

Evaluation of Model-Based Design Using Rapid Control Prototyping on Forklifts

Lovisa Jansson and Amanda Nilsson

Master of Science Thesis in Electrical Engineering

Evaluation of Model-Based Design Using Rapid Control Prototyping on Forklifts:

Lovisa Jansson and Amanda Nilsson

LiTH-ISY-EX--19/5229--SE

Supervisor: **Alberto Zenere, PhD Student**
ISY, Linköpings universitet
Daniel Fahlén, Engineer
Toyota Material Handling
Patrick Blomqvist, Group Manager
Toyota Material Handling

Examiner: **Martin Enqvist, Associate Professor**
ISY, Linköpings universitet

*Division of Automatic Control
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping, Sweden*

Copyright © 2019 Lovisa Jansson and Amanda Nilsson

Sammanfattning

Syftet med detta examensarbete är att utvärdera Rapid Control Prototyping vilket är en del av modellbaserad utveckling som gör det möjligt att enkelt testa modeller av styralgoritmer direkt på det riktiga systemet. Utvärderingen är gjord genom att testa två olika regulatorer, en P-regulator med parameterstyrning och en linjär modellbaserad prediktionsregulator (MPC), för sänkningen av gafflarna på en truck.

De två regulatorerna testas först i en simuleringsmiljö. I arbetet används två olika simuleringsmodeller: en fysikalisk där endast mindre parameterjusteringar görs och en estimerad black-box modell. Efter att regulatorerna utvärderas i simuleringsmiljön testas de även på en riktig truck med hjälp av automatisk kodgenerering och exekvering på en dedikerad hårdvaruplattform.

De konstruerade regulatorerna har olika för- och nackdelar eftersom en är olinjär och envariabel, P-regulatorn, och en är linjär men flervariabel, MPC:n. P-regulatorn har en mjuk rörelse i alla lägen utan att bli för långsam, till skillnad från MPC:n. Nackdelen med P-regulatorn, jämfört med MPC:n är att det inte finns någon garanti för att P-regulatorn håller hastighetsbegränsningen som MPC:n gör.

P-regulatorns bättre prestanda överväger garantin om att hålla hastighetsbegränsningen och därför dras slutsatsen att olinjäriteterna i systemet överväger effekterna av det faktum att det också är flervariabelt. En annan slutsats är att modellbaserad utveckling och Rapid Control Prototyping gör det möjligt att testa flera olika idéer på en riktig gaffeltruck utan att spendera för mycket tid på implementationen.

Abstract

The purpose of this thesis is to evaluate Rapid Control Prototyping which is a part of the Model-Based Design concept that makes it possible to conveniently test prototype control algorithms directly on the real system. The evaluation is here done by designing two different controllers, a gain-scheduled P controller and a linear Model Predictive Controller (MPC), for the lowering of the forks of a forklift.

The two controllers are first tested in a simulation environment. The thesis contains two different simulation models: one physical where only minor parameter adjustments are done and one estimated black-box model. After evaluating the controllers in a simulation environment they are tested on a real forklift with a real-time target machine.

The designed controllers have different strengths and weaknesses as one is non-linear and single variable, the P controller, and the other linear and multivariable, the MPC. The P controller has a smooth movement in all situations without being slow, unlike the MPC. The disadvantage of the P controller compared to the MPC is that there is no guarantee that the P controller will keep the speed limit, whereas the MPC approach gives such a guarantee.

The better performance of the P controller outweighs the speed limit guarantee and thus a conclusion is drawn that the nonlinearities of the system has a larger effect than the multivariable aspect. Also, another conclusion drawn is that working with Model-Based Design and Rapid Control Prototyping makes it possible to test many different ideas on a real forklift without spending a lot of time on implementation.

Acknowledgments

We would like to thank our supervisor from Linköping University, Alberto Zenere, and our examiner, Martin Enqvist, for their invaluable ideas and feedback. We would also like to thank our supervisors at Toyota Material Handling, Patrick Blomqvist and Daniel Fahlén, for their useful inputs. A special thanks to Daniel Fahlén for sharing his time, support and knowledge. Another special thanks to Patrik Hallsjö for taking time to proofread our report. Furthermore, we would like to thank all electronics, hydraulics and software experts at Toyota who have helped us during our work.

Linköping, May 2019
Lovisa Jansson och Amanda Nilsson

Contents

Notation	xi
1 Introduction	1
1.1 Background	2
1.2 Objective	2
1.3 Model-Based Design	3
1.3.1 Rapid Control Prototyping	3
1.3.2 RCP and Lean	4
1.4 Limitations	5
1.5 Literature Study	5
1.6 Thesis Outline	6
2 System Overview	7
2.1 Mast	8
2.2 Hydraulics	9
2.3 Electrical Subsystem	10
2.3.1 Sensors	10
2.3.2 Control System	10
2.4 Speedgoat Real-Time Target Machine	11
3 System Identification	13
3.1 Data Collection	13
3.2 Simulation Models	14
3.2.1 Physical Model	14
3.2.2 Nonlinear ARX Model	18
3.3 Internal Model	20
3.4 Comparison of Models	22
3.5 Discussion	23
3.5.1 Simulation Models	23
3.5.2 Internal Models	24
4 Control Design	25
4.1 Reference Trajectory	25

4.1.1	Integrated Velocity	26
4.1.2	Minimal Jerk	26
4.2	P Controller	28
4.2.1	Control Error	28
4.2.2	Gain Scheduling	29
4.2.3	Transfer Region	29
4.2.4	Implementation	30
4.3	MPC	31
4.3.1	Goal Function	31
4.3.2	Conditions	32
4.3.3	Design	32
4.3.4	Observer	33
5	Results	35
5.1	P Controller	35
5.2	MPC	39
5.3	Comparison of Controllers	42
5.4	Comparison of Simulation Models	47
6	Discussion	51
6.1	Models	51
6.2	Controllers	52
6.2.1	P Controller	52
6.2.2	MPC	52
6.2.3	Comparison of Controllers	53
6.3	Rapid Control Prototyping	54
6.4	Future Work	55
6.5	Final Conclusion	56
A	Measurement Data	59
B	Validation Data	65
	Bibliography	69

Notation

ABBREVIATION

Abbreviation	Meaning
AIC	Akaike Information Criterion [7]
ARX	Autoregressive Exogenous [7]
CAN	Controller Area Network [13]
MBD	Model-Based Design
MCU	Main Controller Unit
MPC	Model Predictive Control [4]
PWM	Pulse Width Modulation
RCP	Rapid Control Prototyping
RRE	Reach Rider Electrical
SCU	Secondary Control Unit
SS	State Space
TMH	Toyota Material Handling

1

Introduction

This thesis focuses on how to use Rapid Control Prototyping (RCP) when implementing control systems on forklifts. This is evaluated by implementing two different control methods of the lowering function on a Toyota Material Handling (TMH) Reach Rider Electrical (RRE) forklift, see Figure 1.1, using RCP methodology and a real-time target machine from the company Speedgoat [23].



Figure 1.1: The RRE forklift [9].

1.1 Background

Today's engineers are expected to design more complex systems and to shorten the development time. One way to speed up this process is to use Model-Based Design (MBD) which makes it possible to continuously test new ideas in an early stage. More on details of use and implementation can be found in Section 1.3.

Today the lowering function of the RRE forklift is controlled by an open loop. Therefore the system is designed in a conservative manner. More precisely, the speed of the forks depends on the temperature of the oil, which changes during operation, and the load on the forks. Since the control system does not contain any feedback, the control parameters are adjusted in a conservative manner to fit the system when the oil has reached its maximum temperature and when the load has maximum weight. This leads to lower maximum speed when the oil is cold or when there is a low weight on the forks. Therefore, there is room for improvement using a closed-loop system.

Another difficulty is the region where the forklift switches from using the main lift to using the free lift, described in Section 2.1, which is called the transfer region. The main lift should be near its end position before the free lift kicks in. This is to get a low mast height as quickly as possible. If the main lift reaches its end position before the free lift starts the forks stop. This creates a jerky motion before the forks continue with the free lift, and also results in an unnecessary decrease in speed.

Other interesting aspects of the fork movement are when the forks stop in mid-air and at the bottom. In both these cases it is important to stop with as little jerk as possible.

The free lift and the main lift are controlled by one hydraulic valve each, making the system multivariable. These two valves are also nonlinear. Both these aspects make the system harder to control than if it was linear and single variable. Using RCP as a method to quickly try different kinds of control algorithms has the potential to significantly improve the performance of the lowering function.

1.2 Objective

The objective of this thesis is to test a platform for RCP and use it to implement control strategies for some function of the RRE forklift. The platform used for implementation will be a Speedgoat real-time target machine and the focus will be on the function for lowering the forks.

Regarding control strategies, one simpler method (P-controller) and one more advanced (MPC) method will be implemented.

In particular, the questions to be answered in this thesis are:

1. What control strategy can be used for the lowering function of an RRE forklift?

2. Is RCP a useful inclusion in the development process for TMH?

1.3 Model-Based Design

Model-Based Design (MBD) is a design methodology that focuses on working with models for the entire development process. A model of the entire system is created and used when developing software for the real system.

There are many advantages of working with models, such as:

- It is easy to oversee how the system works.
- It is easy to build modular models, which makes it easy to change or replace small parts of the system.
- It is possible to start with a simple model and increase complexity over time.

The first point is explained by that, for many, models are easier to understand than code. New concepts can be shown in a simple way and it is a way to get a fast understanding of a system. The second point implies that when designing models in a modular way, it is easy to add new features and test making smaller or larger changes in the system without having to redo large parts of the code. A consequence of this is the last point. When starting to implement a completely new concept it is both possible and fairly easy to start simple and then later develop it further. The model of a system can first be implemented on a very low level, only capturing the most basic features, and then simulated. Smaller parts of the system can then later on be replaced with parts containing more and more features. This process also creates a greater understanding of the system which makes unexpected errors arise early in the process. More about Model-Based Design, its advantages and how to implement it can be found in [1].

The conventional workflow when using C-programming, which is the workflow used at TMH today, can be seen in Figure 1.2a. When using this workflow the steps including programming, testing code on the forklift and debugging on the forklift take a lot of time. Therefore less time is left to be spent on algorithm design. This can be improved using MBD, with workflow as in Figure 1.2b. With this workflow it is easier to concentrate on the algorithm design since the smaller loop with only algorithm design and algorithm testing in simulations can be used first. When the result is satisfying in simulations the code can be generated and integrated on the forklift.

1.3.1 Rapid Control Prototyping

One of the steps in Model-Based Design is Rapid Control Prototyping (RCP), which is illustrated in Figure 1.2b as the loops closed with the dashed arrows. The smaller loops are performed as many times as needed to get a satisfying result before returning to the larger loop.

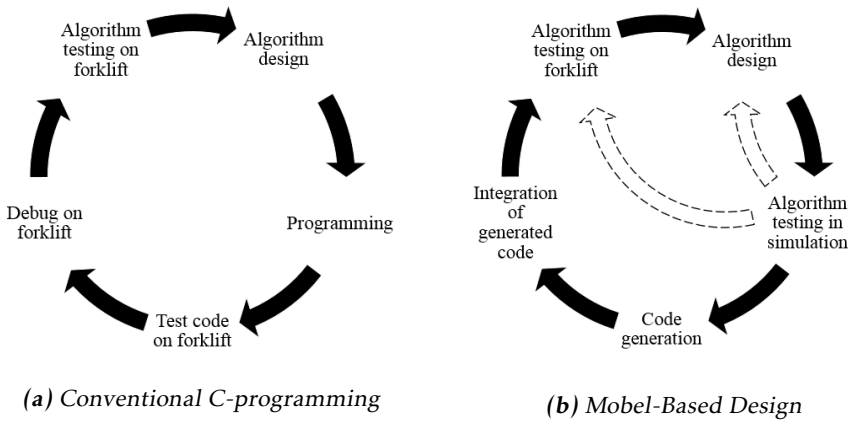


Figure 1.2: Workflow using conventional C-programming to the right and MBD to the left.

RCP makes it possible to test many different kinds of control systems without having to spend a lot of time writing code. This is done by building models of the control algorithm and then simulating these models directly on the hardware. Using RCP removes most limitations in the capacity of the hardware, such as memory and computing power. This makes it possible to choose hardware based on control algorithm and not the other way around.

1.3.2 RCP and Lean

The MBD fits into the Lean development method [21]. Lean is based on two core values, respecting the people and continuous improvement. Some of the key points in Lean development that are specifically used in RCP are:

- Amplify learning
- Eliminate waste
- Decide as late as possible

Working with RCP amplifies learning by creating shorter development cycles which creates more space for trying new ideas. The developer can also get immediate feedback when trying new ideas. The simplicity of trying different ideas also creates a greater understanding of the system for the developer.

By quickly being able to try new ideas, the waste of time can be reduced by eliminating ideas that do not work early. This leaves room for ideas that are more likely to work.

The flexibility of working with models makes it possible, as described above, to try many different strategies even on the hardware. This allows us to make the

decision on which model to finally implement very late in the development process.

1.4 Limitations

The practical implementation of RCP is limited to the use of a mobile real-time target machine from Speedgoat as hardware for computations and an RRE forklift for testing.

The software that is used for modelling and creating control systems is Mathwork's Matlab and Simulink. The reason for this is that Speedgoat only supports Matlab/Simulink.

The control is limited to the hydraulic system that controls the lowering function of the forks. Also, the signals used for feedback are limited to the existing sensors that measure height, pressure and temperature.

1.5 Literature Study

RCP is a method to test a control system on the real system as described in [12]. The article also states that to use RCP, a model of the system is required. The model is used to first simulate the system when a control strategy is designed. When the results are acceptable the same model of the control strategy is used together with RCP hardware and the real plant to test the control strategy. The article also states the benefits of this development method, where the primary one is a shortened development time.

A very common use of RCP is in engine control of vehicles. One article on this matter is [10]. In this article the process of working with RCP is described from beginning to end. The problems mentioned in this article are close to those in control systems for forklifts. The development process is also similar. However, there is a difference in the used hardware and the control problem.

Another application where RCP has been used is in control systems for a synchrotron beamline [19]. It is one of the few examples found on using the real-time target machine from Speedgoat for RCP. Unlike applications in the automotive industry, the control problems contain completely different difficulties. The main benefit of using RCP here is to reduce the time using the beam.

Several books about control design will be used as inspiration in the process of designing the controllers [4, 6]. They contain information on how to construct the controllers and how to choose the design parameters. Another useful book contains a general description of how a hydraulic system works and how to model and control a hydraulic system [18].

Many similar theses focusing on load handling control have previously been done in cooperation with TMH. In [5] the lowering function is modelled and then controlled with a PID controller. In this thesis a look-up table is used to compensate

for the nonlinear behavior. Another thesis on the subject is [2]. Instead of focusing on the lowering function, this thesis focuses on the lifting function and the control system is designed for a different kind of forklift than the one used in this thesis. In [2] a model is created and used for trying two different control strategies, PID control and MPC. Also, an extended Kalman filter is used to get better estimates of the states.

The movement of the forks must be smooth so that the load does not take damage or fall off. Minimizing the jerk, as described in [14], is one way to accomplish a smooth movement. One way to minimize the jerk is to use an MPC. In [20] an MPC is used for the motion of a vehicle while also constraining the jerk below a certain threshold.

An example of using MPC in combination with RCP and a real-time target machine from Speedgoat is [24]. In this example an autonomous kite is controlled to move in a trajectory of an eight by minimizing deflecting movements in the MPC.

Work has been done on how to use PID control for multivariable systems. One way to do this is to use decentralized PID control as presented in [11]. The article also presents a tuning method. Another example of how to get around the problem of nonlinear hydraulic systems is presented in [15]. Here, adaptive tuning of control parameters based on models are presented. The control system is implemented in Matlab/Simulink and run on a hydraulic system with dSPACE [3] hardware.

1.6 Thesis Outline

An overview of the system is presented in Chapter 2, System Overview. It includes the mechanical, hydraulic and electrical parts of the system and also how the mobile real-time target machine is connected.

Chapter 3, System Identification, presents the different models used in this thesis, two nonlinear models and two linear models. The nonlinear models are used for simulation and the linear as internal models in the MPC. The theory behind the models is presented and all the models are validated and briefly discussed.

Chapter 4, Control Design, presents the theory and implementation of the P controller and MPC respectively. The two controllers are compared with different references, models and against each other and the controller currently implemented on the forklift in Chapter 5, Result. This chapter also contains a comparison of the simulation models.

The results of the models, controllers and RCP are discussed in Chapter 6, Discussion. This chapter also presents future work that can be done on the subject. Finally, the questions in Section 1.2 are answered.

2

System Overview

The forklift used in this project is a reach-truck [9]. It has the ability to lift a maximum of 2000 kg on the forks up to 7 m and 1100 kg up to 11 m. The different parts of the system are presented in Figure 2.1. The block that contains the pump and the oil tank is grey since it is not of interest in this thesis. The other parts are described further in this chapter.

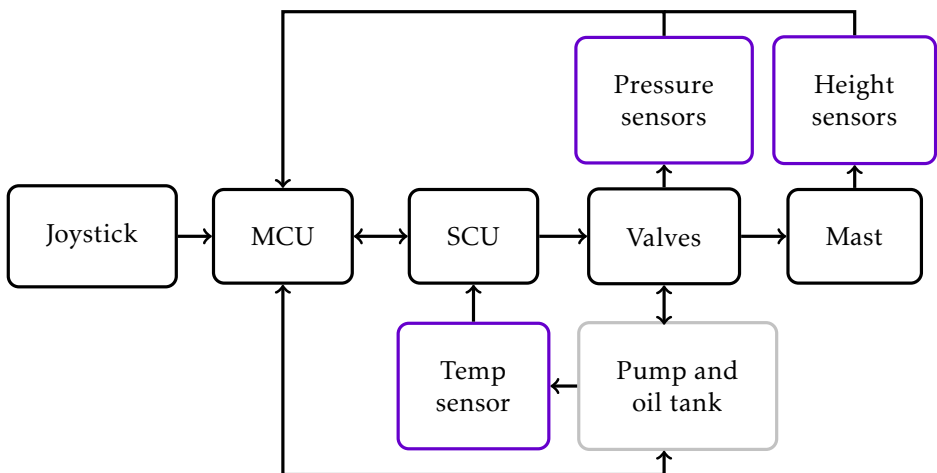


Figure 2.1: Overview of the system including electrical, hydraulic and mechanical parts.

2.1 Mast

The mast of the forklift is mounted as in Figure 2.2. It consists of three parts: the base, the first stage and the second stage. The lifting and lowering function can be divided into two parts, the free lift and the main lift, which behave in different ways. The free lift is defined as the forks sliding up and down the second stage of the mast without moving any part of the mast. The main lift on the other hand expands the whole mast by splitting and lifting the different stages as seen done in Figure 2.2 where the main lift has lifted the first and second stage. The lifting and lowering of the forks and the mast parts are done by the cylinders pulling or easing the chains respectively. Not having to expand the mast to lift the forks unless the forks are at the top of the mast makes it possible to both lift high and handle load in sections with lower ceiling height.

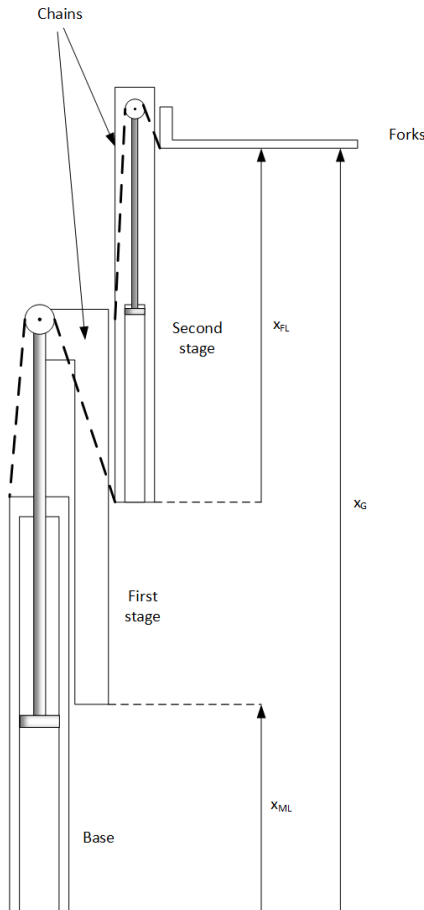


Figure 2.2: Schematic picture of the mast. [5]

2.2 Hydraulics

The main lift is driven by two cylinders whereas the free lift is driven by a single cylinder. In order to lift the forks a pump and two valves are used, one valve for the free lift cylinder and one valve for the two main lift cylinders. The connections of the cylinders and the valves can be seen in Figure 2.3 which gives an overview of the hydraulic system for the lowering function. When lowering the forks no pump-work is needed, instead gravity is made use of.

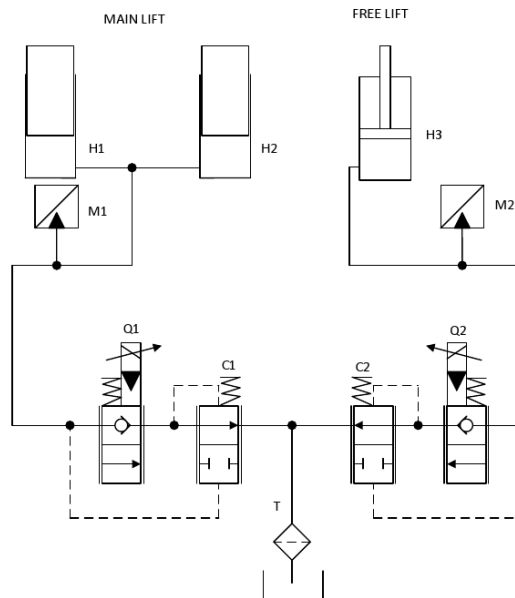


Figure 2.3: An overview of the hydraulics system for the lowering function. H1, H2 are the cylinders for the main lift and H3 is the cylinder for the free lift. M1 and M2 are the pressure sensors. Q1 is the valve for the main lift and C1 is the associated pressure compensator. Q2 and C2 are the corresponding for the free lift [5].

The velocity when lowering is determined by the flow through two proportional valves, one for the main lift and one for the free lift. There are also pressure compensators to get a constant pressure drop over the valves. For safety reasons there is a maximum velocity of 0.6 m/s allowed when lowering the forks. On this particular forklift the actual maximum velocity is 0.53 m/s for the main lift when the oil is cold and without load on the forks. The corresponding velocity for the free lift is 0.39 m/s. The reason for both the differences – between maximum velocity and the actual velocity and between velocity of the main and free lift – is hardware limitations.

2.3 Electrical Subsystem

The forklift has two electronic control units, one main control unit (MCU) and one secondary control unit (SCU). As the name reveals the MCU primarily controls the system and the SCU supports the MCU when needed. These two units communicate with each other by using a controller area network (CAN), which is a serial bus system.

The valves are controlled by a current that is calculated based on the joystick signal and in what height region the forks are. The joystick is sent over the CAN to the MCU where a corresponding current is calculated and then sent over the CAN to the SCU. The SCU determines a pulse width modulation (PWM) signal from the current and this signal is sent to the valves.

2.3.1 Sensors

The forklift has many different sensors and the ones relevant when working with the fork lowering are the sensors that measure pressure, height and temperature. There are two pressure sensors which are measuring the hydraulic oil pressure in the cylinders for the free lift and main lift, respectively. The temperature sensor measures the temperature of the hydraulic oil in the oil tank.

For the height measurement there are two encoders, measuring x_{FL} and x_{ML} in Figure 2.2. The signal x_{FL} is the total height of the free lift while x_{ML} is the height of the main lift cylinders. Due to the separation between the stages only half the distance of what the forks are actually moving is measured. The total height is therefore given by

$$x_G = x_{FL} + 2x_{ML}. \quad (2.1)$$

Since x_{ML} is doubled, the height for the main lift has lower accuracy than the height for the free lift since the sensors have the same resolution.

In addition to the height encoders there are three height reference signals. These reference signals are binary signals which switch at specific heights. One of these sensors is placed in the free lift region and two in the main lift region. These references make sure that the fault in the height encoders does not grow indefinitely since the height measurement can be reset to the reference height when passing a height reference.

2.3.2 Control System

The forks are currently controlled by an open loop system calculating a current based on the joystick. To avoid any hard starts or stops that create unnecessary jerks, the current consists of ramps to smoothly reach the desired velocity. In this way the load on the forks and other interferences are not taken into consideration.

Figure 2.4 is an overview of the control system. The dashed lines from the system indicate the height of the forks. The height is used in the state machine to decide whether the forks are in the region for main lift, transfer or free lift. This state

information and the height is then used to decide how to calculate the current. This makes sure that the current to the valves is ramped down when getting close to the mechanical stop at the ground, the top or when in the transfer region. Therefore the height and state do not make a feedback system, but are only used to decide which subsystem to use. This is indicated with the dashed lines in Figure 2.4.

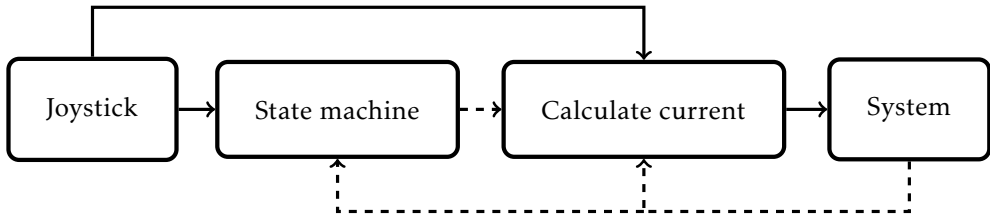


Figure 2.4: Overview of the control system. The dashed lines indicate that the information is used only to choose states or limitations of the current.

To make sure that the valves open completely when there is a maximum joystick input, a current of 200 mA is added when the maximum 1200 mA is reached. In a similar way, an extra current is sent to the main valve when using the free lift to make sure that the main lift reaches the bottom.

2.4 Speedgoat Real-Time Target Machine

The hardware used for RCP, the Speedgoat real-time target machine, is connected to the signals needed for this thesis. To be able to switch easily between using the Speedgoat device for control and the MCU all signals are connected in parallel, as can be seen in Figure 2.5. The differences between Figure 2.5 and the original, Figure 2.1, are marked in red and all new arrows are also dashed. The involved signals are the pressure sensors, the height encoders, the height references, the temperature sensor, joystick values and CAN communication with the MCU and SCU.

The real-time target machine will be used to calculate the current to the valves controlling the lowering instead of using the MCU. All other functions will be controlled by the MCU. All currents, from both the real-time target machine and the MCU, will still be sent to the SCU over CAN in the same way as before.

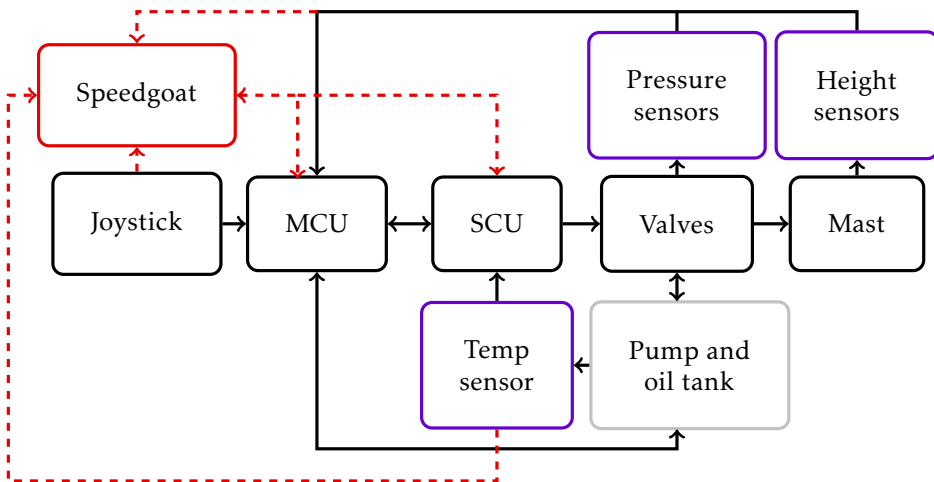


Figure 2.5: Overview of the system with the Speedgoat including electrical, hydraulic and mechanical parts.

3

System Identification

A simplified physical Simulink model for the load handling of the RRE forklift was provided by TMH at the start of the thesis. This model is validated using cross-validation between measurements from the real forklift and outputs from the simulation model. To improve the model accuracy some parameters were estimated. For comparison, a nonlinear black-box model is also constructed. Both models are later used for simulation of the controllers and compared in Section 5.4. The models are discussed and evaluated from an RCP point of view in Chapter 6.

Further on, to be able to use a linear MPC, a linear model of the system in state-space (SS) form is needed. This can be done with either a black-box model estimated from measurements of the real system or by constructing one by hand from known information about the system. The validation and the comparison of the different models are also presented in this chapter.

3.1 Data Collection

Measurement data are needed for estimation of both the parameters in the physical model and the black-box models as well as for model validation. To capture as much of the real systems behaviour as possible, different loads and joystick signals were used. In some of the measurements the joystick was even released a few times, in both the main lift and the free lift, to capture the behaviour of the system when the forks stop in mid-air.

Nine sets of data were collected. The characteristics of each data set are presented in Table 3.1. To make the scenarios more realistic the joystick was controlled by a driver. The measurements with 1700 kg are not from the top of the mast since the mast only manages 1100 kg at the top height. The joystick signals and generated

Table 3.1: Table of the measurements collected for the estimation and validation procedures.

Name	Joystick	Starting height [m]	Stops in main lift	Stops in free lift	Load [kg]
<i>Meas1</i>	100 %	10.9	0	0	0
<i>Meas2</i>	100 %	10.9	2	1	0
<i>Meas3</i>	ca 50 %	10.9	2	1	0
<i>Meas4</i>	Slow ramp from 0-100 %	10.9	0	0	0
<i>Meas5</i>	100 %	10.9	2	1	1000
<i>Meas6</i>	ca 50 %	10.9	2	1	1000
<i>Meas7</i>	Slow ramp from 0-100 %	10.9	0	1	1000
<i>Meas8</i>	100 %	6	1	1	1700
<i>Meas9</i>	ca 50 %	6	1	1	1700

currents can be seen in Appendix A.

3.2 Simulation Models

Two different models for the purpose of simulating the controllers are constructed, a grey-box model mainly based on physical equations and a black-box model estimated from data from the real system.

3.2.1 Physical Model

One model used for simulating the control systems is a grey-box model based on physical equations built in Simulink and Simscape. Due to unknown parameters and parameters that are hard to measure the model contains look-up tables for the characteristics of the valves. To make the model more accurate, parameter estimation is used and the model is later validated against measurements from the actual forklift.

When comparing values from simulations with measurements from the real system it was clear that the flow in the main lift was too high and the flow in the free lift too low. This appears in how fast the height decreases compared to the measurement in Figure 3.1a. As also seen in the figure, the height of the free lift never reaches bottom in the simulation. The reason for this is that the simulation uses the same input signals (currents) as the data in *Meas1* in Table 3.1. The input signals stop when the forks in *Meas1* reach the bottom and since the simulation is slower the forks do not reach the bottom. From Figure 3.1b it can be seen that the pressure for the main lift has the appropriate shape with constant pressure when lowering and a pressure drop when reaching the bottom. The free lift on the other hand only matches when lowering the free lift. The pressure is higher

in the measurements when the free lift is at the top because then it is pressed against the mechanical stop, a phenomenon not captured in the model. After the lowering has finished the pressure is higher in the model than on the real system because in the model the free lift is not at the bottom while it is in the measurement.

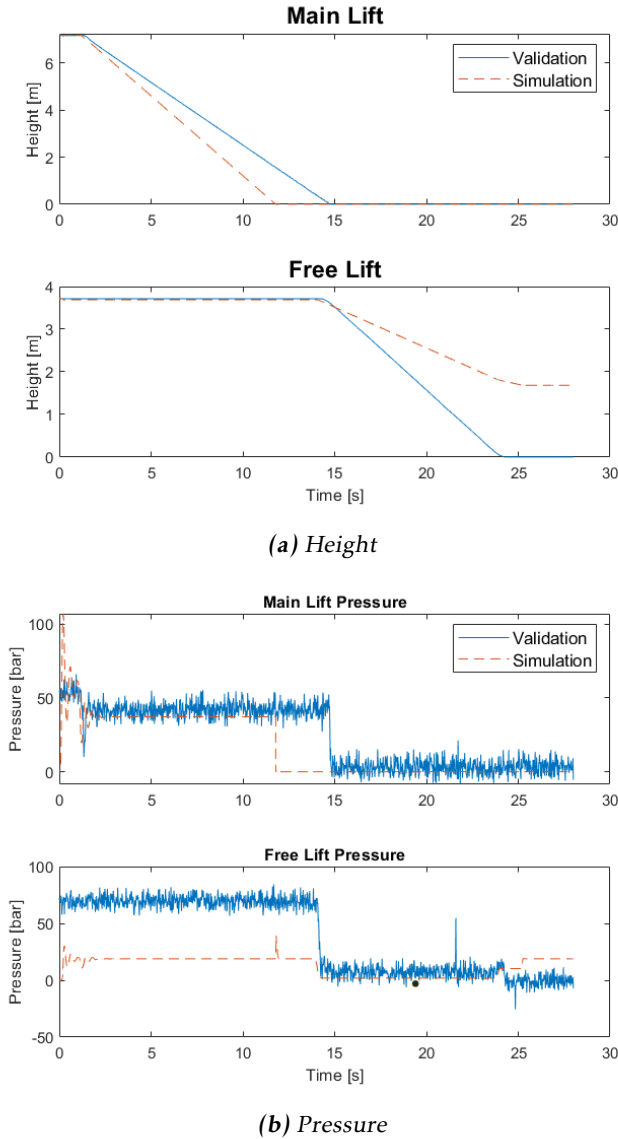


Figure 3.1: Comparison of physical model and measurement data before parameter estimation.

To improve the above described problems, parameter estimation was used. The model contains lots of parameters that have uncertain values. Three parameters that do have uncertain values are the currents needed for each valve to open, called the deadbands, and the friction in the mast. These parameters have a large effect on the speed during lowering and therefore it seems reasonable to estimate these to get a more accurate simulation model.

The desired values for the deadbands and the friction are the values that make the simulation model behave as similar to the real system as possible. To do this a cost function is formulated and the best parameter values are the ones that minimize this function. The cost function used is a quadratic (least-square) function where the difference between the simulated model and estimation data is squared and added together for each sampled time. It is defined as the following

$$J(\beta) = \sum_{i=0}^N (y_i - f(x_i, \beta))^T (y_i - f(x_i, \beta)) \quad (3.1)$$

where N is the number of samples, y_i are the measured values of height and pressure, for both free and main lift, at sample i and $f()$ is the output of the model at sample i with the adjustable parameters β which are the deadbands and the friction.

The optimization algorithm used in the parameter estimation is Trust-Region Reflective with Nonlinear Least-Square as method. Trust-Region Reflective is described by MathWorks in [16].

The data used for the estimation was *Meas1*, *Meas4*, *Meas5* and *Meas6* presented in Table 3.1 in Section 3.1. These measurements were chosen to get different kinds of joystick signals and loads on the forks. No measurement with a load of 1700 kg was chosen to be able to prove that the estimated parameters works on other loads than they are estimated for.

In the parameter estimation, different initial guesses of the deadbands and the friction were used to see if there existed several minima. However, independently of the starting values of the parameters to be estimated the end result was the same.

Different loads affect the result of the simulation the most and therefore *Meas2* and *Meas8* from Table 3.1 were used to validate the model with the estimated parameters. Figure 3.2 and 3.3 show the result of the simulation with both the original parameters and the estimated parameters and the measured data for *Meas2* and *Meas8* respectively. The pressure for both the free and the main lift are improved with the estimated parameters. Besides the load on the forks, the pressure in the model also strongly depends on the speed of the forks. When the speed becomes more accurate so does the pressure. The pressure and more validations can be seen in Appendix B.

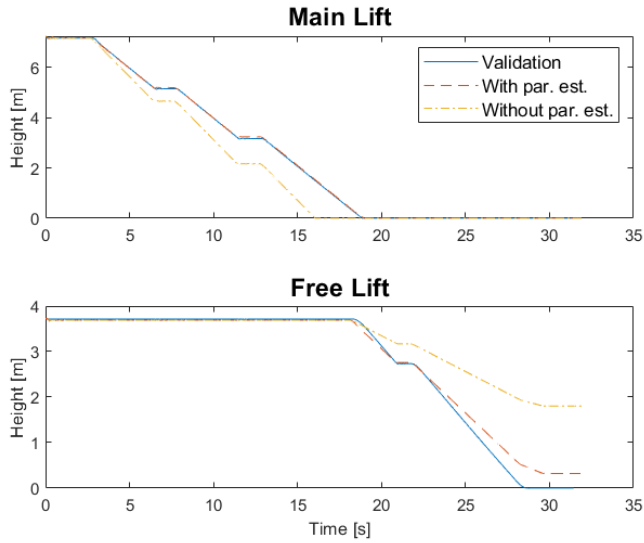


Figure 3.2: Model validation using *Meas2*. The solid blue line shows the measured data, the red dashed line shows the model simulated with the estimated parameters and the dashed-dotted yellow line shows the model simulated with the original parameters.

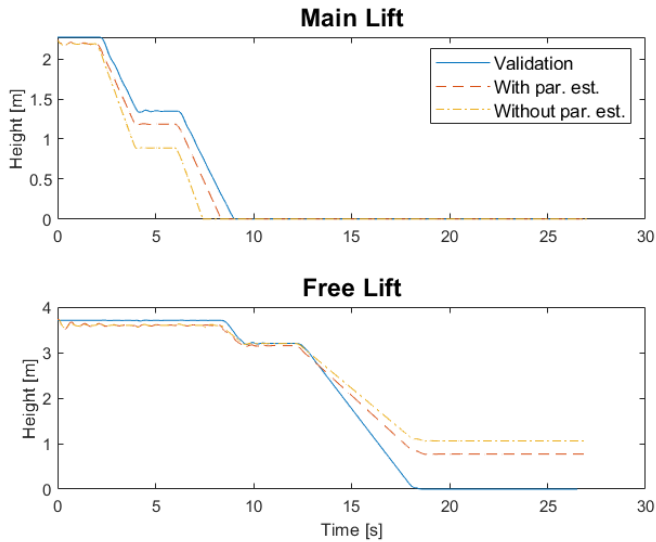


Figure 3.3: Model validation using *Meas8*. The solid blue line shows the measured data, the red dashed line shows the model simulated with the estimated parameters and the dashed-dotted yellow line shows the model simulated with the original parameters.

3.2.2 Nonlinear ARX Model

Another simulation model that is constructed is a black-box model on the form nonlinear ARX, described by Mathworks in [17]. A linear ARX model has the form

$$y(t) = \frac{b_1 q^{-n_k} + b_2 q^{-n_k-1} + \dots + b_{n_b} q^{-n_k-n_b+1}}{1 + a_1 q^{-1} + \dots + a_{n_a} q^{-n_a}} u(t) + \frac{1}{1 + a_1 q^{-1} + \dots + a_{n_a} q^{-n_a}} e(t) \quad (3.2)$$

which means that the output ($y(t)$) is given by the weighted sum of the delayed input ($u(t)$) and output variables and the noise ($e(t)$). A nonlinear ARX model has a more free dependency on the regressors and can simply be written as

$$y(t) = F(q^{-1}y(t), q^{-2}y(t), \dots, q^{-n_a}y(t), q^{-n_k}u(t), q^{-n_k-1}u(t), \dots, q^{-n_k-n_b+1}u(t)) + e(t). \quad (3.3)$$

When estimating a multivariable ARX model the parameters have to be chosen for all inputs and outputs resulting in n_a , n_b and n_k becoming 2x2-matrices. To avoid too many design parameters the matrices are chosen as diagonal matrices times a constant k_i as the following

$$n_i = k_i \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \quad (3.4)$$

The nonlinear ARX model is estimated from two different data sets, *Meas1* and *Meas6* in Table 3.1. These two were chosen so that the mapping from input to output would be as complete as possible. To get more linear data, the current deadband and the extra maximum current, see Section 2.3.2, were cut off. These can be added later on outside the model when using it.

The orders for the model are chosen by comparing different model orders by Akaike Information Criterion (AIC) and model fit to validation data. The result can be seen in Table 3.2; different model orders do not make a big difference in either AIC or the fit to validation data. A model of low order is preferred and therefore $k_a = 1$, $k_b = 1$ and $k_k = 1$ are chosen.

The simulation of the ARX model with the chosen order against validation data from *Meas2* can be seen in Figure 3.4. The model preserves the form of the line but is slower than the actual height.

Table 3.2: Table with model fits for the different nonlinear ARX models. The first value corresponds to the fit of the main lift height and the second value to the free lift height.

Model order ($k_a/k_b/k_k$)	AIC	Fit Meas2 [%]	Fit Meas4 [%]	Fit Meas6 [%]	Fit Meas8 [%]
1/1/1	11.19	94.83/96.45	83.88/95.56	89.27/78.73	64.88/57.27
1/1/2	11.18	95.48/95.88	83.16/94.80	87.45/81.06	55.71/60.56
1/2/1	11.18	95.48/95.87	83.20/94.74	87.44/81.29	55.95/60.72
1/2/2	11.17	95.42/96.04	83.64/95.06	88.20/80.57	61.89/59.97
2/1/1	10.90	95.31/96.18	83.40/95.09	87.99/80.62	58.95/60.00
2/1/2	10.89	94.89/96.41	84.07/95.52	89.25/79.14	67.27/58.22
2/2/1	10.89	95.70/94.85	81.74/92.86	84.52/84.43	38.15/65.32
2/2/2	10.89	95.34/96.15	83.90/95.11	88.56/80.42	65.12/60.12

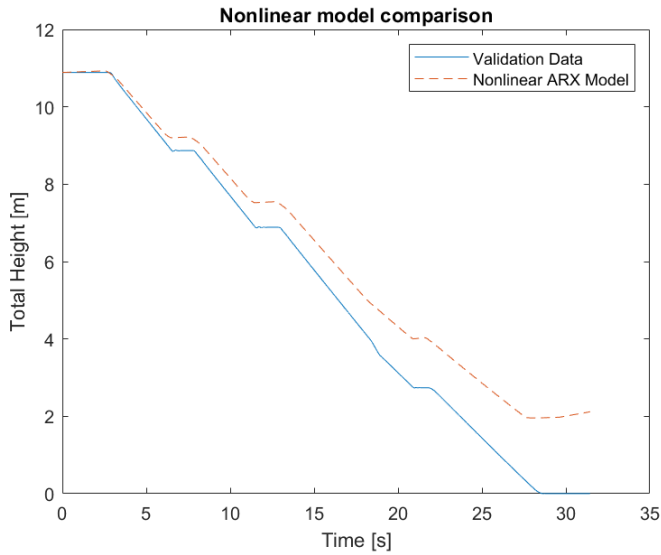


Figure 3.4: Comparison of nonlinear ARX model and validation data Meas2.

3.3 Internal Model

In one of the controllers, the MPC, an internal model is needed. This model has to be linear and in state-space (SS) form. It is estimated as a black-box model.

A linear SS model has the form

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx\end{aligned}\tag{3.5}$$

where x represents the states, u represents the control signal, y represents the measured signals and A , B and C are the matrices that contain the system dynamics. The data sets used when estimating are the same as the ones used when estimating the nonlinear ARX-model, which are *Meas1* and *Meas6* in Table 3.1.

As a comparison, a handmade linear model was created. The handmade model has two states, representing the height of the main lift and the height of the free lift respectively, which means that the C matrix is an identity matrix. The derivatives of the states, the velocities, will then be proportional to the current or input u which leads to the A matrix in (3.5) consisting of zeros. The matrices look as follow

$$A = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, B = \begin{pmatrix} b_1 & 0 \\ 0 & b_2 \end{pmatrix}, C = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.\tag{3.6}$$

Choosing the model order, the number of states (x), for the black-box model is done by comparing models of different orders. To compare the models two methods are used, AIC and the fit to measurement data. The results for the different models can be seen in Table 3.3. In this table it can be seen that the handmade model performs fairly well but not as well as the estimated model of order two.

According to the table, an order of two is one good choice since it has a low AIC and a decent fit. This choice also keeps the model order low making it easier to work with when later using it in the MPC. Figure 3.5 shows the hand-made model against the linear model of order two for the validation data from *Meas2*.

Table 3.3: Table with model fits for the different linear models. The first value corresponds to the fit of the main lift height and the second value the fit to the free lift height. The hand-made model is not estimated and therefore no AIC value can be calculated.

Model order	AIC	Fit Meas2 [%]	Fit Meas4 [%]	Fit Meas6 [%]	Fit Meas8 [%]
<i>Hand-made</i>	-	70.03/54.25	77.66/58.48	71.61/82.75	16.7/89.6
1	23.52	3.184/60.31	4.51/62.26	1.667/50.21	-42.18/63.81
2	11.01	95.5/95.76	89.25/95.73	87.48/78.09	89.92/76.22
3	10.45	94.1/93.98	90.07/96.63	86.8/83.71	76.71/80.83
4	15.03	-622.9/-1199	-565.4/-1071	-335.7/-837.5	-904.3/-154.9
5	10.43	94.18/94.29	90.12/96.92	86.86/83.43	78.38/80.51
6	14.00	-503.6/-809	-474.5/-744.9	-183.4/-348.6	-981.6/-189.5
7	10.39	94.25/94.07	89.99/96.72	86.98/83.44	78.01/80.6
8	10.38	95.7/93.72	91.69/93.84	88.9/78.68	83.76/78.94

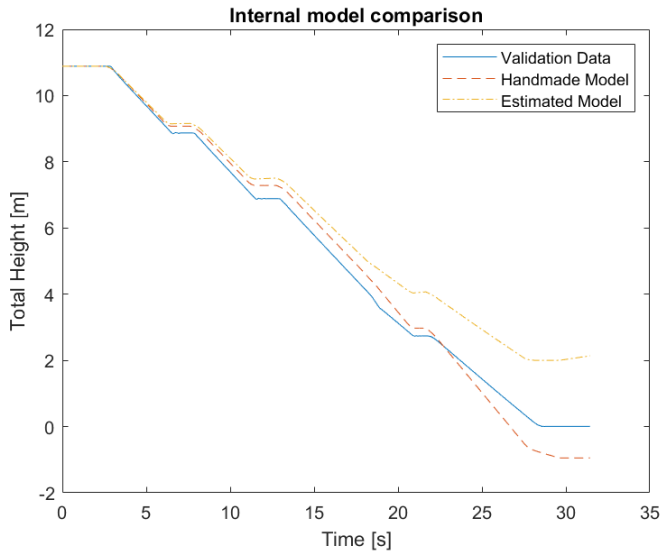


Figure 3.5: Comparison of linear models and validation data Meas2.

3.4 Comparison of Models

All models are compared by simulation with the same input signals as a measured data set. Figure 3.6 shows the models against the validation data in *Meas2* from Table 3.1 and Figure 3.7 for *Meas8*. Important in the figures is that the physical model always preserves the true shape of the height but can have a more or less apparent static error, especially when having a high weight on the forks as in Figure 3.7. The black-box models, nonlinear ARX model and linear SS model, on the other hand do not always preserve the correct form of the height curve. This can especially be seen when the true height reaches the bottom in Figure 3.7 where the two black-box models start rising again.

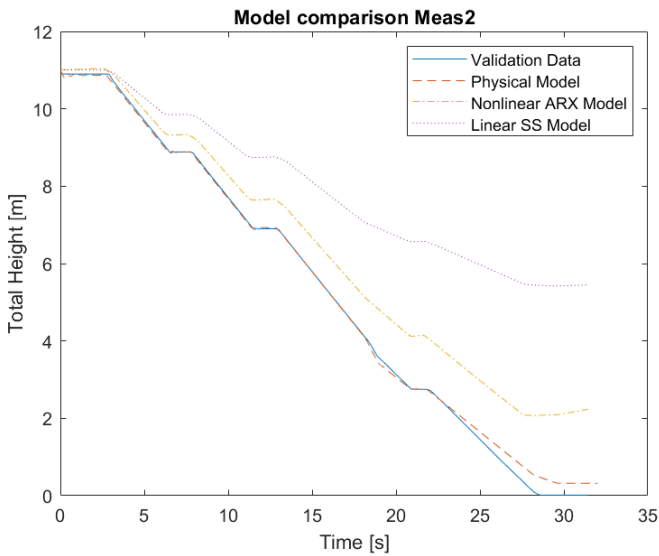


Figure 3.6: Comparison of all the models on validation data *Meas2*.

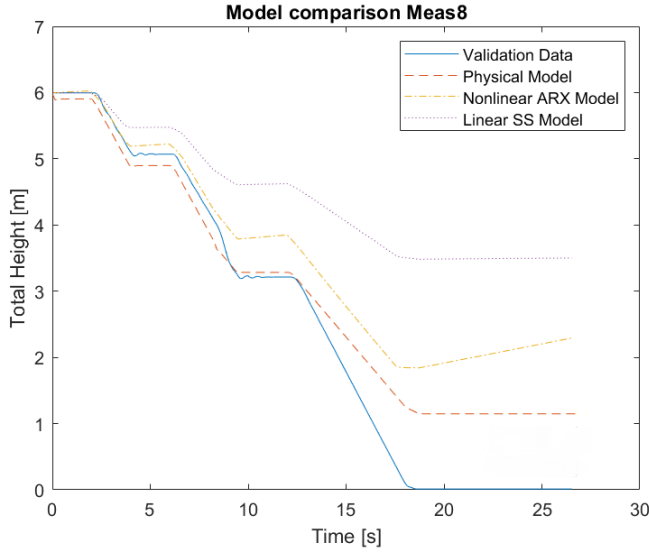


Figure 3.7: Comparison of all the models on validation data *Meas8*.

3.5 Discussion

The provided physical grey-box model is validated and improved with parameter estimation and compared to the nonlinear ARX black-box model. Furthermore, two linear models used for MPC are constructed, one black-box model and one handmade model. The results are discussed in this section.

3.5.1 Simulation Models

The simulation models have the purpose to be used in simulations before trying the control system on the actual plant. These models have to be accurate enough in order for the control designer to judge if the control system is safe to run on the actual plant. To be able to judge this, the simulation model has to be run with the new control system and then finally be compared to the real plant's behaviour with the same control system.

In both Figure 3.2 and Figure 3.3 it is clear that the physical model is more accurate with the estimated parameters than the original parameters. It can be seen that the parameters are better for *Meas2*, which has load 0 kg, than for *Meas8*, which has load 1700 kg. This can be explained by the fact that the model is estimated using only data that has load 0 kg or 1000 kg. The improvement for *Meas8* indicates that the parameters work for 1700 kg and therefore the model should be valid for other loads as well.

The black-box model created in Section 3.2.2 capture the most obvious dynamics of the system but is slower than the real system, as can be seen in Figure 3.4.

3.5.2 Internal Models

Deciding whether the linear models are good enough depends on the purpose. For this thesis, the purpose of a linear model is to use it for predictive control. That means that this model has to provide a reliable prediction only for the near future; feedback can compensate for some model errors. Even though the black-box model performs better than the handmade model, as seen in Table 3.3, there are some advantages of using the handmade model. The handmade model's states have a physical representation which makes it easier to interpret. There is also no need to estimate the states when using it for control since the states in the handmade model are measured. Both models capture the basic dynamics of the system as can be seen in Figure 3.5.

4

Control Design

To evaluate RCP and MBD two different controllers are designed, a P controller with gain scheduling and a Model Predictive Controller (MPC). Both controllers need a reference to follow and in this chapter two different reference trajectories are proposed, one simpler and one more advanced.

4.1 Reference Trajectory

The controller needs a reference to follow and this reference can be designed in different, more or less advanced, ways. Two different reference signals are constructed, one simple and one more advanced. From this it is possible to see if the performance can be improved with a better reference.

The input signal, the joystick, indicates how fast the driver wants the forks to move. The measured feedback signals on the other hand are the height of the free and main lift respectively. Therefore there are mainly two ways to construct the reference signal, either differentiate the measured height to a velocity or integrate the desired velocity to a height. Since differentiating a measured signal often leads to noisy signals the second alternative, integrating the velocity, was chosen. The reference signal ($x_{ref}(t)$) is therefore a trajectory for the forks.

It is not possible for the forks to stop in the instance the joystick is released without risking to damage either forklift or load. For the present controller it takes 0.47 seconds to stop the forks when the joystick is released from 100 %.

4.1.1 Integrated Velocity

The relation between the size of the joystick signal (j) and the velocity (v_{ref}) of the forks is chosen as linear according to

$$v_{ref}(t) = \left| \frac{v_{max}}{j_{max}} \right| j(t), \quad (4.1)$$

where j_{max} is the maximum joystick signal and v_{max} is the maximum velocity. The height is then calculated from

$$x_{ref}(t) = \begin{cases} x_0 - \int_{t_0}^t v_{ref}(\tau) d\tau, & 0 < t < T + k_t \\ x_E, & t \geq T + k_t \end{cases} \quad (4.2)$$

where x_0 is the fork height when the driver initiate the movement, x_E the height where the forks should stop at and T is the time when the joystick signal becomes zero. The allowed time for the forks to stop in the reference is k_t . The size of k_t is larger for larger joystick values because it takes longer for the forks to stop with a high velocity. When this time has passed the trajectory is reset to be equal to the current fork height until a new movement is initiated.

4.1.2 Minimal Jerk

To avoid any unnecessary jerk, that in worst case could make the load fall off, a minimal jerk trajectory is calculated. The minimal jerk trajectory, $x_{ref}(t)$, is the solution, $x(t)$, to minimizing the jerk over a time period as in

$$\text{minimize } \frac{1}{2} \int_0^T \left(\frac{d^3 x(t)}{dt^3} \right)^2 dt \quad (4.3)$$

where $x_0, \dot{x}_0, \ddot{x}_0, x_T, \dot{x}_T$ and \ddot{x}_T are known.

Here x is the total height of the forks. To solve problem (4.3) a function of the following form is designed

$$h(\epsilon, t) = x(t) + \epsilon g(t). \quad (4.4)$$

Here g is an arbitrary function that has continuous second partial derivatives with initial and final conditions satisfying

$$\begin{aligned} g_0 &= g_T = 0 \\ \dot{g}_0 &= \dot{g}_T = 0 \\ \ddot{g}_0 &= \ddot{g}_T = 0. \end{aligned} \quad (4.5)$$

One way of minimizing (4.3) is then to define (where $h^{[3]}$ is the third time derivative of h)

$$F(\epsilon) = \frac{1}{2} \int_0^T (h^{[3]})^2 dt \quad (4.6)$$

with

$$\left. \frac{dF(\epsilon)}{d\epsilon} \right|_{\epsilon=0} = 0. \quad (4.7)$$

The condition (4.7) can be expanded:

$$\left. \frac{dF(\epsilon)}{d\epsilon} \right|_{\epsilon=0} = \int_0^T x^{[3]}(t)g^{[3]}(t) dt = 0. \quad (4.8)$$

By repetitively using integration by parts and the conditions for δ in (4.5) it can be seen that

$$\int_0^T x^{[3]}(t)g^{[3]}(t) dt = - \int_0^T x^{[6]}(t)g(t) dt \quad (4.9)$$

must hold. Hence, to satisfy (4.7), the right side of (4.9) has to be equal to zero. Since this has to be the case for the arbitrary function $g(t)$ it can be concluded that $x^{[6]}(t) = 0$ in the region $[0, T]$. This means that a solution in the form of a fifth order polynomial will solve (4.3):

$$x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5. \quad (4.10)$$

What is left to calculate are the constants a_0, a_1, a_2, a_3, a_4 and a_5 . The first three are simply solved by using the initial conditions in (4.3) resulting in

$$x(0) = a_0 = x_0 \quad (4.11)$$

$$\dot{x}(0) = a_1 = \dot{x}_0 \quad (4.12)$$

$$\ddot{x}(0) = a_2 = \frac{1}{2}\ddot{x}_0. \quad (4.13)$$

The last three are solved by using the terminal conditions in (4.3) which written in matrix form are

$$\begin{bmatrix} x_T - a_0 - a_1 T + a_2 T^2 \\ \dot{x}_T - a_1 - 2a_2 T \\ \ddot{x}_T - 2a_2 \end{bmatrix} = \begin{bmatrix} T^3 & T^4 & T^5 \\ 3T^2 & 4T^3 & 5T^4 \\ 6T & 12T^2 & 20T^3 \end{bmatrix} \begin{bmatrix} a_3 \\ a_4 \\ a_5 \end{bmatrix}. \quad (4.14)$$

The equations (4.11), (4.12), (4.13) and (4.14) are solved to calculate the trajectory (4.10) every time the velocity is changed by the joystick. The initial conditions x_0, \dot{x}_0 and \ddot{x}_0 are the height, velocity and acceleration in the last trajectory point. The final time T is determined from the change in velocity and the maximum allowed time to stop according to

$$T = T_{max} \frac{|v_{current}(t) - v_{ref}(t)|}{v_{max}} \quad (4.15)$$

where v_{ref} is calculated as in (4.1). The final height is then calculated from

$$x_T = \max(x_{old} - T \frac{|v_{current}(t) - v_{ref}(t)|}{2}, 0). \quad (4.16)$$

The final velocity (\dot{x}_T) is set to the reference velocity (v_{ref}) and the acceleration is the derivative of v_{ref} . To check if the forks are close to the bottom v_{ref} is set to zero in both (4.15) and (4.16). If that results in x_T becoming equal to zero, a new trajectory is calculated with the bottom as final height and with zero velocity and acceleration.

When the velocity is constant and the forks are not close to the bottom, the trajectory is calculated as

$$x_{ref}(t + T_s) = x_{old} - T_s v_{ref}(t) \quad (4.17)$$

where the sample time (T_s) is 0.02s.

4.2 P Controller

One simpler controller that is implemented is a P controller. The controller is designed with gain scheduling to be able to better handle the nonlinearities in the system. A block scheme of the P controller when only using one of the valves can be seen in Figure 4.1. How the controller works when using both valves is described in Section 4.2.3.

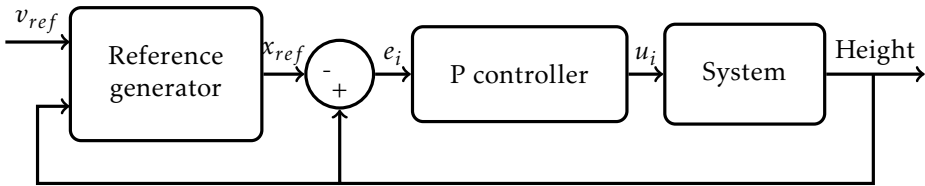


Figure 4.1: Block scheme of the single variable P controller. The variable i is the involved valve, ML in the main lift and FL in the free lift.

4.2.1 Control Error

When the forks are lowered with the main valve the control error ($e_{ML}(t)$) is calculated from the following equation

$$e_{ML}(t) = 2x_{ML}(t) - \max(0, x_{ref}(t) - x_{FL}(t)) \quad (4.18)$$

where $x_{ref}(t) - x_{FL}(t)$ is the height reference for only the main lift, which has to be positive and hence the max function is needed. As described in Section 2.3.1 the measured height x_{ML} is half the distance the main lift is actually moving and therefore the control error is two times x_{ML} subtracted with the reference.

The control error for the free lift is calculated in a similar way to the main lift

but simplified since the total height is equal to the free lift height after leaving the transfer region. This means that the system in this region is a single variable control system. The control error for the free lift is given by

$$e_{FL}(t) = x_{FL}(t) - x_{ref}(t). \quad (4.19)$$

4.2.2 Gain Scheduling

A linear P controller does a poor job controlling the nonlinear valves. A low P value makes the controller slow at ramping up and down the current, resulting in slow acceleration of the forks, while a high P value creates larger and longer lasting oscillations when stopped. One way to avoid having to choose between these trade-offs is by using gain scheduling. The basics of the gain scheduling principle are described in [4].

Gain scheduling is implemented, for both the main lift and the free lift, using a look-up table to map different P-values to different regions. The regions are decided by the height and the joystick signal. The structure can be seen in Figure 4.2. The height is needed when the forks are reaching the bottom of the main lift or the free lift. These two cases are dealt with in a specific way. At the bottom of the main lift the gain scheduling is constant while the gain scheduling for the free lift has a higher value when the height decreases. Otherwise the gain schedulings are based on the velocity and are increased with a higher velocity.

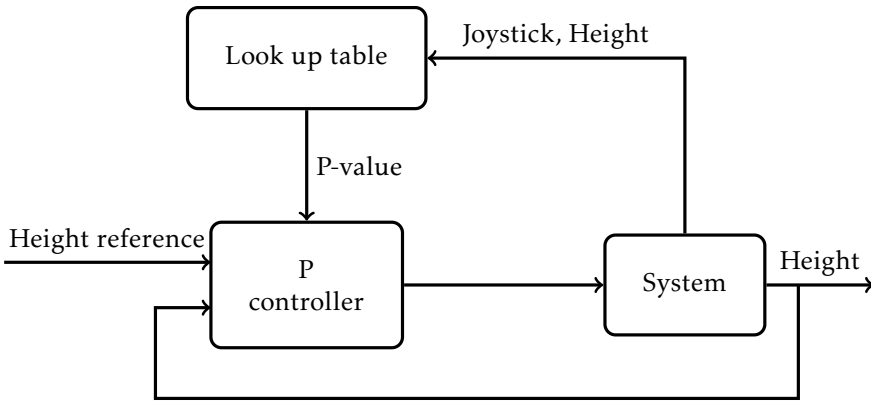


Figure 4.2: The structure of the P controller with gain scheduling.

4.2.3 Transfer Region

In the transfer region, where the control of the lowering function switches from the valve for the main lift to the free lift, the goal is to ramp down the main lift as well as ramp up the free lift to maintain a constant total velocity. The height reference is calculated for the total height of the forks. Therefore when the main lift approaches the transfer region the height reference for the main lift approaches zero. This leads to a difference between the reference velocity, from

the joystick, and the velocity obtained with only the main lift. To compensate for this difference, a current for the free lift valve is calculated based on the difference in velocity and fed forward to the free lift valve in a process which can be seen in Figure 4.3. The current for the main lift is still calculated as in Figure 4.1.

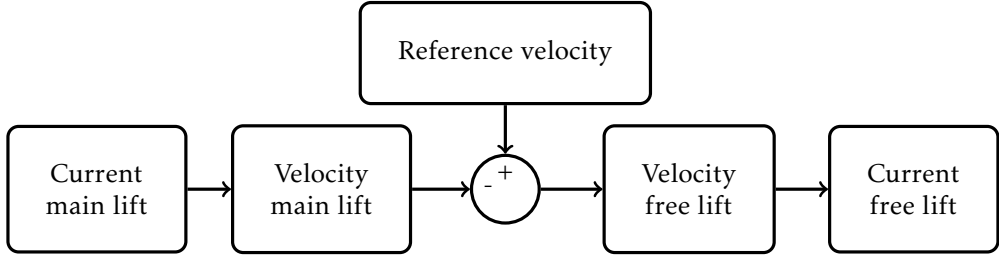


Figure 4.3: The process for calculating the free lift current in the transfer region.

The P controller for the main lift has a current as output. By using a look-up table, based on measurements on the actual valve, the current is mapped to a flow. This flow (q) and the velocity (v) of the forks have the relation according to

$$v(t) = \frac{v_{max}}{q_{max}} q(t) \quad (4.20)$$

where v_{max} is 0.57 m/s and q_{max} has different values for the main and free lift, 54.4 l/min and 48.3 l/min respectively. The reason for v_{max} being less than 0.6 m/s is to make sure that the velocity never exceeds 0.6 m/s no matter what load and what temperature the oil in the hydraulic system has reached.

The needed velocity from the free lift (v_{FL}) is calculated as the difference

$$v_{FL}(t) = v_{ref}(t) - v_{ML}(t) \quad (4.21)$$

between the reference velocity from the joystick (v_{ref}) and the generated main lift velocity (v_{ML}). The flow for the free lift value is calculated from (4.20) using the velocity from (4.21). The corresponding current is obtained from a look-up table for the valve.

4.2.4 Implementation

The control error, for either the main lift (4.18) or the free lift (4.19) depending on where the forks are, is scaled via a P controller and saturated to make sure that the valves do not get a negative current or that the current is higher than the maximum current 1200 mA.

The valves have an opening current, which means that it takes a certain current for the valves to start opening. This opening current, also called the deadbands, were estimated in Section 3.2.1 to be 465 mA for the main valve and 265 mA for the free valve. To compensate for this nonlinearity the corresponding current is always added directly to the output from the P controller. A similar nonlinearity

appears when the opening area is close to the maximum opening area, so to make sure that the valves open completely when there is a maximal joystick input, a current of 200 mA is added when the current reaches 1200 mA.

4.3 MPC

A more advanced control strategy that has been implemented is an MPC (Model Predictive Control) that uses an internal model to predict how a control signal is going to affect the system over time. The MPC uses the references, x_{ref} , presented in Section 4.1. To avoid confusion with the state variable x here they will be indicated with r . A block scheme of the controller can be seen in Figure 4.4.

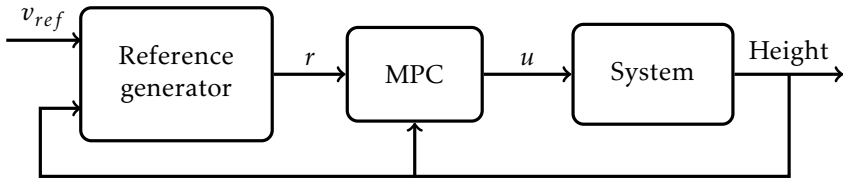


Figure 4.4: Block scheme of the MPC.

4.3.1 Goal Function

The MPC chooses a control signal that aims to minimize a goal function over a certain amount of time. The goal function for this application can be divided into three parts as the following

$$\begin{aligned}
 J_N(x(k)) = & \sum_{j=0}^{N-1} \|z_1'''(k+j) + z_2'''(k+j)\|_{Q_1}^2 + \\
 & \|z_1(k+j) + z_2(k+j) - r(k+j)\|_{Q_2}^2 + \\
 & \|u(k+j) - u(k+j-1)\|_{Q_3}^2
 \end{aligned} \tag{4.22}$$

where the first part, $z'''(k+j)$, punishes the jerk of the forks. It has the same function as the reference signal in Section 4.1.2 and this term makes it possible to keep the reference simple while still not getting unnecessary jerk. The discrete third derivative of $z(k)$ is derived using *Euler forward* which is

$$z'(k) = \frac{1}{T_s} (z(k+1) - z(k)). \tag{4.23}$$

This can be described as a function of the states, $x(k)$, by using

$$z(k) = Mx(k) \tag{4.24}$$

so that

$$\begin{aligned}
 z'(k) &= \frac{1}{T_s} M(x(k+1) - x(k)) = \\
 &= \frac{1}{T_s} M(Fx(k) + Gu(k) - x(k)) = \\
 &= \frac{1}{T_s} M((F - I)x(k) + Gu(k)).
 \end{aligned} \tag{4.25}$$

Repeating this gives the third derivative

$$z'''(k) = \frac{1}{T_s^3} M((F - I)^3 x(k) + (F^2 - 3F + 3I)Gu(k) + (F - 3I)Gu(k+1) + Gu(k+2)). \tag{4.26}$$

The second part of (4.22), $z_1(k+j) + z_2(k+j) - r(k+j)$, punishes deviation from the reference. The $z_1 + z_2$ term represents the total fork height and r is the reference for the total fork height. The reference is decided from the joystick signal, according to Section 4.1, and therefore this part makes sure that the forks will follow the joystick signal.

The last part of (4.22), $u(k+j) - u(k+j-1)$, punishes a difference in the control signals. This makes sure that no unnecessarily fast changes are made to the control signal.

4.3.2 Conditions

One of the big advantages of using MPC is that it is easy to include constraints. In this thesis there are four different conditions:

$$\begin{aligned}
 0 &\leq u_i(k) \leq 1400, & i &= 1, 2 \\
 u_2(k) &< 1400 \left\lfloor \frac{0.2}{z_1(k)} \right\rfloor \\
 z_i(k) &\geq 0, & i &= 1, 2 \\
 z'_i(k) &< 0.6, & i &= 1, 2.
 \end{aligned} \tag{4.27}$$

The first one limits the control signals to their limits in mA. The second condition makes sure that the free lift can not start lowering until the main lift is less than 0.2 m. The third condition is telling the MPC that the heights of the main and free lift never can be less than zero. The forth, and last, condition limits the speed of the forks to be less than 0.6 m/s which is the legal limit.

4.3.3 Design

The MPC has four design parameters, three matrix-valued weight parameters in the goal function and one prediction horizon. The three weight parameters, Q_1 , Q_2 , and Q_3 , decide how much each term in the goal function (4.22) should be penalized. The prediction horizon, N , decides how far into the future the MPC will try to predict. There is no use having N too large since the future reference is unknown and can only be guessed. Also, the model used internally in the MPC

is more reliable close to the starting point, making a large N unnecessary.

Another step in the design of an MPC is choosing a reference to follow. To see what difference the reference can make in performance, both references from Section 4.1 are tested. The MPC also needs future values of the reference which means that it has to predict the driver's behaviour. A simple way to do this is to interpolate the derivative of the current reference signal and the previous one. This can be described as

$$r(k+1) = r(k) + \frac{1}{T_s}(r(k) - r(k-1))T_s = 2r(k) - r(k-1) \quad (4.28)$$

where the derivative of $r(k)$ is calculated using *Euler backward*.

The last designer choice is which model to use for the internal model. As described earlier the MPC needs a linear model. From Section 3.3 there are two to choose from with different advantages and drawbacks. Therefore both models are tried.

4.3.4 Observer

When using the black-box model in Section 3.3, the states do not have any physical representation and are therefore not possible to measure. To still be able to feed back the states to the MPC an observer is used. The observer used in this thesis is a Kalman Filter. How an observer and the Kalman Filter works can be read in [8] and the implementation is inspired by [22].

5

Results

In this chapter the result of the P controller and MPC with different references are presented. The MPC is also tested with the different internal models presented in Section 3.3. From this, the result of the best performing P controller and MPC are compared to each other and the currently implemented controller. Finally, the different simulation models, used for simulating the controllers, are compared.

5.1 P Controller

The performance of the P controller when running it on the real system, with tuned parameters, can be seen in Figure 5.1. The simpler reference using integrated velocity is used on the left hand side and the more advanced reference is used to the right. The control signals for both references can be seen in Figure 5.2. From what can be seen in these figures the heights with both references are very similar, the difference between them can instead be seen in the currents where the advanced reference falls behind and acts slower. The reason for keeping the main lift current at about 500 mA and the free lift at 300 mA is to make sure that they really reach the bottom.

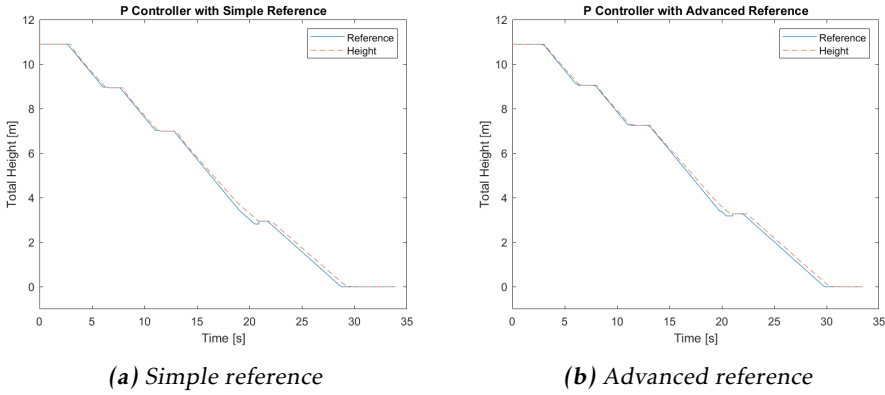


Figure 5.1: The performance of the forklift with P control and a joystick signal from Meas2.

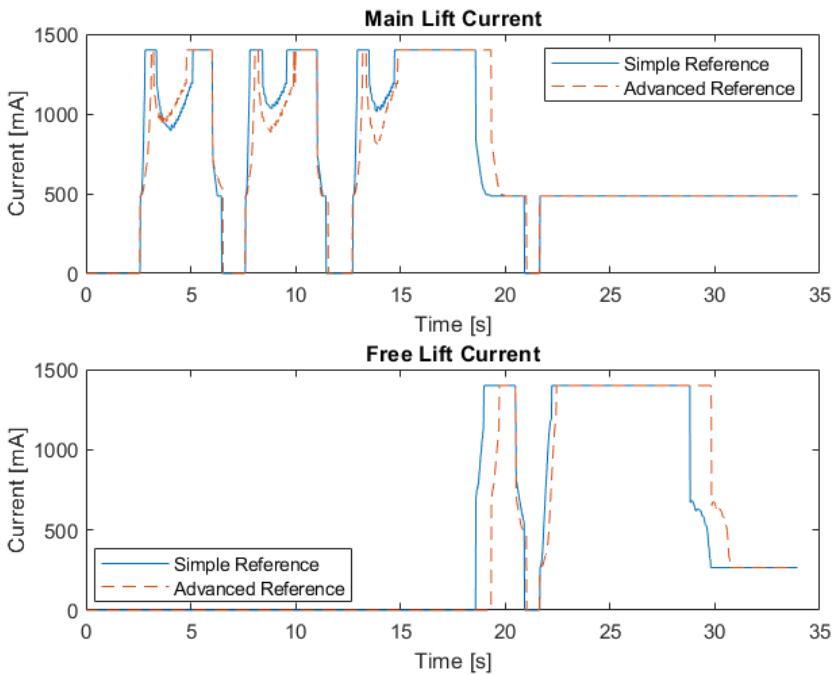


Figure 5.2: The control signals, currents for main lift and free lift, used for the two different references.

It is of interest to see what happens with the height when the forks start and stop. Figure 5.3 and 5.4 are zoomed in on the second and third stop in Figure 5.1. The second stop is in the main lift region and the third stop is in the free lift region.

The different references lead to very similar behaviour at the second stop. In the third stop there can be seen a difference both when stopping and starting. In both Figure 5.3 and 5.4 it can be seen that the advanced reference has smoother stops and starts. However, the advanced reference is also slower. This can, for the stops, be seen more clearly in Figure 5.5 where it takes 0.52 s for the forks in Figure 5.5a to stop and 0.64 s for the forks in Figure 5.5b. The starts are zoomed in in Figure 5.6. For the height in Figure 5.6a it takes 0.62s to move 0.06 m and the corresponding time for Figure 5.6b is 0.88 s.

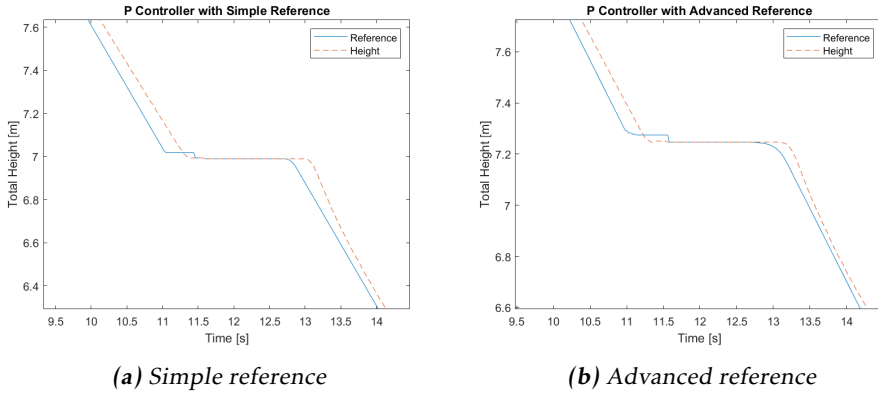


Figure 5.3: Zoomed in heights for the second stop in Figure 5.1

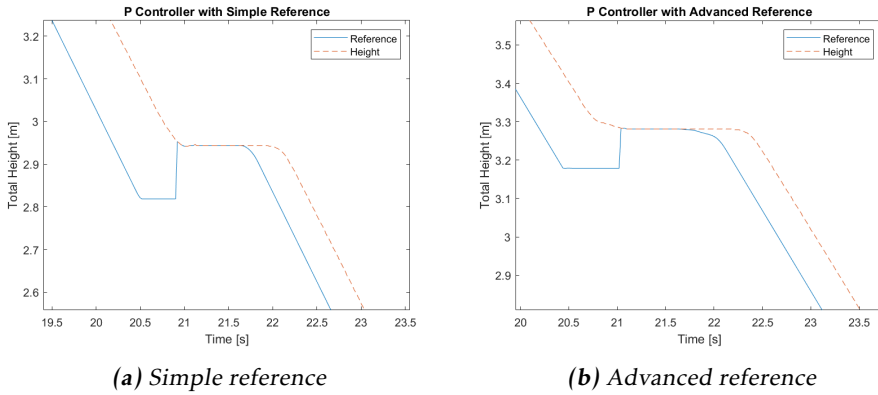
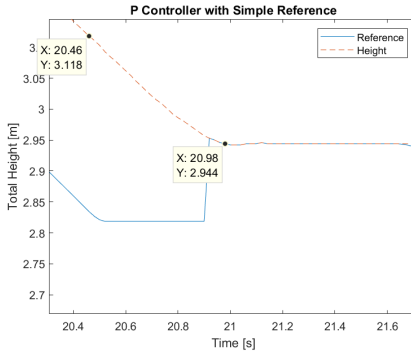
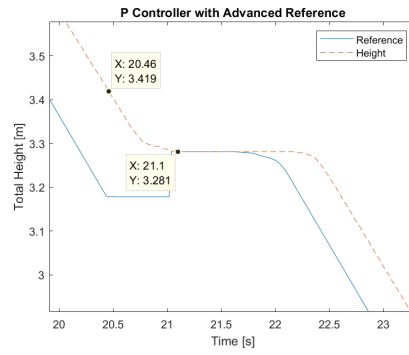


Figure 5.4: Zoomed in heights for the third stop in Figure 5.1.

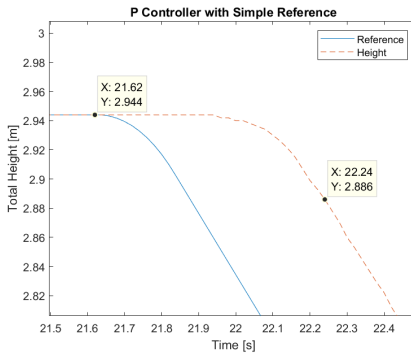


(a) Simple reference

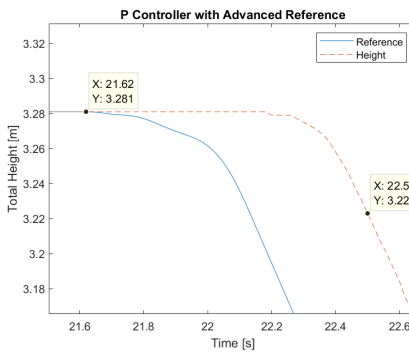


(b) Advanced reference

Figure 5.5: Zoomed in heights for the stop in Figure 5.4. The first timepoint ($X=20.46$) indicates the moment when the joystick is released. The latter timepoint highlights when the forks stopped.



(a) Simple reference



(b) Advanced reference

Figure 5.6: Zoomed in heights for the start in Figure 5.4. The first timepoint in each figure ($X = 21.62$) is the time when the joystick is pushed and the other when the forks have moved 0.06 m.

5.2 MPC

The first test for the MPC is a comparison of using the handmade or the estimated model from Section 3.3. The results can be seen in Figure 5.7 and 5.8. From Figure 5.7 it can be seen that after the MPC switches from using the main lift to using the free lift, at the time 18 s, both models start to differ from the reference. However, the estimated model has a more obvious difference than the handmade model, which can be explained when looking at Figure 5.8. The current for the free lift never goes above 700 mA for the estimated model while the main current continues on maximum current. This suggest that the estimated model does not capture the fact that the main lift has reached its endpoint and only the free lift current can affect the height of the forks. On the other hand, this property of the system is enforced in the handmade model.

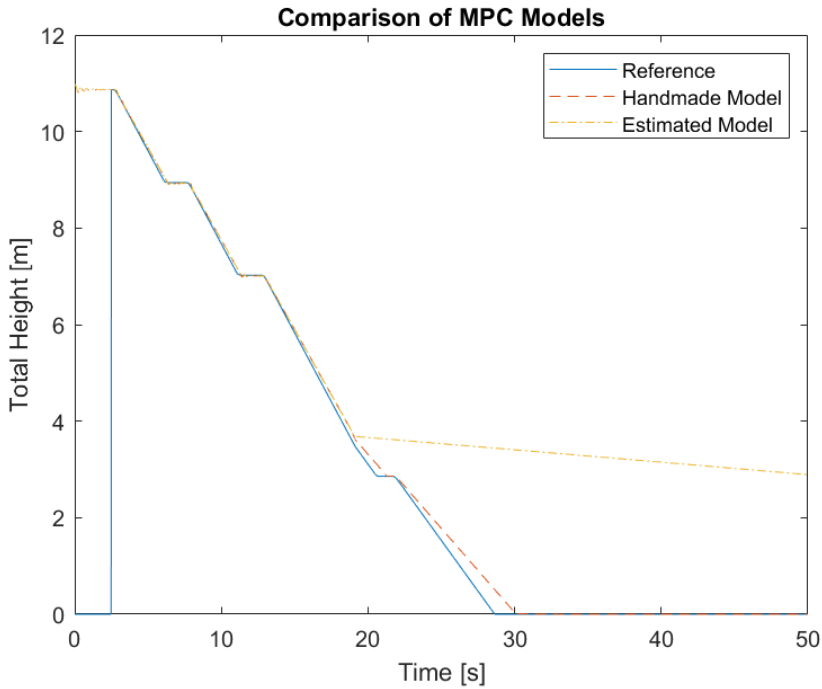


Figure 5.7: The total height for different internal models.

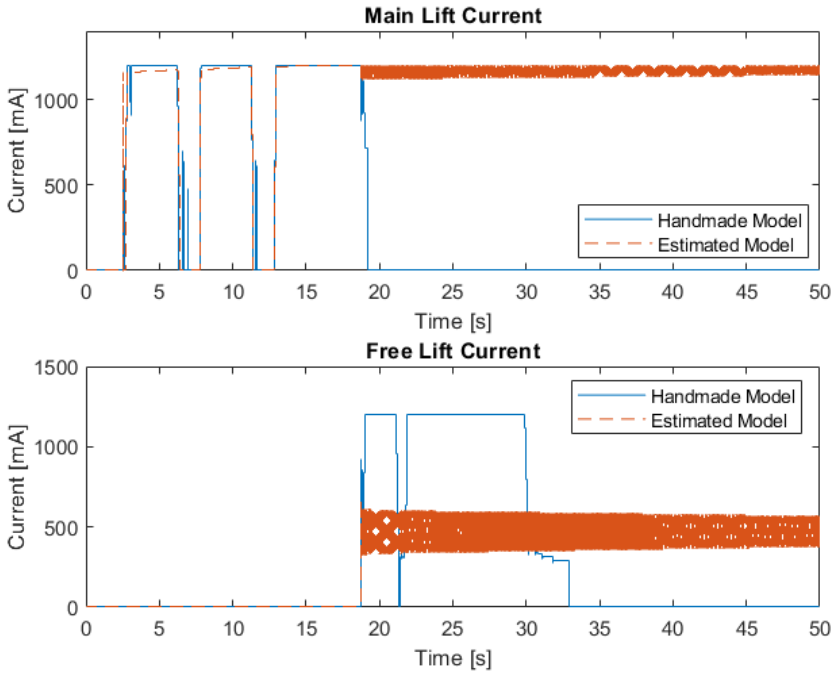


Figure 5.8: The main and free current when using different internal models in the MPC.

The result when using different references for the MPC on the real system, with the handmade model as internal model, can be seen in Figure 5.9 and 5.10. The behaviour is very similar but some things that should be noticed is that the MPC with the advanced reference takes longer to reach the bottom and the current is slightly more oscillating than for the simple reference.

Figure 5.11 shows the heights of the main lift and the free lift separately in the transfer region, when using the simpler reference. The main lift hits the bottom a bit hard which leads to the bump in the free lift—marked with the red circle. The behaviour is similar when using the advanced reference.

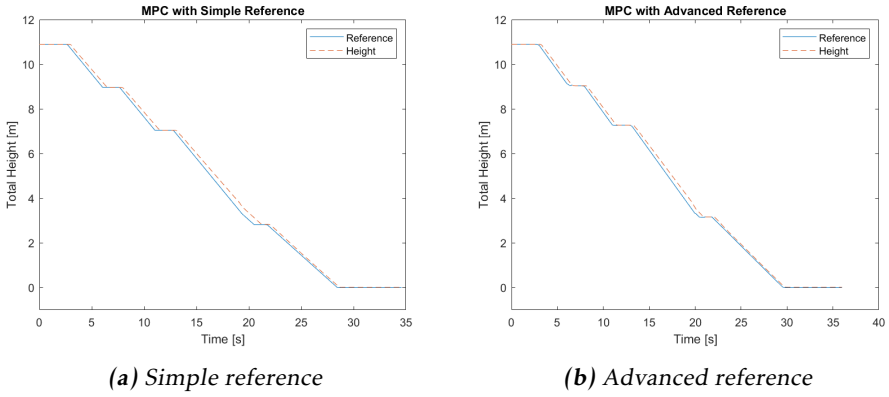


Figure 5.9: The difference in height when using the simple and advanced reference for MPC, respectively.

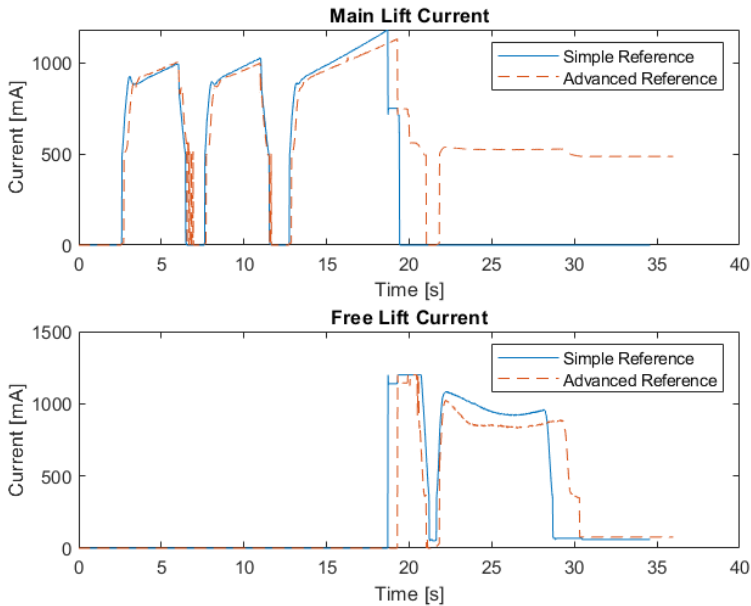


Figure 5.10: The main and free current when using different references in the MPC.

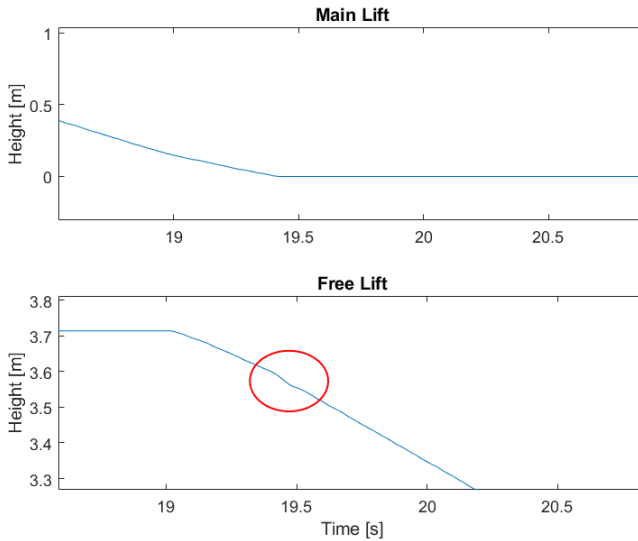


Figure 5.11: Zoomed in heights of the main lift and free lift in the transfer region. The red circle marks a bump in the free lift height at the same instance the main lift reaches the bottom.

5.3 Comparison of Controllers

The two controllers designed in this thesis are compared with the currently implemented controller. The total height of all controllers can be seen in Figure 5.12 and the corresponding currents in Figure 5.13.

There are some parts of Figure 5.12 that are of special interest: the starts, the stops and the transfer region. These can be seen in Figure 5.14-5.18, which show zoomed in heights at the interesting parts. The start of the movement can be seen in Figure 5.14 and the corresponding current in Figure 5.15. There it can be seen that the present controller starts first. After comes the P controller, which also passes the currently implemented controller. The MPC is the slowest.

The first stop is shown in Figure 5.16; the joystick signal, see Figure A.2a, becomes zero at 6.1 s. The time it takes for the different controllers to stop the forks is 0.5 s for the current one, 0.4 s for the P controller and 0.6 s for the MPC. For the present code it takes another 0.5 s for the oscillations to diminish.

The third figure, Figure 5.17, aims to highlight the transfer region. This figure also shows the third stop, which is in the free lift region. Here, the stop time is 0.5 s for both the P controller and the present controller and 0.6 s for the MPC.

In Figure 5.18 we can observe how the controllers work when reaching the bottom. In particular we notice that the currently implemented controller hits the

ground harder than the P controller. The MPC also has a smoother stop, however, it does not quite reach the bottom; it stops 0.02 m above the ground.

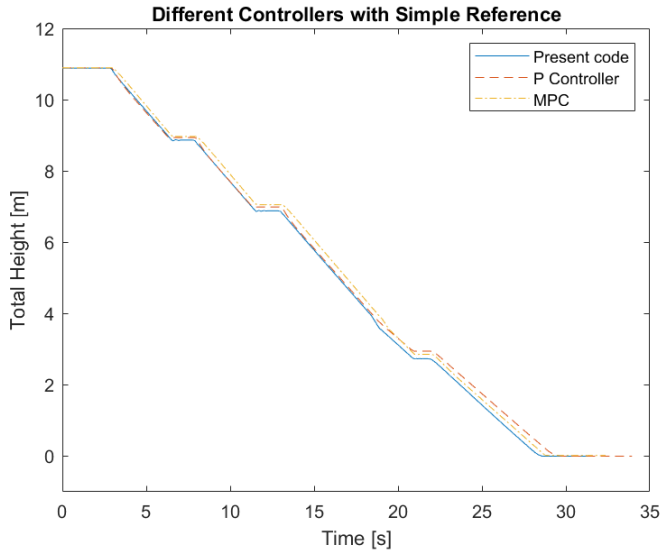


Figure 5.12: Comparison of the height for all three controllers: the present, the PID and the MPC.

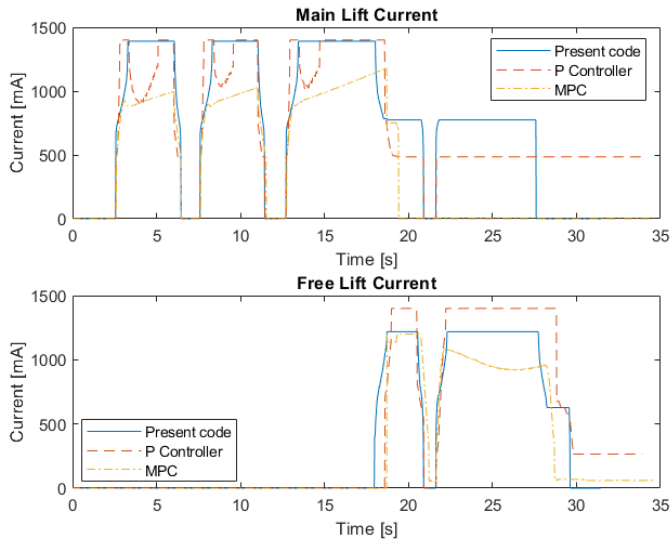


Figure 5.13: Comparison of the current for all three controllers: the present, the PID and the MPC.

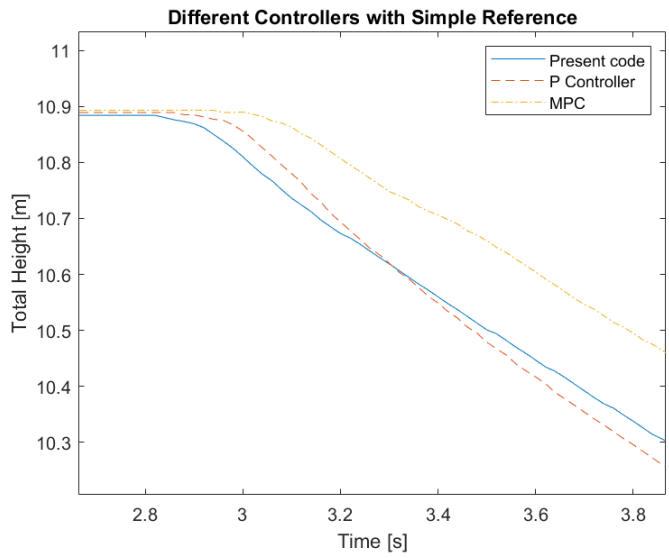


Figure 5.14: Zoomed in at the start of the movement in Figure 5.12.

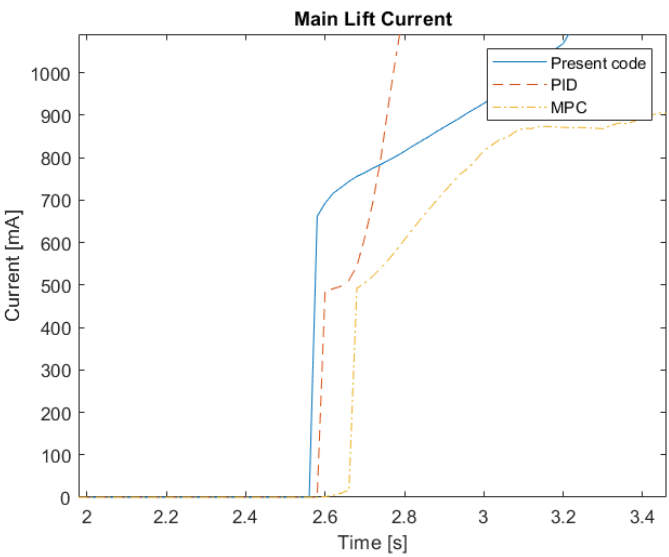


Figure 5.15: Zoomed in height at the start of the movement in Figure 5.13.

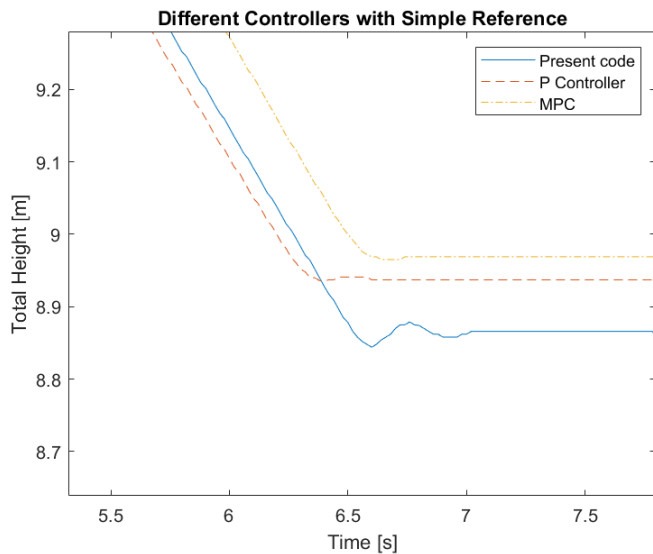


Figure 5.16: Zoomed in height at the first stop in Figure 5.12.

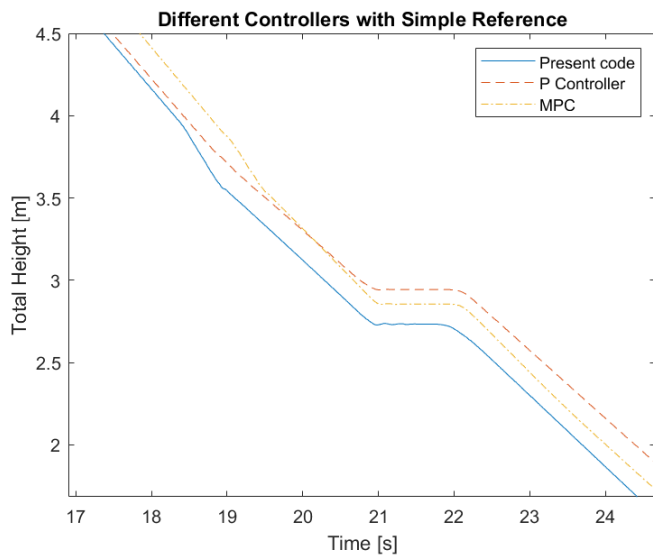


Figure 5.17: Zoomed in height at the transfer region in Figure 5.12.

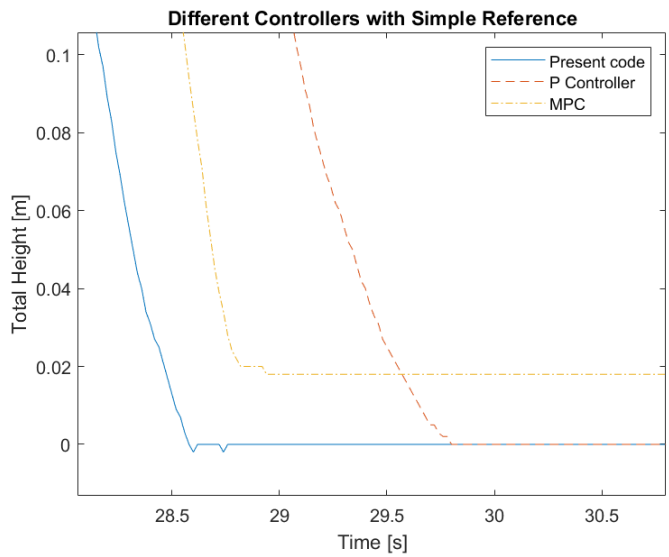


Figure 5.18: Zoomed in height at the bottom in Figure 5.12.

5.4 Comparison of Simulation Models

Both controllers, the P controller and the MPC, are simulated with both simulation models, the physical and the nonlinear ARX from Chapter 3. Moreover, they are also compared to the physical system. The results for the P controller can be seen in Figure 5.19 and Figure 5.20. The corresponding figures for the MPC are Figure 5.21 and Figure 5.22. It can be noticed that the output obtained using the physical model has a similar shape as one from the real forklift, the red line seen in Figure 5.19 or Figure 5.21. On the other hand, when using the nonlinear model the output does not reach the bottom, in Figure 5.19b. The current for the nonlinear model used in both the P controller and the MPC is different than for the other two.

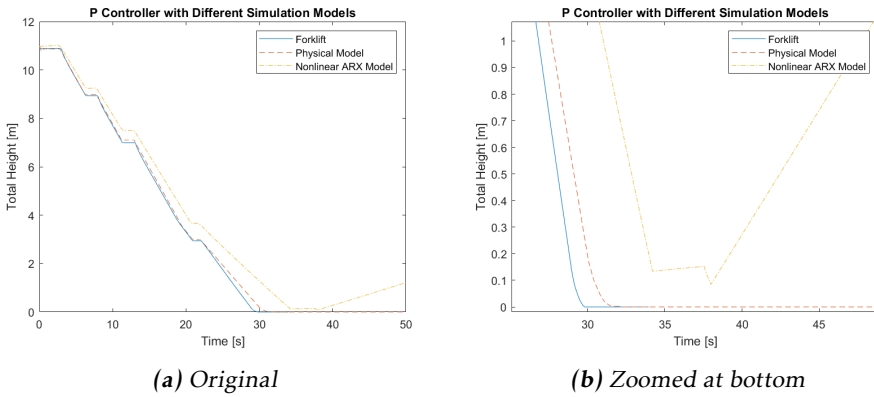


Figure 5.19: The forklift height for the P controller simulated with both the physical model and the nonlinear model compared with height measurements from the real forklift with P control.

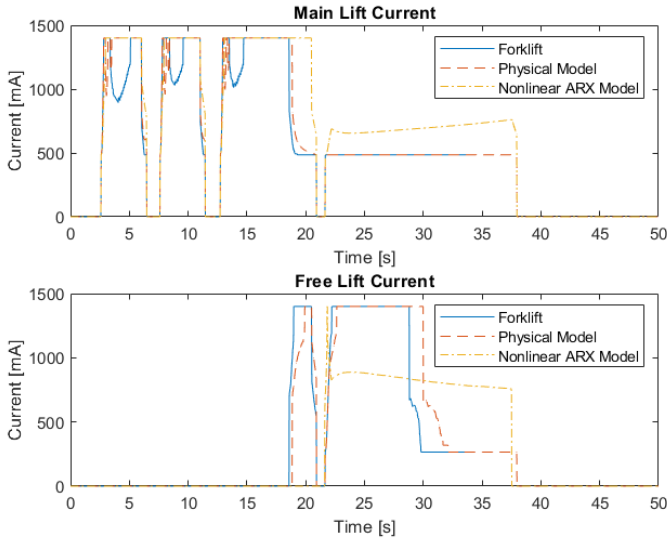


Figure 5.20: The current used by the P controller in both simulation models and on the real forklift.

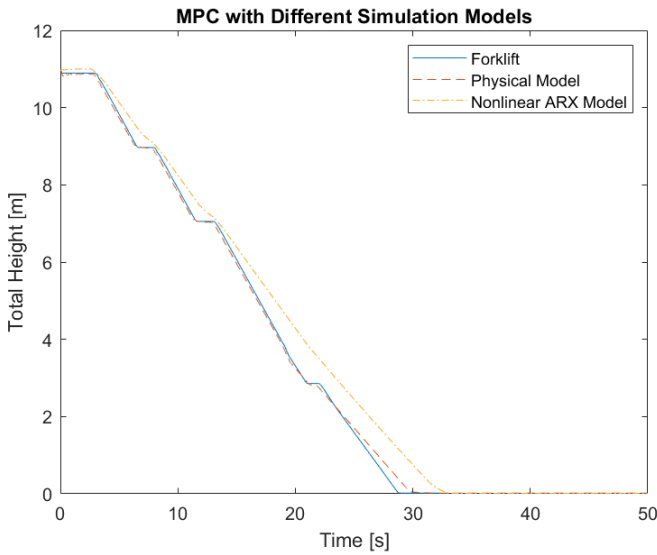


Figure 5.21: The forklift height for the MPC simulated with both the physical model and the nonlinear model compared with height measurements from the real forklift with MPC.

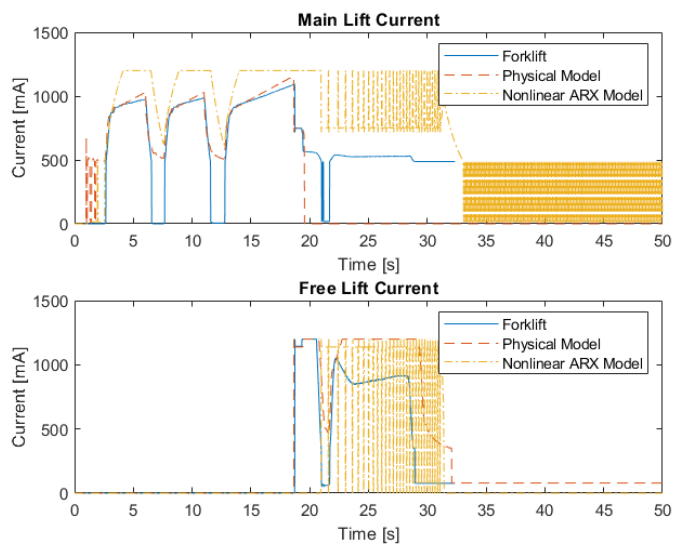


Figure 5.22: The current used by the MPC in both simulation models and on the real forklift.

6

Discussion

This chapter contains a discussion of the results in Chapter 5 and on the development method in Section 1.3. Furthermore, future work is proposed and finally conclusions are drawn that answer the objective questions in Section 1.2.

6.1 Models

A large part of using Model-Based Development and Rapid Control Prototyping is the simulation of the controller before using it on the real system. For this, a simulation model is used and in this thesis two different models for this purpose were presented in Chapter 3. For a simulation model to be useful it needs to be able to ensure the control designer that the controller mainly works as intended. This leads to faster development time, as it is often faster to simulate than to run on the real system. It also reduces the risk of damaging any of the hardware.

The black-box model in Section 3.2.2 has a good fit to the validation data but also contains some nonphysical behaviour. This can for example be seen when comparing the height in Figure 5.19 to the current in Figure 5.20. The model increases the height when the current is zero which is physically impossible.

Even the physical model that is presented is not fault free and shows some behaviours that do not coincide with the real system. For example, the free lift is slower for the model than the real system, which can be seen in Figure 5.19 and Figure 5.21. However, for the most part the model works in a similar way as the physical system. This makes it possible to draw conclusions whether the controller is likely to work on the real system. However, the errors mentioned earlier make tuning the controller in the model insufficient, as the parameters that work well for the model do not always do so for the real system.

For the MPC, in Figure 5.21, the height does not rise for the nonlinear ARX model as it does for the P controller in Figure 5.19. The reason for this is that the MPC sends out a current when the height starts to rise and not that the model is working appropriately. The internal model in the MPC does not contain this behaviour and is therefore unable to predict the lifting. This leads to an oscillating current, as can be seen in Figure 5.22.

The behaviours described for both the P controller and the MPC in the estimated model make it impossible to draw conclusions whether the controller is likely to work on the real system. Having an inadequate simulation model reduces the benefits of simulating the system since it makes it difficult to draw conclusions of the behavior.

In conclusion, the nonlinear model does not fulfill its purpose of realistic simulations of the system. The physical model on the other hand can be useful as a first judgement if the controller will behave in a proper way. However, for tuning the design parameters a better simulation model is needed.

6.2 Controllers

Both controllers presented in this thesis have, in at least some aspects, shown better behaviours than the currently implemented controller. For example, when the forks stop in mid-air both controllers are almost completely free from oscillations and even the transfer region is smoother than with the present control, as described in Section 2.3.2.

6.2.1 P Controller

A P controller with gain scheduling has been constructed with two alternatives for the reference signal. Both references give acceptable behaviors but there are some differences worth mentioning.

The stops and starts, zoomed in in Figure 5.5 and Figure 5.6 are slower for the advanced reference than for the simple reference. On the other hand the simple reference causes small bumps right when the forks stop, as can be seen in Figure 5.4.

Since the advanced reference does not provide a clearly better result, the simple reference is judged as the better one. The simple reference is faster and needs less computing power. In practice the smoother starts in the advanced reference are only experienced as slower and not safer.

6.2.2 MPC

From the results in Figure 5.7 and 5.8 it can be concluded that even though the estimated model shows better performance in Section 3.3, the handmade model has a better performance when using it as an internal model. This could be explained by its physical nature. Since the estimated model is not judged safe to be

used on the actual hardware, the handmade model is the only one tested on the real hardware.

From Figure 5.9 the conclusion can be drawn that the more advanced reference does not necessarily lead to better performance. In this case the simple reference even leads to better performance, since it is both faster, uses a smoother control signal and demands less computing power.

The transfer region is not perfect, as can be seen in Figure 5.11. The main lift does not slow down as much as desired, which leads to a hard stop in the main lift cylinder. This is a consequence of the very fast down ramping of the main current and the very fast up ramping of the free current, as seen in Figure 5.10. The MPC is implemented so that the forks follow a reference for the total height. To assure a smoother stop when reaching the end of the main lift there is a constraint that the height of the main lift can not be negative, see (4.27). It was not possible to make the transfer region bump free with only tuning the weight matrices Q_1 , Q_2 and Q_3 in (4.22). This is probably due to the fact that the MPC is a linear controller while the real system is nonlinear, especially where the valves are nearly closed — which are the problematic regions.

6.2.3 Comparison of Controllers

When the forks stop in the main lift, which can be seen in Figure 5.16, the P controller is able to stop the forks 0.1 s faster than the present controller and still obtain a smooth stop. The faster stops can also be seen in the current in Figure 5.13 where the current is decreasing faster for the P controller than for the currently implemented controller. The MPC on the other hand stops the forks in the same amount of time but is better in the aspect that the forks do not oscillate.

In the transfer region, shown in Figure 5.17, the velocity for the present controller clearly speeds up and then creates a small bump. The same phenomenon can be seen in the MPC but not as strongly as for the present controller. For the P controller on the other hand, it is not apparent exactly where the transfer region is which means that it is very smooth.

At the start of a movement the present controller is faster than both the P controller and the MPC. This is mainly due to the fact that both the P controller and the MPC have softer starts, which can be seen in Figure 5.14. This is also confirmed by looking at the main lift current in Figure 5.15. Here, both the currents for the P controller and MPC take longer time to reach a level of about 700 mA, which in practice indicates a current with little impact.

Another advantage of both the P controller and the MPC is that they do not hit the ground as hard as the present controller, as can be seen in Figure 5.18. Between the P controller and MPC, the P controller can be judged as the better one since it goes all the way down to zero.

In general the MPC is using a lower current than both the P controller and the present controller, see Figure 5.13, without being much slower, see Figure 5.12. A lower current is better from an energy consumption point of view. On the other

hand, the power saved by the controller is reasonably neglected compared to that of the whole forklift.

A clear advantage of the MPC is that it takes constraints into consideration. This means that if the internal model is good enough the speed limit can be guaranteed to be satisfied. A consequence is that the marginal can drastically be decreased without breaking any laws. This is not the case for the P controller where additional tests are needed to make sure that it does not exceed the speed limit.

We conclude that developed controllers have a more preferable behaviour than the currently implemented one. Between the two, the P controller and the MPC, the P controller is preferred since it stops faster and has a better transfer region. This indicates that a nonlinear controller is more preferable than an advanced linear controller.

6.3 Rapid Control Prototyping

Through out this thesis, one of the focuses has been working with Model-Based Design (MBD) and especially Rapid Control Prototyping (RCP). Working model-based has made it possible to focus on the design instead of the implementation. Furthermore, building models in a modular fashion has made it possible to make changes in references and internal models without having to redo the entire implementation.

Working with RCP has made it possible to test two different control methods, both with two different references and one of them with two different internal models, in a short amount of time. The working flow has been the following:

1. Decide on the structure of the controller
2. Implement in Simulink
3. Test using the simulation model
4. Test on hardware

Depending on the result of step three, steps one to three are repeated until the result is good enough to try on the hardware. When reaching the final step, steps one to four are repeated—step three can be skipped depending on how extensive the changes are—until the results are satisfying.

Rapid Control Prototyping has been the part of MBD that this thesis has focused on. RCP requires additional hardware that has to be connected to the hardware that is to be tested. This process was time consuming with the used system. However, when it was done, the part of the development process that needed the hardware, the forklift and real-time target machine, was fairly small. When using a good enough model the main part that needed to be done on the actual hardware was tuning and possibly finding problems that did not appear in the model.

As mentioned in Section 1.3.2 Lean has two core values, respecting the people

and continuous improvement. The work in this thesis, constructing two different controllers, has mainly involved two persons. This means that the information flow is simple and therefore results can be achieved without taking regards to the first value. The second value however, has been a constant core in the work. This has been achieved by starting with an idea and continuously developing it, for example by changing the reference signal the or internal model depending on results from both simulations and the actual hardware.

One of the key points in Lean was to eliminate waste. By quickly being able to try new ideas, the time waste has been reduced by early eliminating the ideas that did not work. This could often be done as early as in step three in the work flow above or in step four if it was faults not included in the model.

The disadvantage that made itself noticeable early on was the long start up time. Before being able to use RCP, a simulation model that captures most of the system dynamics is needed. Even after completing a simulation model the computing hardware needs to be connected to the plant which is not necessarily accustomed to this. Some other disadvantages that did not affect this thesis but could be a threat are both the expensive hardware and licences that are needed. If the developers are not used to working with models this could also prolong the start up time.

The conclusion of working with RCP is that it made it possible to test many different ideas quickly even though the start up time was long. The start up time leads to it being more useful in the longer run than in the shorter. It also keeps complex systems understandable unlike code projects. First testing control systems in a simulation model made the testing on the actual hardware safer. However, in this part there is room for improvement as a better model has better tuning opportunities. This would make the development process both safer and probably give better results since more tuning can be done faster.

6.4 Future Work

The controllers presented in this thesis have not taken into consideration the weight of the load. However, a complete control system for the lowering should preferably work the same for any weight that is acceptable as load. The P controller can be tuned for different weights but no set of parameters that could work in an acceptable manner for all loads was found. This means that the parameters need to automatically change during run time in some way. One way to do this is to make use of the pressure sensor. The load on the forks largely affects the pressure and, similar the height that is used in the feedback presented here, the pressure could be added to the feedback. As the pressure depends on many different things other than the weight, such as the speed and height of the forks, this data would need to be processed in a more thoroughly manner than there was time for in this thesis project.

Another measured signal that was not used in the designed controllers was the

temperature of the oil in the hydraulic system. The speed of the forks is faster with the same valve opening if the oil is warmer. Using temperature as feedback could lead to a more constant speed over time and also make it possible to overall get closer to the maximum allowed velocity, as there is no need for an extra margin.

A small portion of time was spent on refining the physical simulation model that was provided at the start of the thesis. As can be seen in Figure 3.2 and 3.3 this model can be further developed with new knowledge of the system to increase the model accuracy. A better model could make more parameter tuning possible in a simulation environment. This would make it possible to try more parameters and probably increase the performance of both the P controller and the MPC.

To improve the performance of the MPC in the transfer region, the implementation can be changed so that the reference for the main and free lift are separate. This could make the main lift slow down more and the current for the free lift to ramp up. Together they lead to a smaller value of the goal function.

6.5 Final Conclusion

The questions to be answered by this thesis from Section 1.2 were:

1. What control strategy can be used for the lowering function of the RRE fork-lift?
2. Is RCP a useful inclusion in the development process for TMH?

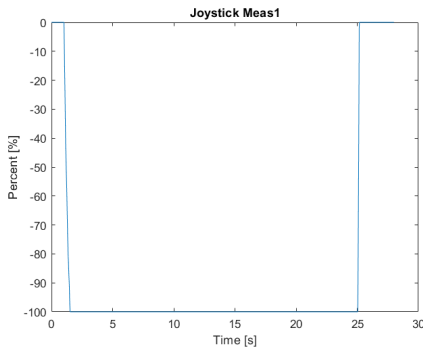
The first question is answered by that both the P controller and the MPC can be used for the lowering and outperform the present implemented control system. Of the two designed controllers, the P controller has the better performance, probably due to its nonlinearities.

The answer to the second question is yes. RCP is useful in the sense that in a short amount of time many different ideas can be tried out. The disadvantage is the long start up time but in the long run this has a small impact on the total time.

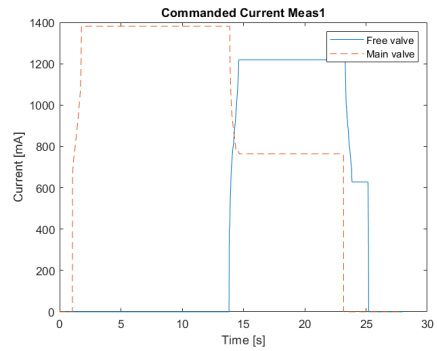
Appendix

A

Measurement Data

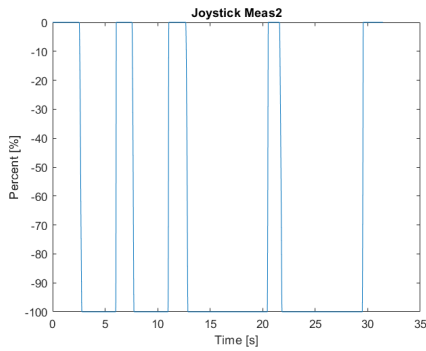


(a) Joystick signal

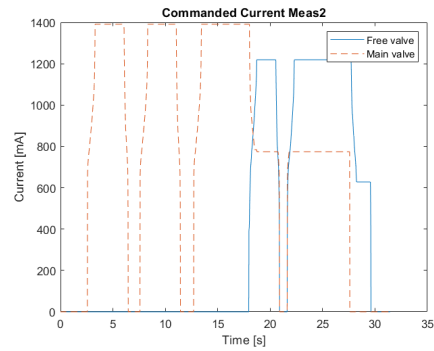


(b) Commanded current

Figure A.1: The measured joystick signal and the commanded current based on the joystick from Meas1.

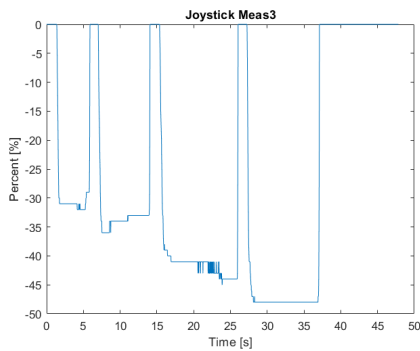


(a) Joystick signal

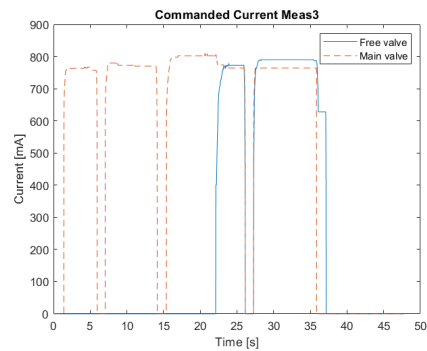


(b) Commanded current

Figure A.2: The measured joystick signal and the commanded current based on the joystick from Meas2.

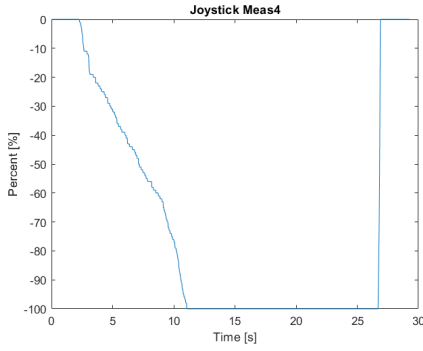


(a) Joystick signal

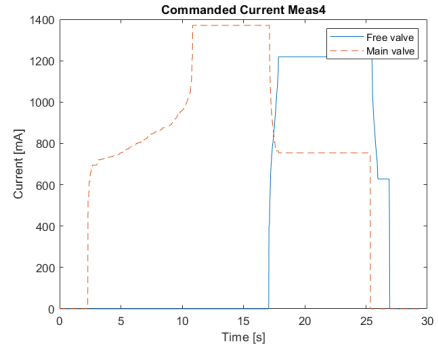


(b) Commanded current

Figure A.3: The measured joystick signal and the commanded current based on the joystick from Meas3.

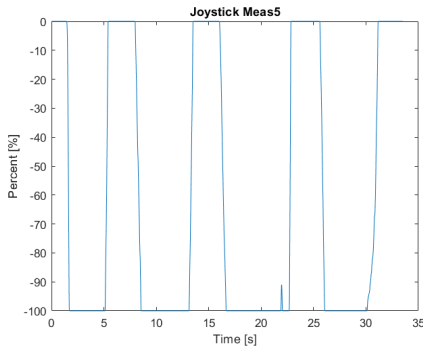


(a) Joystick signal

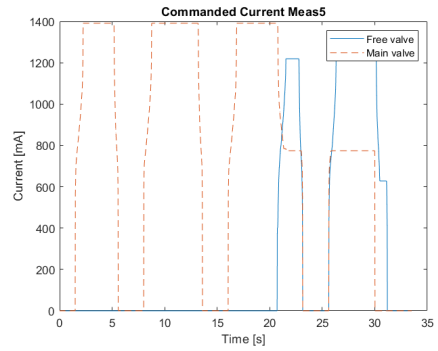


(b) Commanded current

Figure A.4: The measured joystick signal and the commanded current based on the joystick from Meas4.



(a) Joystick signal



(b) Commanded current

Figure A.5: The measured joystick signal and the commanded current based on the joystick from Meas5.

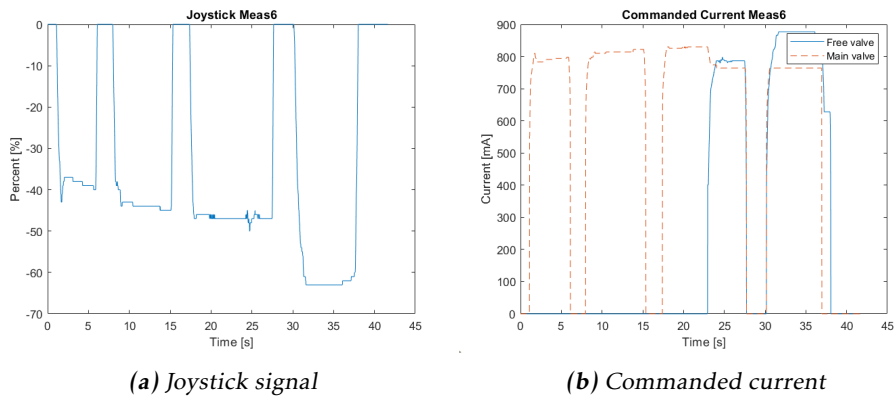


Figure A.6: The measured joystick signal and the commanded current based on the joystick from Meas6.

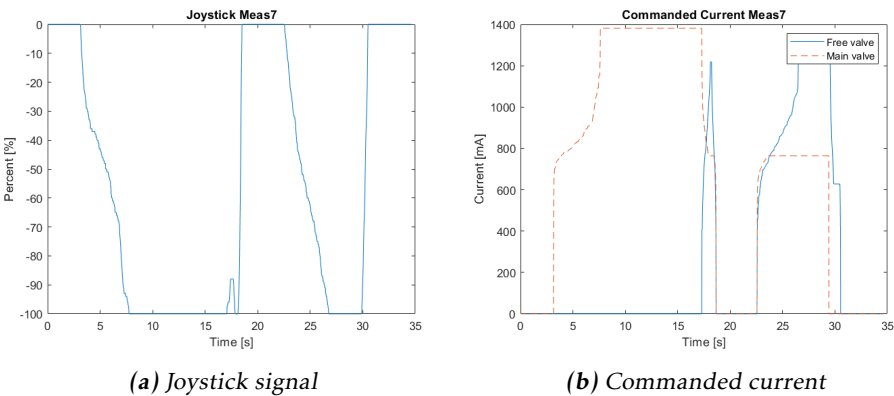
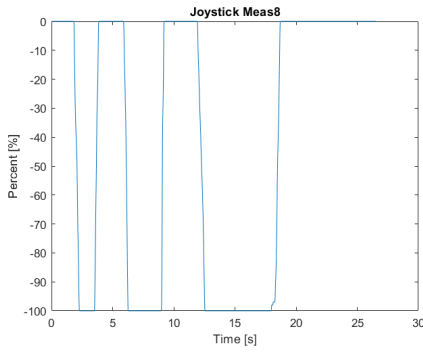
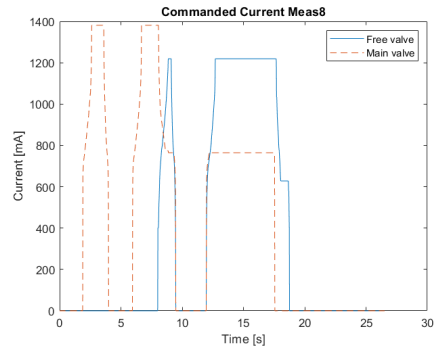


Figure A.7: The measured joystick signal and the commanded current based on the joystick from Meas7.

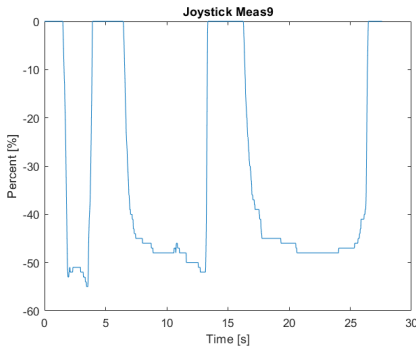


(a) Joystick signal

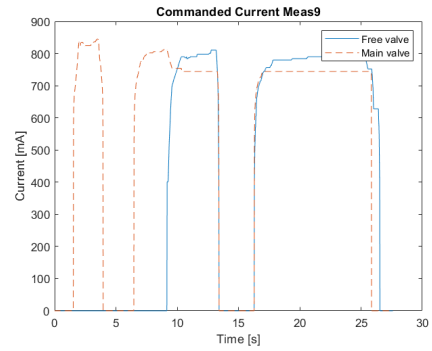


(b) Commanded current

Figure A.8: The measured joystick signal and the commanded current based on the joystick from Meas8.



(a) Joystick signal



(b) Commanded current

Figure A.9: The measured joystick signal and the commanded current based on the joystick from Meas9.

B

Validation Data

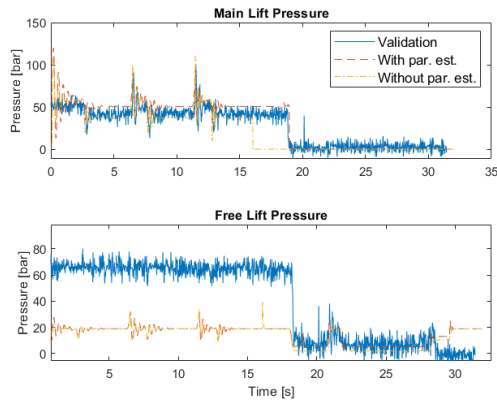


Figure B.1: Model validation for pressures using *Meas2*. The solid blue line shows the measured data, the red dashed line shows the model simulated with the estimated parameters and the dashed-dotted yellow line shows the model simulated with the original parameters.

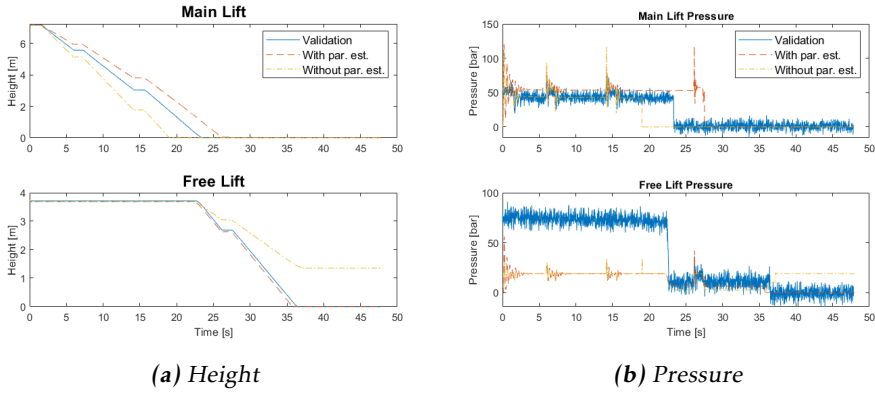


Figure B.2: Model validation for heights and pressures using *Meas3*. The solid blue line shows the measured data, the red dashed line shows the model simulated with the estimated parameters and the dashed-dotted yellow line shows the model simulated with the original parameters.

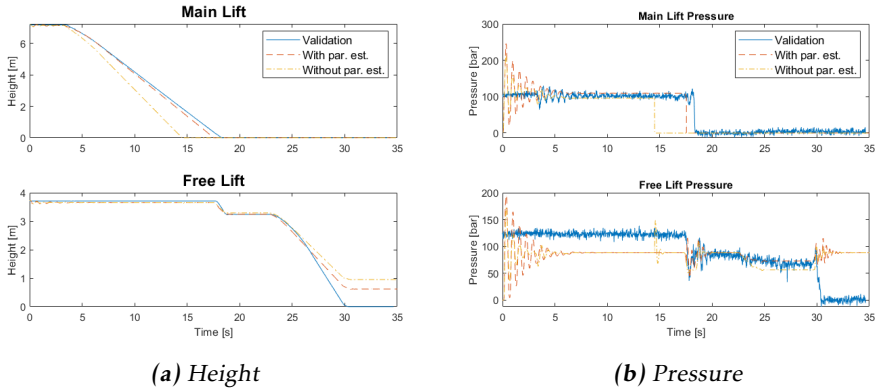


Figure B.3: Model validation for heights and pressures using *Meas4*. The solid blue line shows the measured data, the red dashed line shows the model simulated with the estimated parameters and the dashed-dotted yellow line shows the model simulated with the original parameters.

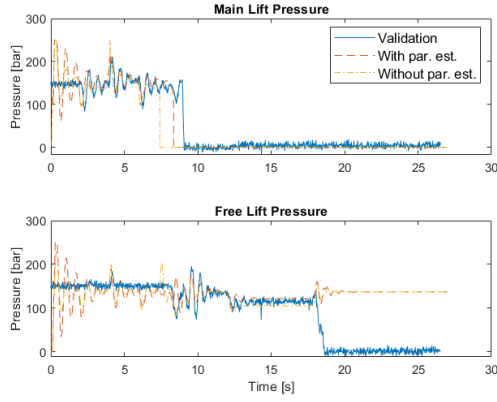


Figure B.4: Model validation for pressure using Meas8. The solid blue line shows the measured data, the red dashed line shows the model simulated with the estimated parameters and the dashed-dotted yellow line shows the model simulated with the original parameters.

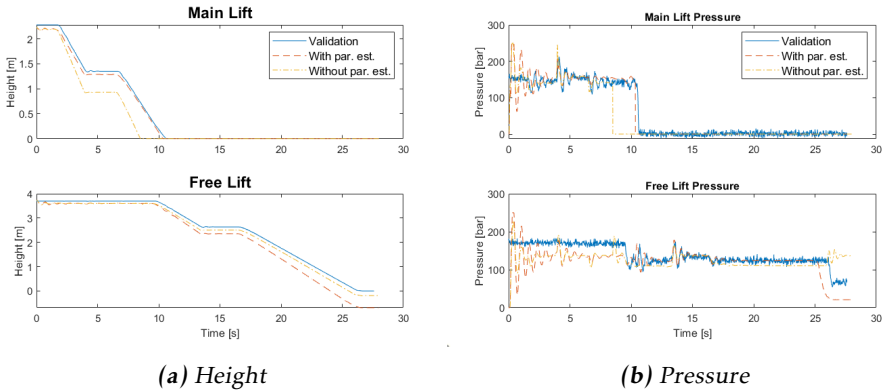


Figure B.5: Model validation for heights and pressures using Meas9. The solid blue line shows the measured data, the red dashed line shows the model simulated with the estimated parameters and the dashed-dotted yellow line shows the model simulated with the original parameters.

Bibliography

- [1] Roger Aarenstrup. *Managing Model-Based Design*. The MathWorks, Inc, 2015.
- [2] Anders Brändström and Henrik Bäckman. Modelling and control of an electro-hydraulic forklift. Master's thesis, Linköping University, 2016. ISRN: LiTH-ISY-EX-16/4970-SE.
- [3] dSPACE. dspace prototyping systems, 2019. URL <https://www.dspace.com/en/inc/home/products/systems/functp.cfm>.
- [4] Martin Enqvist, Torkel Glad, Svante Gunnarsson, Peter Lindskog, Lennart Ljung, Johan Löfberg, Tomas McKelvey, Anders Stenman, and Jan-Erik Strömberg. *Industriell reglerteknik - Kurskompendium*. Reglerteknik, Institutionen för systemteknik, Linköping University, 2014.
- [5] Daniel Fahlén and Ludvig Fri. Modelling and control of a forklift's hydraulic lowering function. Master's thesis, Linköping University, 2017. ISRN: LiTH-ISY-EX-17/5050-SE.
- [6] Torkel Glad and Lennart Ljung. *Reglerteori - Flervariabla och olinjära metoder*. Studentlitteratur, 2:8 edition, 2003.
- [7] Torkel Glad and Lennart Ljung. *Modellbygge och simulering*. Studentlitteratur AB, 2:5 edition, 2004.
- [8] Torkel Glad and Lennart Ljung. *Reglerteknik - Grundläggande teori*. Studentlitteratur, 4:12 edition, 2006.
- [9] Toyota Material Handling. Våra truckar, 2015. URL <https://shop.toyota-forklifts.se/vara-truckar/?area=0&sort=Popularity&page=1&f.applicationusage=Skjutstativtruckar%20med%20tiltande%20hytt&f.producttypekey=ProductTypeReachtruck>.
- [10] H. Hanselmann. Automotive control: from concept to experiment to product. In *Proceedings of Joint Conference on Control Applications Intelligent*

- Control and Computer Aided Control System Design*, pages 129–134, Sep. 1996. doi: 10.1109/CACSD.1996.555244.
- [11] Mao-Jun He, Wen-Jian Cai, Bing-Fang Wu Wu, and He Ming. Simple decentralized pid controller design method based on dynamic relative interaction analysis. pages 8334–8344, 2005. doi: 10.1021/ie050327e.
- [12] Vesa Hölttä, Lauri Palmroth, and Lasse Eriksson. Rapid control prototyping tutorial with application examples. *Sim-Serv*, Jan 2004.
- [13] National Instruments. Controller area network (can) overview, 2019. URL <http://www.ni.com/sv-se/innovations/white-papers/06/controller-area-network--can--overview.html>.
- [14] Konstantinos J. Kyriakopoulos and George N. Saridis. Minimum jerk path generation. In *Proc. IEEE Int. Conf. Robotics and Automation*, volume 1, pages 364–369, 1988.
- [15] G.P. Liu and S. Daley. Optimal-tuning nonlinear pid control of hydraulic systems. In *Control Engineering Practice*, volume 8, pages 1045–1053, 2000. doi: [https://doi.org/10.1016/S0967-0661\(00\)00042-3](https://doi.org/10.1016/S0967-0661(00)00042-3).
- [16] MathWorks. Least-squares (model fitting) algorithms, 2019. URL <https://se.mathworks.com/help/optim/ug/least-squares-model-fitting-algorithms.html>.
- [17] MathWorks. Nonlinear arx models, 2019. URL <https://se.mathworks.com/help/ident/nonlinear-arx-models.html>.
- [18] Jelali Mohieddine and Andreas Kroll. *Hydraulic Servo-systems - Modelling, Identification and Control*. Springer, 2003.
- [19] G.B.Z.L. Moreno, R.M. Caliari, R.R. Geraldés, and M.A.L. Moraes. Rapid Control Prototyping Tool for the Sirius High-Dynamic DCM Control System. In *Proc. of International Conference on Accelerator and Large Experimental Control Systems (ICALEPCS'17), Barcelona, Spain, 8-13 October 2017*, pages 1941–1946. ISBN 978-3-95450-193-9. doi: <https://doi.org/10.18429/JACoW-ICALEPCS2017-THPHA214>.
- [20] Gerrit Naus, Roel van den Bleek, Jeroen Ploeg, Bart Scheepers, M.J.G. Molengraft, and M Steinbuch. Explicit mpc design and performance evaluation of an acc stop-&-go. pages 224 – 229, 07 2008. ISBN 978-1-4244-2078-0. doi: 10.1109/ACC.2008.4586495.
- [21] Mary Poppendieck and Tom Poppendieck. *Lean software development : an agile toolkit*. The agile software development series. Addison-Wesley, 2003. ISBN 0321150783. URL <https://login.e.bibl.liu.se/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=cat00115a&AN=lkp.397364&site=eds-live&scope=site>.

- [22] Verica Radisavljevic-Gajic. Linear observers design and implementation. 2014. ISBN 978-1-4799-5233-5 978-1-4799-5232-8. doi: <https://doi.org/10.11092FASEEZone1.2014.6820636>.
- [23] Sppeedgoat. Mobile real-time target machine, 2019. URL <https://www.speedgoat.com/products-services/real-time-target-machines/mobile>.
- [24] Tony A. Wood, Henrik Hesse, and Roy S. Smith. Predictive control of autonomous kites in tow test experiments. pages 110 – 115, 07 2017. doi: 10.1109/LCSYS.2017.2708984.