated by an increased slope elsewhere. Notice how the controller sends input signals to the system before actual changes in the reference

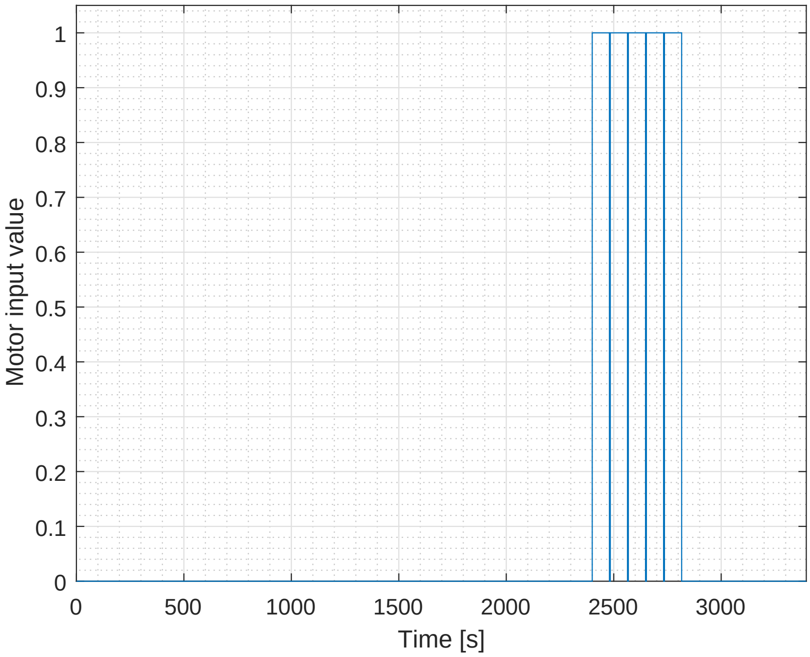


Figure 6-10: Motor disturbance signal used for simulation

signal are measured. This is one of the main advantages of MPC, and decreases the overall tracking error.

In the next section a model predictive controller controlling the extruder temperature is simulated, where the motor input acts as a measured disturbance.

6-4 Model predictive control for temperature regulation

The model predictive control strategy can also be implemented to track a reference temperature for the extruder, since a linear model of the temperature is available. The controller is tuned following the rules of chapter 5, and the inequality constraints on the heater are defined as $0 \leq u_h \leq 1$. The situation is simulated where the temperature reference signal is not known a priori, because an operator might wants to change it during the printing process. Furthermore, the machine is first brought up to temperature, and then the motor acts as a disturbance when the printing process starts. The worst-case scenario is simulated which means that u_m switches between being totally on, to being totally off when the robot arm moves in a new direction, see figure 6-10. The resulting temperature response and heater signal values can be seen in figure 6-11. Note that the disturbance is counteracted by the heaters accordingly, and that no serious perturbation of the reference temperatures occur. In fact, one must zoom in a lot to even see the effect, as can be seen in figure 6-12.

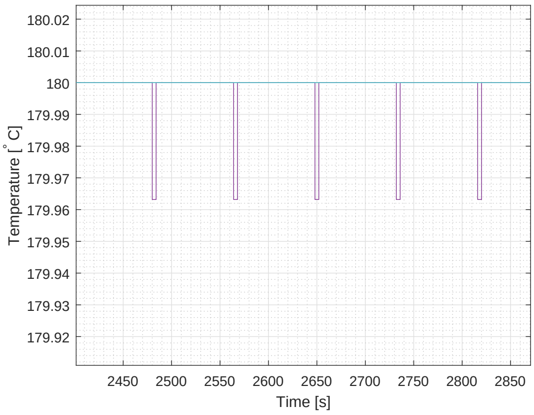
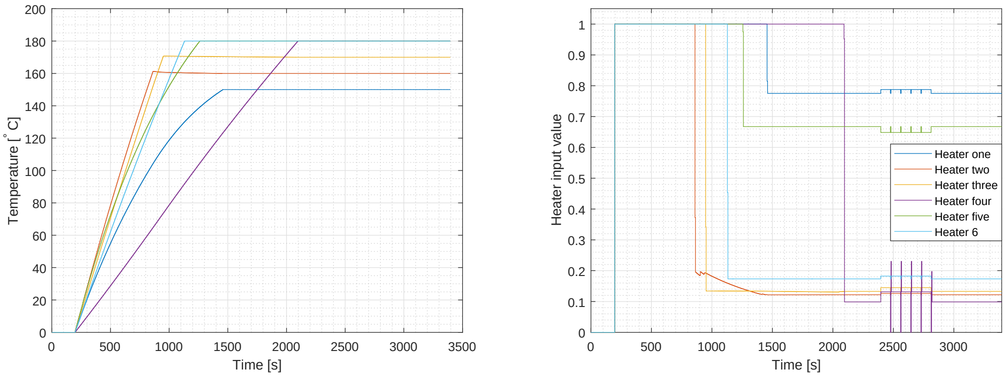


Figure 6-12: Zoomed in temperature disturbance



(a) Temperature simulation study

(b) Accompanying heater values

Figure 6-11: Temperature simulation

The temperature simulation study showed that the model predictive controller can reject a step disturbance of the motor input adequately. This is in part due to the relatively small effect the motor input disturbance has on the temperature, see the model derived in chapter 3. Even though the model predictive controller rejects the disturbance, it must first measure an error before action is undertaken. In the next section a feedforward method is explained which infuences the heater inputs as soon as the motor input is changed, because then the disturbance is rejected before the effect is even measured.

Using feedforward to counteract the influence of the motor input on the temperature

Recall that the temperature dynamics of the extruder, hose, and nozzle can be written as a linear time-invariant state-space equation, where the states represent the temperature values, inputs one through six are the heater inputs, and input seven is the motor input. To see how to counteract the effect of the motor input on the temperature, rewrite the state-space equation as:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ &= Ax(k) + \begin{bmatrix} \bar{B} & B_7 \end{bmatrix} \begin{bmatrix} \bar{u} \\ u_7 \end{bmatrix} \end{aligned}$$

Where $\bar{B}$ is a six by six invertible matrix (see chapter 3) representing the effect of the heater inputs on the temperature, and B_7 represents the effect of the motor input on the temperature. The idea is to use a feedforward controller which affects the heater inputs $\bar{u}$ such that the effect of the motor input u_7 on the temperature is removed. This is easily accomplished by setting

$$\bar{u} = \bar{u}_{fb} - \bar{B}^{-1} B_7 u_7 \tag{6-5}$$

Where $\bar{u}_{fb}$ is the heater input of the feedback controller used to regulate the temperature [28].

6-5 Discussion

In the previous sections it was seen that the synthesized model predictive controller had a good steady-state behaviour for linear pressure control, but struggled during the rapid change of moving from the steady-state value to zero and vice versa. This struggling was caused by actuator saturation, and the fact that no negative pressures could be generated by the system. Therefore, to improve the reference tracking accuracy changes should be made to the system itself. For one, if the range of the allowed motor input signal is increased then steeper reference signals can be tracked. Another improvement would be if the motor input is allowed to take on negative values, which can be interpreted as having the screw spinning in the opposite direction. This could then decrease the output flow more quickly than the free response currently allows.

Furthermore, the current approach is essentially trying to make the pressure track a predefined feedforward signal for controlling the flow rate. One of the drawbacks of feedforward control is that the system model must be very accurately known, which is definitely not the case since it has been identified from CFD simulation software data. If the goal is to let the volumetric flow rate track a reference signal accurately then it is imperative that the flow rate is measured in real-time. That is the only way in which any disturbances that might occur to the flow rate will be noticed and can be compensated.

The disturbance rejection properties of the temperature reference tracking model predictive controller are deemed adequate. The worst-case scenario, when the motor input is described by the unit step function, was simulated. It was shown that the disturbance effect is quickly counteracted by the controller by changing the heater input values. Howver, the controller could only counteract the disturbance after the resulting tracking error was measured. Therefore, a feedforward approach for disturbance rejection was presented.

The next chapter provides an overview of the conclusions and recommendations made in this thesis.

Chapter 7

Conclusions and recommendations

The goal of this thesis was to synthesize a controller to eliminate volumetric flow rate variations for a large-scale 3D printer. In order to achieve this, the large-scale 3D printer was divided in three subsystems which were to be modeled; namely the temperature model, the pressure model, and the flow rate model. Experiments were performed to quantify the effect of the heaters and motor speed on the temperature and pressure of the system. The data obtained from the experiments were used in conjunction with system identificatin techniques to obtain system models; where the temperature model is described by a MIMO linear state-space model, while the SISO pressure model is described by a wiener model with a strictly increasing piecewise linear function as static output nonlinearity. Since the flow rate of the machine is not measured, a computational fluid dynamics model was created to describe the effect of pressure on the flow rate. The obtained simulation data is then used for model simplification by estimating a wiener model to fit the data. Finally, the models were used in conjunction with a linear model predictive control algorithm to simulate the tracking of three reference volumetric flow rates; shaped as a ramp, exponential, and sigmoid function. The model predictive controller tracked the ramp reference signal best, this is due to the fact that the ramp signal has a constant slope. It was seen that the linear pressure model predictive control tracking error is transformed in a non-linear way due to the static output non-linearities of the wiener models. The disturbance rejection properties of the model predictive controller for temperature tracking was investigated by using the maximum value of the motor input as a disturbance signal. It was shown to have a negligble effect on the temperature response of the system.

7-1 Recommendations for future work

The reason that perfect flow rate reference tracking was not achieved is that the motor input saturated for each of the three reference trackign signals. To circumvent this the motor input should be allowed to take on higher values to avoid actuator saturation. Furthermore, the time it takes for the flow rate to be stopped is determined by the free response of te system since the motor currently can only push the flow. This could possibly be improved by allowing the motor to rotate in the opposite direction to decrease the flow rate.

Another important recommendation is to install a flow rate measurement device. Currently, the pressure reference tracking acts as a feedforward signal for the feedforward flow model. This means that any disturbances acting on the flow rate are not measured and can therefore not be removed. If the flow rate is measured a feedback controller can be implemented. Furthermore, the flow rate model is derived entirely by first principles, without any actual measurements to validate the model by comparing results with the real-life scenario.

Bibliography

[1] H. Hamane, K. Matuki, F. Hiroki, and K. Miyazaki, “Thermal mimo controller for set-point regulation and load disturbance rejection,” *Control Engineering Practice*, vol. 18, pp. 198–208, 2010.

[2] F. M. White, *Fluid mechanics Seventh Edition in SI Units*. McGraw-Hill, 2011.

[3] S. Misik, A. Cela, and Z. Bradac, “Optimal predictive control - a brief review of theory and practice,” *International Federation of Automatic Control*, vol. 49, no. 25, pp. 324–329, 2016.

[4] J. P. Kruth and M. C. Leu and T. Nakagawa, “Progress in additive manufacturing and rapid prototyping,” *CIRP Annals - Manufacturing Technology*, vol. 47, no. 2, pp. 525–540, 1998.

[5] O. Diegel, “Additive manufacturing: An overview,” *Comprehensive Materials Processing*, vol. 10, pp. 3–18, 2014.

[6] R. Singh and S. Singh, “Additive manufacturing: An overview,” *Reference Module in Materials Science and Materials Engineering*, 2017.

[7] R. Singh and H. K. Garg, “Fused deposition modeling - a state of the art review and future applications,” *Reference Module in Materials Science and Materials Engineering*, 2016.

[8] R. Anitha and S. Arunachalam and P. Radhakrishnan, “Critical parameters influencing the quality of prototypes in fused deposition modelling,” *Journal of Materials Processing Technology*, vol. 118, pp. 385–388, 2001.

[9] D. Horvath, R. Noorani, and M. Mendelson, “Improvement of surface roughness on abs 400 polymer materials using design of experiments,” *The sixth Pacific Rim International Conference On Advanced Materials and Processing*, 2007.

[10] T. Nancharaiah, R. Raju, and V. Ramach, "An experimental investigation on surface quality and dimensional accuracy of fdm components," *International Journal on Emerging Technologies*, pp. 106–111, 2010.

[11] C. Bellehumeur and L. Li and Q. Sun and P. Gu, "Modeling of bond formation between polymer filaments in the fused deposition modeling process," *Journal of Manufacturig Processes*, vol. 6, no. 2, pp. 170–178, 2004.

[12] S. Ahn, "Anisotropic material properties of fused deposition modeling abs," *Rapid Prototyping Journal*, vol. 8, pp. 248–257, 2002.

[13] S. Song and A. Wang and Q. Huang and F. Tsung, "Shape deviation modeling for fused deposition modeling processes," in *2014 IEEE International Conference on Automation Science and Engineering*, CASE, 2014.

[14] C. Bayley, L. Bochmann, C. Hurlbut, M. Helu, R. Transchel, and D. Dornfeld, "Understanding error generation in fused deposition modeling," *ASPE - Spring Topical Meeting*, vol. 57, pp. 98–103, 2014.

[15] T. A. Savvas, N. C. Markatos, and C. D. Papaspyrides, "On the flow of non-newtonian polymer solutions," *Applied Mathematical Modelling*, vol. 18, no. January, pp. 14–22, 1994.

[16] C. Rauwendaal, *Polymer Extrusion.* Postfach 86 04 20, 81631 MÃijnchen, Germany: Hanser, 4th ed., 2001.

[17] L. Ljung, "Perspectives on system identification," *Proceedings of the 17th World Congress of The International Federation of Automatic Control*, 2008.

[18] M. Verhaegen and V. Verdult, *Filtering and system identification: a least squares approach.* Cambridge university press, 2007.

[19] Ton J. J. van den Boom and Bart de Schutter, *Optimization in systems and control lecture notes.* 2014.

[20] John F. Wendt, *Computational Fluid Dynamics: An Introduction.* Springer., 2009.

[21] H. H. J. Bloemen, T. J. J. van den Boom, and H. B. Verbruggen, "Model-based predictive control for hammerstein-wiener systems," *International Journal of Control*, vol. 74, no. 5, pp. 482–495, 2001.

[22] M. Lawrynczuk, "Nonlinear predictive control for hammerstein-wiener systems," *ISA Transactions*, vol. 55, pp. 49–62, 2015.

[23] S. N. Huang, K. K. Tan, and T. Lee, "Adaptive gpc control of melt temperature in injection moulding," *ISA Transactions*, vol. 38, pp. 361–373, 1999.

[24] Z. Jiang et al., "Polymer extrusion: From control system design to product quality," *Industrial and Engineering Chemistry Research*, vol. 151, pp. 14759–14770, 2012.

[25] D. W. Clarke, C. Mohtadi, and P. S. Tuffs, "Generalized predictive control - part 1. the basic algorithm," *Automatica*, vol. 23, no. 2, pp. 137–148, 1987.

[26] D. W. Clarke, C. Mohtadi, and P. S. Tuffs, "Generalized predictive control - part 2. extensions and interpretations," *Automatica*, vol. 23, no. 2, pp. 149–160, 1987.

[27] Ton J. J. van den Boom, *Model Predictive Control lecture notes.* 2013.

[28] K. J. Astrom and B. Witenmark, *Computer controlled systems theory and design.* Dover Publications, Inc., 2011.

Appendix A

Glossary

List of Acronyms

AM	Additive Manufacturing
CAD	Computer Aided Design
DoF	Degrees of Freedom
FDM	Fused Deposition Modeling
AC	Alternating Current
CFD	Computational Fluid Dynamics
GBN	Generalized Binary Noise
MIMO	Multiple Inputs Multiple Outputs
LQ	Linear-Quadratic
LTI	Linear Time Invariant
MPC	Model Predictive Control
GPC	Generalized Predictive Control
LQPC	Linear Quadratic Predictive Control
PPG	PolyPropylene with Glass
SISO	Single Input Single Output
PID	Proportional Integral Derivative
NRMSE	Normalized-Root-Mean-Squared-Error
VAF	Variance-Accounted-For