Model-Based Design of a Fork Control System in Very Narrow Aisle Forklifts

Erik Bodin and Henric Davidsson
Master of Science Thesis in Electrical Engineering

Model-Based Design of a Fork Control System in Very Narrow Aisle Forklifts

Erik Bodin and Henric Davidsson
LiTH-ISY-EX--17/5043--SE

Supervisors:  Måns Klingspor
             ISY, Linköping University
             Håkan Frid, Johan Österskog
             Toyota Material Handling

Examiner:     Johan Löfberg
             ISY, Linköping University

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

Copyright © 2017 Erik Bodin and Henric Davidsson
ok
Abstract

This thesis explains the model-based design of a fork control system in a turret head operated Very Narrow Aisle forklift in order to evaluate and push the limits of the current hardware architecture. The turret head movement consists of two separate motions, traversing and rotation, which both are hydraulically actuated. The plant is thoroughly modeled in the Mathworks softwares SIMULINK / SIMSCAPE to assist in the design of the control system. The control system is designed in SIMULINK / STATEFLOW and code-generated to be evaluated in the actual forklift.

Optimal control theory is used to generate a minimum-jerk trajectory for auto-rotation, that is simultaneous traversing and rotation with the load kept in centre. The new control system is able to control the system within the positioning requirements of ±10 mm and ±9 mrad for traversing and rotation, respectively. It also shows good overall performance in terms of robustness since it has been tested and validated with different loads and on different versions of the forklift. However, the study also shows that the non-linearities of the system, especially in the hydraulic proportional valves, causes problems in a closed-loop control system.

The work serves as a proof of concept for model-based development at the company since the development time of the new control system was significantly lower than for the original control system.
Acknowledgments

First, we would like to thank Johan Österskog and Håkan Frid, our supervisors at Toyota Material Handling, and Måns Klingspor, our supervisor at Linköping University, for all their help. Second of all we would like to thank Andreas Lassi at TMH for all his help with software and hardware related issues.

Last but not least, thanks to Jerry and Elias for the great company and the after-work burgers. However, we sincerely apologize for introducing you to the burgers served at Flamman, we hope you will forgive and forget.

Finally, a big thanks to Mathworks Inc. for their endless supply of awesome engineering software.

*Mjölby, June 2017*

*Erik Bodin and Henric Davidsson*
# Contents

Notation xi

1 Introduction 1
  1.1 Background .................................................. 1
  1.2 Definitions .................................................. 2
  1.3 Problem Formulation ......................................... 3
  1.4 Delimitations ................................................ 4
  1.5 Related Work ................................................ 4
  1.6 Model-Based Design .......................................... 4

2 System Overview 7
  2.1 Hydraulic System ............................................. 8
    2.1.1 Proportional valves .................................... 8
  2.2 Electrical System ........................................... 8
  2.3 Measurements and Sensors ................................... 8
    2.3.1 Rotational sensor ........................................ 9
    2.3.2 Translational sensor ................................... 9
    2.3.3 FCU analog input ........................................ 9

3 Modelling 13
  3.1 Mechanical System ........................................... 13
  3.2 Hydraulic System ........................................... 15
    3.2.1 Proportional valves .................................... 15
    3.2.2 Hydraulic motor ........................................ 17
    3.2.3 Pump .................................................... 17
  3.3 Electrical System ........................................... 20
  3.4 Model Parameters ........................................... 20
    3.4.1 Parameter optimization ................................ 20
    3.4.2 Validation ................................................ 23

4 Control 25
  4.1 Design ....................................................... 25
    4.1.1 Requirements ............................................. 25
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1.2</td>
<td>Reference signal</td>
<td>25</td>
</tr>
<tr>
<td>4.1.3</td>
<td>Feedforward</td>
<td>26</td>
</tr>
<tr>
<td>4.1.4</td>
<td>Feedback</td>
<td>28</td>
</tr>
<tr>
<td>4.1.5</td>
<td>Controller summary</td>
<td>28</td>
</tr>
<tr>
<td>4.2</td>
<td>Optimal Control</td>
<td>29</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Derivation of equations of motion</td>
<td>29</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Minimum jerk trajectory motivation</td>
<td>30</td>
</tr>
<tr>
<td>4.2.3</td>
<td>Optimal control problem</td>
<td>31</td>
</tr>
<tr>
<td>4.2.4</td>
<td>Remarks about the geometric constraints</td>
<td>32</td>
</tr>
<tr>
<td>5</td>
<td>Results</td>
<td>35</td>
</tr>
<tr>
<td>5.1</td>
<td>Optimized Trajectories for Auto-rotation</td>
<td>35</td>
</tr>
<tr>
<td>5.2</td>
<td>Controller Testing</td>
<td>37</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Traversing controller</td>
<td>37</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Rotation controller</td>
<td>37</td>
</tr>
<tr>
<td>5.2.3</td>
<td>Auto-rotation to 180°</td>
<td>38</td>
</tr>
<tr>
<td>5.2.4</td>
<td>Auto-rotation to 90°</td>
<td>41</td>
</tr>
<tr>
<td>5.3</td>
<td>Validation of Traversing Accuracy</td>
<td>42</td>
</tr>
<tr>
<td>6</td>
<td>Conclusions</td>
<td>45</td>
</tr>
<tr>
<td>6.1</td>
<td>Model Accuracy and Complexity</td>
<td>45</td>
</tr>
<tr>
<td>6.2</td>
<td>Control System</td>
<td>46</td>
</tr>
<tr>
<td>6.3</td>
<td>Model-Based Design</td>
<td>46</td>
</tr>
<tr>
<td>6.4</td>
<td>Future Work</td>
<td>47</td>
</tr>
<tr>
<td>Bibliography</td>
<td></td>
<td>49</td>
</tr>
</tbody>
</table>
## Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>TMH</td>
<td>Toyota Material Handling</td>
</tr>
<tr>
<td>VNA</td>
<td>Very Narrow Aisle</td>
</tr>
<tr>
<td>PWM</td>
<td>Pulse Width Modulation</td>
</tr>
<tr>
<td>FCU</td>
<td>Fork Control Unit</td>
</tr>
<tr>
<td>MCU</td>
<td>Main Control Unit</td>
</tr>
<tr>
<td>LSB</td>
<td>Least Significant Bit</td>
</tr>
<tr>
<td>PID</td>
<td>Proportional, integral, differential (controller)</td>
</tr>
<tr>
<td>MBD</td>
<td>Model-Based Design</td>
</tr>
<tr>
<td>VRE</td>
<td>Very Narrow Aisle Rider Electric</td>
</tr>
</tbody>
</table>
This work focuses on the possibilities of improving the control of the turret head movements in the Very Narrow Aisle (VNA) warehouse forklift VRE150 which is manufactured by Toyota Material Handling (TMH). It is carried out in collaboration with TMH and will mainly serve as a basis for further development of VNA warehouse forklifts. The work also serves as a proof of concept of Model-Based Design and automatic code generation. This first chapter will cover definitions, the background of the problem and related work.

1.1 Background

A VNA forklift drives in aisles which are almost as narrow as the forklift itself making it impossible to turn around without leaving the aisle. Because of this, the forklift must be able to handle the pallets from the side, see Figure 1.1. A VNA forklift utilises either shuttle-forks or, as in the figure, a turret head in order to accomplish this.

The forklift industry is currently experiencing a large shift towards fully automatic forklifts. Every major forklift manufacturer including TMH is currently investing a lot of time, effort and money into the development of automatic forklifts. At TMH many of the first generation automatic forklifts are essentially just an automatic version of the corresponding manual forklift. To make it automatic several sensors are added and new software is installed but the fundamental construction is in many cases very similar to the manual version.

This thesis is set to investigate the turret head movement in the VNA forklift VRE150 with the aim to make it fully automatic in the future.
1 Introduction

Standard aisle configuration

Very Narrow Aisle configuration

Figure 1.1: An illustration of the difference between a standard aisle configuration and a VNA configuration. Note the different driving directions of the forklift.

1.2 Definitions

Turret head is the complete mechanical assembly which is handling the pallets. The turret head consists of all parts depicted in Figure 1.2.

Rack is the main part of the turret head. This is what is assembled onto the mast of the forklift. Marked with 1 in Figure 1.2.

Boom is the part assembled onto the rack. Marked with 2 in Figure 1.2.

Pinion is the part connecting the rack and the boom. Marked with 3 in Figure 1.2. Note that the boom is transparent in the figure. The pinion is mounted inside the boom and would otherwise not be visible.

Forks are what is mounted onto the boom. The forks are handling the pallets. Marked with 4 in Figure 1.2.

Load is what is “lying” on the forks. This would in real life be a pallet with something on it but is for simplicity modelled as a solid block. Marked with 5 in Figure 1.2.

Traversing is the translational motion of the boom. A traverse position of 0 means that the boom is in the leftmost position (looking along the positive x-axis in Figure 1.2).
Rotation is the rotational motion of the forks. A rotation position of 0 means that the forks have been rotated to their maximum clockwise position (looking along the negative z-axis in Figure 1.2).

Auto-rotation is the combined traverse and rotational motion from a traverse position of 0 to its maximum and a rotation from 0° to 180° or the other way around. The motions are in general done automatically by pushing a button, hence the name. The rotation can be halved, resulting in a traverse position in the middle and the rotation at 90°.

Figure 1.2: The turret head assembly. Both traverse and rotation are in their zero positions. 1: the rack, 2: the boom, 3 (inside the boom): the pinion, 4: the forks and 5: the load.

1.3 Problem Formulation

The overall problem formulation is to investigate the turret head and benchmark how well it can perform with a refined control strategy. More specifically, if it is possible to achieve a steady-state traversing accuracy of ±10 mm and a steady-state angle accuracy of ±0.5° ≈ 9mrad for the rotating motion. If the current hardware architecture does not reach these requirements, a new concept is proposed which will improve the accuracy and reliability of the system.
Further, the concept of Model-Based Design will be applied to the process which makes the work serve as an evaluation of how to apply this method to the development of other mechatronic systems.

1.4 Delimitations

Only the turret head on the forklift VRE150 will be investigated. The thesis is to serve as a proof of concept of Model-Based Design in general and turret head control specifically, which means that the control algorithms proposed will not work in all possible working conditions. This includes all lifting heights, all loads and different ambient conditions such as temperature. In order to ensure proper function in all operating points, further investigations will need to be carried out.

1.5 Related Work

The turret head system can be seen as a electro-hydraulic servo system, which is well studied in several books such as [10] and [12]. These sources focus mainly on servo-class valves, but serves as a good inspiration for basic hydraulic understanding and modelling techniques. The specific problem of position control of a proportional valve controlled system is studied in [6], [13] and [14]. The general conclusion by these papers is that proportional valves can be used in closed loop position control applications when sophisticated control algorithms are used. The problem with PWM control of solenoid valves has been studied in [8], [11] and in [16], and can be seen as a well proven method for controlling proportional valves. The positioning problem has been solved by several different control algorithms, such as fuzzy adaptive control [6] [15], or with simpler PID control [14] [13].

Several theses dealing with modelling the hydraulic actuation system in various forklifts at the company have been presented throughout the years. In [4] modelling and control of a different forklift from TMH is studied. In [5] the very same system studied in this thesis is modelled, simulated and evaluated. Generally modelling in previous work have been done in Hopsan, an open-source software developed at the Department of Fluid and Mechatronic systems at Linköping University.

1.6 Model-Based Design

Model-Based Design as a method has been used in the industry for quite some time and its popularity has been rising in recent years. The automotive industry have been using MBD for well over 20 years and the same hold for the aerospace industry. The ever rising complexity of today’s systems and the amount of software used in them is a challenging task for today’s development engineers. MBD can in this sense assist development teams cutting development time and costs.
MBD development is centred around a model of a physical system. In the case of physical models this is essentially just a set of differential equations describing the behaviour of the system. Usually models are created of a small subsystem and different models of subsystems are then assembled into a model of the overall system. This enables different development teams working on different subsystems to work in parallel.

By using a model, simulations can be run either at a component level or at a system level. This makes it possible to investigate system performance, feasibility, sensitivity and to detect and identify errors early in the development cycle. It’s even possible to do all this long before there are any hardware prototypes of the system. All this makes it possible to cut development time and cost dramatically compared to traditional development work-flows.

To summarize, Model-Based Design is a proven method used to efficiently develop complex mechatronic systems [1].
This chapter covers the current architecture of the VRE150 forklift which is studied in this work. The rotating motion of the forks is performed by two single-acting pistons which is connected to a chain that turns a rod which the forks are attached to. For translational motion, an hydraulic motor is used which is directly connected to a rod which uses a rack and pinion motion to move along the width of the forklift. In order to illustrate this, the motion is drawn in Figure 2.1.

Figure 2.1: The combined traversing and rotating motion of the forks. Note that the forks are turned by two single-acting cylinders attached to a chain and that a hydraulic motor generates the traversing motion.
2.1 Hydraulic System

The hydraulic system is fed by a single pump which also delivers flow to the lift function. Since the lift function is unused when the turret head operates, the lift section is neglected. The hydraulic system for the turret head is presented in Figure 2.2 and shows the two pistons and the hydraulic motor mentioned above. Both the fork rotation pistons and the traverse motor are controlled by two proportional directional valves. The valves are controlled by the four inputs Q26...Q29, one input for each direction of the two valves. The system features a load-sensing system which always tries to keep the pressure over the proportional valves equal to 7 bar. Also, each hydraulic function has counter-balance valves (DV1, DV2, DV5 and DV6 in Figure 2.2) which damps away pressure spikes and makes the motion more smooth. The other valves are pressure-relief valves included mainly for safety-purposes.

2.1.1 Proportional valves

From a control perspective, the proportional valves are the most important parts of the hydraulic system. They ultimately control the flow of oil to the actuators and hence also the speed. The working principle of a proportional valve is explained in Figure 2.3. The characteristics of the proportional valves are most often given as a relationship between the current through the solenoids to the oil flow through the valve. This relationship is highly non-linear and include a wide dead-zone and a hysteresis effect, see Figure 2.4.

The solenoid currents are controlled by PWM controlled signals with a frequency of 2 kHz. This frequency is superpositioned with another PWM signal with lower frequency at 100 Hz called a "dither" signal which has been proven to reduce the dead-zone and hysteresis [16]. This can partially be explained to be due to stick-slip effects (the spool get "stuck" due to static friction when not in motion) which are reduced when the spool oscillates by the lower frequency.

2.2 Electrical System

The central pump is driven by an electrical motor controlled by a dedicated motor controller. The Fork Control Unit (FCU) sends a requested pump speed to the Main Control Unit (MCU) which forwards it to the motor controller.

The FCU also sets the PWM duty cycles according to the desired flow through the valves. But since the characteristics of the proportional valves are given from current to flow only, the FCU must be able to control the current by adjusting the PWM duty cycle, which can be seen as the voltage over the solenoid.

2.3 Measurements and Sensors

In this section, the sensor solutions for the two motions are presented.
2.3 Measurements and Sensors

2.3.1 Rotational sensor

The rotational motion is currently measured using a RFA4000 type sensor from the company Novotechnik. The sensor is measuring the angle of the rotating axle in a contactless manner using the Hall effect, see Figure 2.5.

The sensor measures angles over a 190° (3.32 rad) span from 85° to 275°. The measured angle is proportional to the sensor output voltage (0.25-4.75 V), see Figure 2.6. This voltage span gives a proportionality constant of 1.36 V/rad. Furthermore, the sensor has an internal resolution of 12 bits giving an angular resolution of 3.32 rad/2^{12} ≈ 0.8 mrad. According to the datasheet, the repeatability of the sensor is 0.1° ≈ 1.7 mrad giving a total maximum internal error of 0.8 + 1.7 = 2.5 mrad ≈ 0.14°.

2.3.2 Translational sensor

The translational motion is measured in a combined bearing and encoder of type BMB-6202 from the company SKF. This encoder is placed directly on the shaft which makes the forks translate sideways by the rack and pinion mechanism. The pinion has a radius of 45 mm which gives that one revolution of the sensor corresponds to a translation of 2π·45 mm ≈ 282 mm. The encoder has a resolution of 32 pulses per revolution, but since it is a quadrature encoder, it is possible to distinguish a total of 128 pulses per revolution. This gives a resolution of 2π/128 ≈ 0.05 rad or together with the rack and pinion mechanism a resolution of 2.2 mm.

2.3.3 FCU analog input

The FCU, already mentioned above, is the central processing unit in the turret head. It has several analog and digital inputs where the sensors described above are connected. The translational sensor is digital which implies that no additional error exist except for the sensor resolution. The rotational sensor however outputs a voltage which is measured by the analog input of the FCU. The analog input have a 10 bit resolution giving a 4.9 mV measuring resolution over the measuring range of 5V. According to the datasheet, the input also has maximum error of 14 mV. Together with the proportionality constant of the rotational sensor these errors correspond to an angle error of 4.9/1.36 + 14/1.36 = 3.6 + 10.3 = 13.9 mrad ≈ 0.8°.

It is clear that the accuracy of the analog sensor is substantially higher than the measuring capability of the FCU unit. However, the maximum input error of 14 mV is mainly due to component tolerances in the measurement circuitry, leakage currents and errors in the A/D-converter itself. This means that the majority of the error can be assumed to be constant, or at the very least slowly varying. Hence it can be assumed that the input error of 14 mV can be eliminated by calibration and the total measurement error of the FCU is then limited by the resolution, 0.2°. Including the sensor error this gives a total error of less than 0.5°, which is the requirement.
Figure 2.2: The hydraulic system in the turret head. Only the important parts are visible due to confidentiality reasons. The other components are various other pressure controlling valves mainly included for safety and damping. The system is supplied from the main pump and the tank through the ports $P$ and $T$ in the bottom.
2.3 Measurements and Sensors

Figure 2.3: A proportional valve. Current is fed through the solenoids A or B. This induces a force which moves the spool (light grey) to the left or right which allows the flow from the supply port, \( P_s \) (and to the tank, \( P_t \)) to the ports \( P_A \) and \( P_B \) to be controlled.

Figure 2.4: The measured relationship between current and flow for one of the four valve inputs. Measurements have been done by the manufacturer. The lower curve correspond to an increasing current and the higher curve to a decreasing current.

Figure 2.5: The hall sensor for the rotary motion. The sensor is mounted in the frame and the magnet on the rotating axle.
Figure 2.6: Output characteristics of the sensor. The constant $a$ is equal to 170 in this case. Taken from the sensor datasheet.
This chapter covers the modelling of the system, which includes three main parts: Hydraulic system, mechanical system and electrical system. Each of these subsystems have been thoroughly studied and modelled in order to cover all essential properties. The modelling is essential in the development of the control algorithm since it minimizes the amount of trial-and-error on the real system.

All parts of the system is modelled in SIMULINK/SIMSCAPE by Mathworks Inc. The system components are described in Chapter 2.

3.1 Mechanical System

The mechanical system is conceptually quite simple. There are four main parts, the carriage, the boom, the pinion and the forks. The carriage acting as the reference frame is rigidly attached to the world/ground frame. The boom and pinion attached to the carriage followed by the forks attached to the boom.

Figure 3.2 shows the implementation in SIMSCAPE MULTIBODY. Each part above has been exported from its CAD-model and imported into SIMSCAPE. This enables SIMSCAPE to automatically calculate the inertia of each component as well as provide a visual 3D animation of how the components move during simulation, see Figure 3.1.

The basics of SIMSCAPE MULTIBODY modelling consists of the connection of reference frames (coordinate systems) of different parts to each other in various ways. A direct connection between frames of two components represent a rigid connection between the two. Instead of fixed connections, different joints connections are used. Joints typically enables relative motion between the two parts in one or more degrees of freedom and restricts it in the other ones. Rigid transform blocks in between a connection specifies that the frames are rotated and/or translated in some way with respect to each other.
Figure 3.1: SIMSCAPE visualization of the assembly. During simulation the motion of the parts are animated.

Again, in Figure 3.2 one can see the carriage part has one of its reference frames connected to the world frame. This makes the carriage rigidly connected to the environment. Hence it cannot move and will act as the reference part of the assembly. Next is the pinion part connected to the carriage through a rack-and-pinion constraint. This constraint will cause the two components to move in sync such that the rotation of the pinion will correspond to a translational motion. The pinion is then connected to the boom through an actuated revolute joint. This enables the pinion to rotate with respect to the boom. Also present is a prismatic joint enabling the boom and pinion to traverse the carriage subject to the rack-and-pinion constraint. The forks are then connected to the boom through another actuated revolute joint enabling the forks to rotate with respect to the boom. Finally rigidly connected to the forks is a rectangular solid block with the dimensions of a standard pallet representing the load on the forks. The mass of this block may vary.

Several input and output ports of the mechanical system have been added enabling interaction with the other systems.
3.2 Hydraulic System

The hydraulic system schematic shown in Figure 2.2 has been modelled using blocks from SIMSCAPE FLUIDS. Most of the components in the schematic are standard hydraulic components available as off-the-shelf blocks in SIMSCAPE FLUIDS. This essentially means that the schematic has just been mirrored into SIMSCAPE as can be seen in the model in Figure 3.3.

Most of the components have just a few parameters that are either directly available in datasheets or easily estimated. A few components have a more complex parameter setting requiring a lot more work to extract from datasheets. These are explained in the following sections.

3.2.1 Proportional valves

The proportional valves are modelled using the standard 4-way directional valve in SIMSCAPE FLUIDS. In order to capture the typical characteristics of the real valve, the current-flow curve from the manufacturer is replicated. Since every valve is unique, this curve is then adjusted in the identification process described below in Section 3.4.1. The characteristics for one valve was previously shown in
Figure 3.3: **SIMSCAPE** model of the hydraulic system. Only the pump and the actuators are visible due to confidentiality reasons.

Figure 2.4 and a fitted curve is shown in Figure 3.4.

Figure 3.4: Valve characteristics from the simulation model. Note that the simulated curve is slightly modified from the one in Figure 2.4 to correspond to the characteristics of the valve used in the parameter optimization. As in Fig. 2.4 the right curve corresponds to a increasing current and the left curve to a decreasing current, ideally.
3.2.2 Hydraulic motor

Figure 3.5: A diagram showing the characteristics of the hydraulic motor. From this it is possible to extract the flow and torque at a given motor speed and pressure difference. The grey area shows the recommended operating region. Taken from the manufacturer data sheet.

The hydraulic motor is modelled using the block "Fixed-Displacement Motor" with the displacement \( D_m = 98\text{cm}^3\text{/rev} \). In order to make the model more accurate, the characteristics diagram from the manufacturer is used, see Figure 3.5. In this diagram, several points are evaluated in terms of hydromechanical and volumetric efficiency. These are defined as

\[
\eta_{hm} = \frac{\Delta p D_m}{M}, \quad \eta_{vol} = \frac{Q}{nD_m}
\]  

(3.1)

where \( M \) is the motor torque, \( n \) is the angular velocity, \( \Delta p \) is the pressure drop and \( Q \) is the flow through the hydraulic motor. These two efficiency grids are used as the model for the motor and can be seen in Figure 3.6.

3.2.3 Pump

The pump is modelled with the block "Fixed-Displacement Pump" and a constant velocity source. Because of the ideal velocity source the mechanical efficiency which describes the mechanical power usage can be neglected. Hence, only the volumetric efficiency is of importance. The expression for deriving the volumetric efficiency is the same as for the motor. A measurement on a used hydraulic pump serves as reference, see Figure 3.7. The fitted volumetric efficiencies can be seen in Figure 3.8.
Figure 3.6: The volumetric and mechanical efficiencies for the motor extracted from Figure 3.5 using (3.1).
Figure 3.7: The pump efficiencies from the manufacturer datasheet for 1450 rpm. Efficiencies for other speeds are obtained from company internal measurements.

Figure 3.8: The volumetric efficiency model for the pump. The values are calculated using Figure 3.7, company internal measurements and Equation 3.1.
3.3 Electrical System

The electrical system has been highly simplified in the model. One could, in theory, quite easily model the actual PWM-voltage cycle and the way this creates a current in the spool controlling the valve. However, since the timescale of the electrical system is several orders of magnitude faster than those of the mechanical and hydraulic systems (the reason why PWM "works"), detailed modelling of it would only slow down and complicate the simulation without improving accuracy. Hence, this is instead modelled using mean-values.

The current in one of the spools is directly proportional to the duty cycle through Equation 3.2 where $I_j$ is the current in spool $j$, $DUTY_j$ is the duty-cycle ($\in [0, 1]$) in spool $j$, $V_S$ the supply voltage and $R_{SPOOL}$, the resistances of the spool.

$$I_j = \frac{DUTY_j \cdot V_S}{R_{SPOOL}}$$

3.4 Model Parameters

Many of the parameters in the model are directly available in manufacturer data-sheets, internal company drawings and schematics or are easily measurable on the actual forklift. Others are not directly available and it would be very tedious to construct a test to measure them.

Generally all parameters that are considered "important" in the sense that their values are of great importance to the behaviour and performance of the model are directly available. Typical examples are pump and motor displacement, cylinder size and valve characteristics.

In contrast to this, several different parameters do not affect the overall performance of the simulation as long as their values are "reasonable". Examples are valve time constants, valve leakage area, pipe critical Reynolds number etc. In theory, a test could be constructed to measure most of these variables but that would be very time consuming while providing little to no model improvement. Instead, the values of all of these variables are estimated based on the authors and company employees engineering knowledge as well as research of similar products with more detailed datasheets.

3.4.1 Parameter optimization

Different types of parameters acquired from datasheets and drawings have different accuracy due to manufacturing tolerances, wear and tear and so on which might cause the simulation response to a certain input to differ from the actual forklift response to the same input. By using system identification methods, these parameters can be adjusted in order to make the model response more accurate. In MATLAB/SIMULINK, there are tools which uses system identification theory to help find values of selected parameters. In this work, the toolbox "SIMULINK Design Optimization" is used. The tool simulates the model response to a certain
input and adjusts the parameters in order to match the measured output as close as possible using a least-squares method.

An important parameter which is identified with this method is the “Initial openings” between ports in the proportional valves. These indicate how much the spool needs to be displaced before the oil starts to flow, which makes it a way to adjust the valve deadzone. In the tables below "Initial opening PA" are for flow from port P to A, and so on.

**Rotation**

By using the control lever, the forks were rotated in sine wave like manner with varying speed and the relevant signals were logged. During the subsequent optimization, the parameters in Table 3.1 were then allowed to vary within the given range.

**Table 3.1: Rotational parameter optimization. Nominal values and tolerances.**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Nominal value</th>
<th>Tolerance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valve hysteresis</td>
<td>35 mA</td>
<td>±10 mA</td>
</tr>
<tr>
<td>Joint friction</td>
<td>50 Nm/(deg/s)</td>
<td>±50 Nm/(deg/s)</td>
</tr>
<tr>
<td>Cog radius</td>
<td>56 mm</td>
<td>±5 mm</td>
</tr>
<tr>
<td>Initial opening PA</td>
<td>-3 mm</td>
<td>±1 mm</td>
</tr>
<tr>
<td>Initial opening PB</td>
<td>-3 mm</td>
<td>±1 mm</td>
</tr>
<tr>
<td>Initial opening AT</td>
<td>3 mm</td>
<td>±1 mm</td>
</tr>
<tr>
<td>Initial opening BT</td>
<td>3 mm</td>
<td>±1 mm</td>
</tr>
</tbody>
</table>

Figure 3.9 shows the measured system response and the simulated response after optimization. Clearly the curves match quite well, especially during the first 10 seconds. Table 3.2 shows the optimized parameter values.

**Table 3.2: Rotational parameters after optimization.**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valve hysteresis</td>
<td>44 mA</td>
</tr>
<tr>
<td>Joint friction</td>
<td>41 Nm/(deg/s)</td>
</tr>
<tr>
<td>Cog radius</td>
<td>59 mm</td>
</tr>
<tr>
<td>Initial opening PA</td>
<td>-3.0 mm</td>
</tr>
<tr>
<td>Initial opening PB</td>
<td>-2.7 mm</td>
</tr>
<tr>
<td>Initial opening AT</td>
<td>3.1 mm</td>
</tr>
<tr>
<td>Initial opening BT</td>
<td>3.6 mm</td>
</tr>
</tbody>
</table>

**Traverse**

As with the rotating motion, the control lever for the traverse motion was operated in a sine wave like manner and relevant signals were logged. During the
optimization the parameters in Table 3.3 was optimized within the range given in the same table.

Table 3.3: Traverse parameter optimization. Nominal values and tolerances.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Nominal value</th>
<th>Tolerance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valve hysteresis</td>
<td>30 mA</td>
<td>± 10 mA</td>
</tr>
<tr>
<td>Joint friction</td>
<td>250 N/(m/s)</td>
<td>± 200 N/(m/s)</td>
</tr>
<tr>
<td>Cog radius</td>
<td>45 mm</td>
<td>± 5 mm</td>
</tr>
<tr>
<td>Initial opening PA</td>
<td>-3 mm</td>
<td>± 1 mm</td>
</tr>
<tr>
<td>Initial opening PB</td>
<td>-3 mm</td>
<td>± 1 mm</td>
</tr>
<tr>
<td>Initial opening AT</td>
<td>3 mm</td>
<td>± 1 mm</td>
</tr>
<tr>
<td>Initial opening BT</td>
<td>3 mm</td>
<td>± 1 mm</td>
</tr>
</tbody>
</table>

Figure 3.10 shows the measured system response and the simulated response after optimization. Clearly the curves match quite well. Table 3.4 shows the parameter values after optimization.
3.4 Model Parameters

Figure 3.10: Simulated and measured traverse response (velocity) after parameter optimization. The measured velocity has been obtained offline by differentiating the position followed by zero-phase filtering.

Table 3.4: Traverse parameters after optimization.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valve hysteresis</td>
<td>36 mA</td>
</tr>
<tr>
<td>Joint friction</td>
<td>202 N/(m/s)</td>
</tr>
<tr>
<td>Cog radius</td>
<td>44 mm</td>
</tr>
<tr>
<td>Initial opening PA</td>
<td>-3.0 mm</td>
</tr>
<tr>
<td>Initial opening PB</td>
<td>-2.8 mm</td>
</tr>
<tr>
<td>Initial opening AT</td>
<td>2.8 mm</td>
</tr>
<tr>
<td>Initial opening BT</td>
<td>3.3 mm</td>
</tr>
</tbody>
</table>

3.4.2 Validation

In order to validate the model with the parameters found a new test is used. In this test the rotation and traverse are operated at the same time.

Figure 3.11a and 3.11b show the simulated and measured response of the rotation and traverse respectively.
(a) Traverse and rotational velocity. Note that mechanical hard-stops are included in the model causing the spikes and sharp drops seen especially in the simulated rotational velocity.

(b) Traverse and rotational position. Essentially just the integral of the figure above.

**Figure 3.11:** Simulated and measured motions in the validation experiments.
This chapter deals with the design and implementation of the control system on the forklift.

4.1 Design

The traverse and rotational motions are controlled independently by two separate but structure-wise identical controllers. The overall design ideas are described in the following sections.

4.1.1 Requirements

Two basic controller requirements exist:

1. The controller must be able to track a position trajectory with sufficient accuracy. This is needed during auto-rotation.

2. The controller must be able to hold a reference velocity with sufficient steady state accuracy. This is needed during all normal operation except for auto-rotation.

The first requirement essentially implies requirement two since tracking a linearly increasing position trajectory would require holding a constant velocity.

4.1.2 Reference signal

With the basic requirements outlined in Section 4.1.1 in mind the question as to what reference signal to use arises. Since position tracking is required a reference position is definitely required. However, since velocity tracking is also required it would seem natural to also use reference velocity.
There is also the problem as to what measurements that are available. A position measurement is available for both motions but no velocity measurement. In the traverse case the measurement is done by an encoder but in the rotation case it is by an analog sensor. Without going into details it can be stated it is a lot easier to get a good velocity measurement by differentiating the position measurement from an encoder than it is from an analog sensor. In the analog case, heavy low-pass filtering would be required introducing a substantial phase shift (time delay) severely limiting the controller performance.

Another option would be to use an observer to estimate the velocities. Examining Equation 4.14 (the mechanical model), one can note that these equations alone are quite complex. However, in order to construct an observer one would have to know $F$ and $\tau$ as a function of the control signals, i.e. the equations of the hydraulic system. One quickly realises that this model would turn out to be highly complex and nonlinear. Given the limited computing power in the forklift the idea of using an observer is ruled out to be unfeasible.

A possible work around is to use velocity as the reference signal and integrate it to give a reference position to be used in the feedback loop. Hence to track a position trajectory the corresponding velocity trajectory is supplied to the controller. In this way the reference velocity can be used in feedforward and integrated velocity in feedback.

### 4.1.3 Feedforward

In a constant velocity state the velocity and flow are directly proportional to each other through the geometric properties of the hydromechanical converters (the cylinders and the motor). This trivial relationship is used to calculate a feedforward control signal given the reference velocity.

Disregarding any leakage in the components the oil flow through the proportional valves will equal the oil flow entering the hydromechanical converters, respectively. The flow $q_r$ entering the rotation cylinders equals the piston velocity $v_p$ times the piston area $A_p$ as

$$q_r = v_p A_p. \tag{4.1}$$

Similarly the flow $q_t$ through the traversing motor ideally equals its displacement $D_m$ times the rotational velocity $\omega_t$ as

$$q_t = \omega_t D_m. \tag{4.2}$$

As stated before, the translational motion of the cylinders are converted to a rotational motion through a cog which the forks are connected to. Hence, given a reference rotational velocity $\omega_{ref}$ the required flow for rotation can be calculated according to

$$q_r = R_r \omega_{ref} A_p \tag{4.3}$$

where $R_r$ denotes the cog radius of the rotating motion. Similarly, the rotational velocity of the hydraulic motor is converted to a translational motion through a rack and pinion construction. Hence, given a reference translational velocity $v_{ref}$...
the required flow for traversing can be calculated as

\[ q_t = \frac{v_{\text{ref}}}{R_p} D_m \]  

(4.4)

where \( R_p \) denotes the pinion radius. With the required flow one can use the inverse of the corresponding current-flow curve described in Section 3.2.1 to find the required current. Figure 4.1 shows the inverse of the traverse valve curve. Please observe that the deadband in the curve is removed.

\[ \text{Figure 4.1: Current as a function of flow without deadband for the traverse valve.} \]

Following the required current obtained from the inverse the deadband current \( I_{\text{opening}} \) is added. In general the deadband current corresponding to the rightmost curve in Figure 3.4 (the opening curve) is added. However, in order to compensate for the hysteresis when rapid closing of the valve is required the deadband current is lowered with the width of the hysteresis, \( I_{\text{hysteresis}} \). This is done when the derivative of the absolute value of the reference velocity falls below a certain threshold value. Hence, the feedback will take care of hysteresis compensation as long as rapid closing is not required, such as when releasing control levers or approaching mechanical end-positions.

To summarize the feedforward part of the rotational control signal is given in Equation 4.5 where \( f_R(\cdot) \) denotes the current-flow inverse and \( g(\cdot) \) is given in Equation 4.7. Similarly, the feedforward part of the traverse control signal is given in equation 4.6.

\[ u_{ff,\text{rot}} = f_R(R_r \omega_{\text{ref}} A_p) + g(\omega_{\text{ref}}) \]  

(4.5)

\[ u_{ff,\text{trav}} = f_T\left(\frac{v_{\text{ref}} D_p}{R_p}\right) + g(v_{\text{ref}}) \]  

(4.6)
\[ g(x) = \begin{cases} 
I_{\text{opening}} - I_{\text{hysteresis}}, & \text{if } \frac{\partial}{\partial t}|x| < a_{\text{thres}} \\
I_{\text{opening}}, & \text{otherwise}
\end{cases} \tag{4.7} \]

**Remark:** The implementation of the derivative in \( g(x) \) will be discrete and hence the case of \( x = 0 \) will not cause trouble.

### 4.1.4 Feedback

In addition to the feedforward compensation feedback from measured position is implemented. Since the reference is a velocity but the measurement is a position the velocity reference is integrated to give a reference position to be compared to measured position. Equation 4.8 shows how the reference velocity is integrated to give reference position. Furthermore, \( v_{\text{ref}} \) denotes either traverse or rotational velocity and \( x_{\text{ref}} \) denotes traverse or rotational position depending on context.

\[ x_{\text{ref}}(t) = \int_0^t v_{\text{ref}}(\tau) d\tau + x(0) \tag{4.8} \]

The reference position is compared to the measured position and the error is fed into a simple PID-controller. Hence the feedback control signal is given according to

\[
u_{\text{fb}}(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \dot{e}(t)
\]

\[ e(t) = x_{\text{ref}}(t) - x(t) \tag{4.9} \]

where \( K_p \), \( K_i \) and \( K_d \) are the controller gains.

### 4.1.5 Controller summary

The total control signal is the sum of the feedforward and the feedback terms. The feedforward and feedback terms are requested current through the solenoids. Current through the solenoids is controlled by the PWM-duty cycles. Actual current through the solenoids depend not only on the duty cycle but also the supply voltage from battery as well as the resistance in the solenoid circuit, both of which are time-varying.

Hence, the requested current from the controllers is multiplied by the nominal circuit resistance \( R_{\text{nom}} \) and divided by the measured supply voltage \( U_s \) in order to find the required duty cycle according to equation 4.10.

\[
u_{\text{duty}}(t) = \left[ u_{ff}(t) + u_{fb}(t) \right] \frac{R_{\text{nom}}}{U_s(t)} \tag{4.10} \]

The overall controller structure is illustrated in the block diagram in Figure 4.2.
4.2 Optimal Control

In order to find the optimal path for an auto-rotation, optimal control theory is employed. The system is expressed with its fundamental motion equation which are used to perform a nonlinear optimization of the motion. The objective function is set to minimize the jerk of the motion, which is defined as the derivative of the acceleration. This strategy is chosen because it is previously proven that a jerk-minimizing strategy will reduce oscillations in the system and create a smooth trajectory [3].

4.2.1 Derivation of equations of motion

The system can be seen as a force and torque actuated pendulum on a cart with gravitational forces neglected, see Figure 4.3. The state vector $q = [x, \phi]^T$ and the external force vector $u = [F, \tau]^T$ are introduced. By using Lagrangian mechanics, the equation of motion is then calculated as
\[
\frac{d}{dt} \left[ \frac{\partial}{\partial \dot{q}} \mathcal{L} \right] - \frac{\partial}{\partial q} \mathcal{L} = u
\]  
(4.11)

where \( \mathcal{L} = \mathcal{K} - \mathcal{V} \) is the Lagrangian in which \( \mathcal{K} \) and \( \mathcal{V} \) represents kinetic and potential energy, respectively. Since the motion is in the horizontal plane, the potential energy is equal to zero. The kinetic energy for the system is

\[
\frac{1}{2} M \ddot{x}^2 + m \left( \frac{1}{2} \dot{x}^2 + \frac{1}{2} l^2 \dot{\phi}^2 + l \cos(\phi) \dot{\phi} \right) + \frac{1}{2} \theta_G \dot{\phi}^2
\]  
(4.12)

which yields that Equation 4.11 can be written as

\[
M \ddot{x} + m \ddot{x} + ml \cos(\phi) \ddot{\phi} - ml \sin(\phi) \dot{\phi}^2 = F
\]  
\[
ml^2 \ddot{\phi} + ml \cos(\phi) \ddot{x} + \theta_G \ddot{\phi} = \tau.
\]  
(4.13)

By introducing the new states \( v := \dot{x}, \omega := \dot{\phi} \) in the state vector \( \mathbf{x} = [x, v, \phi, \omega, F, \tau]^T \) and \( \Delta F := \dot{F} \) and \( \Delta \tau := \dot{\tau} \) as control signals, the equations in 4.13 can be written in state-space form as

\[
\dot{x} = v
\]
\[
\dot{v} = \frac{F \theta_G + Fl^2 m + l^3 m^2 \omega \sin(\phi) - lm \tau \cos(\phi) + l m \theta_G \omega \sin(\phi)}{(M + m) \theta_G + l^2 m^2 + l^2 M m - l^2 m^2 \cos^2(\phi)}
\]
\[
\dot{\phi} = \omega
\]
\[
\dot{\omega} = \frac{-\omega \cos(\phi) \sin(\phi) l^2 m^2 - F \cos(\phi) l m + (M + m) \tau}{(M + m) \theta_G + l^2 m^2 + l^2 M m - l^2 m^2 \cos^2(\phi)}
\]
\[
\dot{F} = \Delta F
\]
\[
\dot{\tau} = \Delta \tau.
\]  
(4.14)

With the system in state-space form, the next task is to formulate the optimization problem.

### 4.2.2 Minimum jerk trajectory motivation

The trajectory generated from the optimization should create a movement as smooth as possible. In [7] and as previously mentioned, [3], this is discussed. The general opinion from these sources is that a jerk-minimizing optimization creates the most "natural" trajectory which also is easier for a controller to track. In this context, natural means that it resembles how a human would have controlled the motion in order to make it as smooth as possible.

Since the controls are expressed as the derivative of the force and torque which are proportional to the acceleration, the controls can be interpreted as the jerk.
4.2.3 Optimal control problem

As explained in the previous section, the optimal path should minimize the jerk of the movement. Because of this, the objective function is chosen as the square of the control, as in [7]. Also, the path has geometric constraints due to the limited space between the aisles which is illustrated in Figure 4.3. Another important constraint for the problem is the maximum speed allowed for the current load and lift height which must be taken into account. This maximum speed, and also the maximum reaction forces are given as a requirement from the mechanical engineering department at the company. By introducing $Q = \text{diag}([Q_1, Q_2])$ as penalties for the control inputs, the optimal control problem is now formulated as:

Figure 4.3: The geometry of the optimal control problem. The red dotted lines represent the geometric constraints.
Given Equation 4.14, find the minimizing control signal \( U(t) = (\Delta F(t), \Delta \tau(t))^T \) of the performance measure

\[
J(U(t)) = \int_0^T U(t)^T Q U(t) dt
\]

subject to the path constraints

\[
x + (h + d) \sin(\varphi) + \frac{b_1}{2} \cos(\varphi) \leq L + \delta \tag{4.16}
\]
\[
-x - (h + d) \sin(\varphi) + \frac{b_1}{2} \cos(\varphi) \leq \delta \tag{4.17}
\]
\[
x + d \sin(\varphi) + \frac{b_2}{2} \cos(\varphi) \leq L + \delta \tag{4.18}
\]
\[
-x - d \sin(\varphi) + \frac{b_2}{2} \cos(\varphi) \leq \delta, \tag{4.19}
\]

the initial and final state

\[
x(0) = (0, 0, \pi/2, 0, 0, 0)^T \tag{4.20}
\]
\[
x(T) = (L, 0, -\pi/2, 0, 0, 0)^T, \tag{4.21}
\]

and the force and velocity constraints

\[
|F| \leq F_{\text{max}} \tag{4.22}
\]
\[
|\tau| \leq \tau_{\text{max}} \tag{4.23}
\]
\[
|v| \leq v_{\text{max}} \tag{4.24}
\]
\[
|\omega| \leq \omega_{\text{max}}. \tag{4.25}
\]

Since the core problem is to move the forks as quickly as possible, a time-optimal strategy would in theory be more appropriate. This approach however lead to major numerical issues while performing the optimization and was therefore discarded. Instead, the final time \( T \) is lowered until infeasibility of the optimization is reached.

The problem is not analytically solvable by classic optimal control due to the heavily nonlinear motion equations. However, there exists a variety of numerical solvers which can deal with this kind of problem. In this work, the MATLAB optimal control software ACADO [9] is used. For details of how to use ACADO in MATLAB, please consult [2].

### 4.2.4 Remarks about the geometric constraints

The limit \( \delta \) above has to be chosen carefully since a too small value leads to infeasibility. The feasible geometric region for the position states \( x \) and \( \varphi \) with the numerical values in Table 4.1 can be seen in Figure 4.4. In this figure, an optimized path is also plotted. However, if \( \delta \) is chosen too big, the motion will be unnecessarily "uncentered" since the optimization will choose the shortest path,
4.2 Optimal Control

Figure 4.4: The allowed geometric region in the \((x, \varphi)\)-space, that is the constraints (4.16) to (4.19) for three different \(\delta\). The other numerical values used to generate the plot can be seen in Table 4.1. The limit \(\delta\) should be chosen s.t. the allowed corridor gets as narrow as possible in order to maximize the minimum distance to the pallet rack. In this case, \(\delta = 0.15\) is a good choice.

Table 4.1: Numerical values used to generate the path seen in Figure 4.4. Masses and inertia are calculated in CAD-software.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>(L)</td>
<td>1.35</td>
<td>m</td>
</tr>
<tr>
<td>(b_1)</td>
<td>0.8</td>
<td>m</td>
</tr>
<tr>
<td>(b_2)</td>
<td>1.2</td>
<td>m</td>
</tr>
<tr>
<td>(h)</td>
<td>1.2</td>
<td>m</td>
</tr>
<tr>
<td>(d)</td>
<td>0.09</td>
<td>m</td>
</tr>
<tr>
<td>(m)</td>
<td>950</td>
<td>kg</td>
</tr>
<tr>
<td>(\theta)</td>
<td>191</td>
<td>kgm^2</td>
</tr>
<tr>
<td>(l)</td>
<td>0.692</td>
<td>m</td>
</tr>
<tr>
<td>(M)</td>
<td>151</td>
<td>kg</td>
</tr>
</tbody>
</table>

and not the path where the largest margin to the pallet racks is maintained. This leads to a "maximin-problem" where the minimum distance to the pallet rack should be maximized. A simple solution to this problem is to adjust \(\delta\) to a value which makes the allowed corridor, seen in Figure 4.4, as narrow as possible. In this way, the optimized path will be as centered as possible with respect to the aisle.
In this chapter, the results from the work are presented. This includes the optimized trajectories obtained using optimal control. The control system described in Section 4.1 is implemented in the existing hardware using automatic code generation. This enables quantitative analysis of the system and gives the developers a qualitative feel of the motions.

5.1 Optimized Trajectories for Auto-rotation

The optimal control problem described in 4.2 is solved for three different weights (0, 750 and 1500 kg) which defines the speed limits $v_{\text{max}}$ and $\omega_{\text{max}}$. The resulting trajectories can be seen in Figure 5.1 (please note the time scaling in the figure) and the limits are found in Table 5.1. The force and torque limits are never reached, but are included for completeness. As mentioned before, the terminal time $T$ is chosen as the lowest time which leads to a feasible solution.

Since these trajectories are generated using rough approximations they will most likely not be optimal, but optimized. However, they do keep the forks as far away as possible from the pallet rack. Without a structural approach like this, it would otherwise be hard to come up with a good trajectory without violating the geometric and/or velocity limits of the system.
Figure 5.1: Solutions for the optimal control problem for three different loading conditions. The parameters used for each optimization can be seen in Table 5.1 Note that the time scales are normalized over time for easier comparison and that the x-axis corresponds to the discretization grid (50 points in all cases).

Table 5.1: Parameters used for the optimizations in Figure 5.1. Limits on speeds, force and torque are mechanical constraints set by the mechanical engineers at the company.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>0 kg</th>
<th>750 kg</th>
<th>1500 kg</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>s</td>
<td>6.3</td>
<td>9.8</td>
<td>12.5</td>
</tr>
<tr>
<td>$v_{max}$</td>
<td>mm/s</td>
<td>350</td>
<td>270</td>
<td>210</td>
</tr>
<tr>
<td>$\omega_{max}$</td>
<td>deg/s</td>
<td>40</td>
<td>30</td>
<td>20</td>
</tr>
<tr>
<td>$F_{max}$</td>
<td>N</td>
<td>440</td>
<td>440</td>
<td>440</td>
</tr>
<tr>
<td>$\tau_{max}$</td>
<td>Nm</td>
<td>250</td>
<td>250</td>
<td>250</td>
</tr>
</tbody>
</table>
5.2 Controller Testing

The controllers are tested by performing various manoeuvres and supervising the tracking performance. Each of the controllers are tested individually by using each function isolated. Finally, the total system performance is measured by several auto-rotation manoeuvres where both function are used simultaneously.

Note that in all cases except Section 5.3 the position measurements are from the internal sensors, i.e. they have not been made using an independent external measurement system with high accuracy. However, the test in Section 5.3 showed that the internal sensors are in fact very accurate.

5.2.1 Traversing controller

The traversing motion is tested by traversing to both sides and measuring the tracking performance in both speed and position.

![Traversing motion position tracking](image1)

![Traversing motion velocity tracking](image2)

![Traversing position error](image3)

(a) Traversing motion position tracking  (b) Traversing motion velocity tracking  (c) Traversing position error

Figure 5.2: Traverse controller performance.

5.2.2 Rotation controller

The rotating motion is tested similarly as the traversing motion by rotating to both end positions and measuring the tracking performance. Figure 5.3a shows the position tracking performance and Figure 5.3b shows the velocity tracking. The error of the velocity tracking can be seen in Figure 5.3c.
5.2.3 Auto-rotation to 180°

Figure 5.4, 5.5, 5.6 and 5.7 show the tracking performance of the control system during auto-rotation with 0, 500, 1000 and 1500 kg on the forks respectively. From the figures it can be seen that the tracking performance is the best with no load on the forks and the performance degrades as the load increases. In particular the rotation tracking performance degrades considerably and an oscillatory behaviour is induced as the load is increased. The traverse tracking performance is similar regardless of the load on the forks.
5.2 Controller Testing

Figure 5.4: Auto-rotation (180°) tracking performance with no load on the forks. Velocity has been obtained offline.

Figure 5.5: Auto-rotation (180°) tracking performance with 500 kg load on the forks. Velocity has been obtained offline.
Figure 5.6: Auto-rotation (180°) tracking performance with 1000 kg load on the forks. Velocity has been obtained offline.

Figure 5.7: Auto-rotation (180°) tracking performance with 1500 kg load on the forks. Velocity has been obtained offline.
5.2.4 Auto-rotation to 90°

In order to validate the accuracy and robustness of the traverse and rotation controllers a series of tests of the auto-rotation to 90° were done. These were done using four different load cases using each load case twice. This test series was also carried out on two different forklift units with quite different characteristics. Unit one is an older version of the forklift and hence it has suffered from a lot of "wear and tear". Unit two is a much newer version of the forklift and hence runs much "smoother" than unit one. Most notable is that the friction in all the joints is higher in the older version of the forklift.

To measure the traverse position an independent laser measurement system with an accuracy of ±0.1 mm is used. The rotation angle is measured externally using the setup depicted in Figure 5.8.

![Figure 5.8: Rotation position measurement setup. By measuring the two distances the angle offset from 90° can be estimated using $\theta_{error} = \arcsin\left(\frac{|d_{left} - d_{right}|}{b}\right)$](image)

Tables 5.2, 5.3 and 5.4 show the results of the test series. It can be seen that the error of the traverse and rotation is within the requirements (10 mm and 9 mrad) in all cases. The traverse error seems uncorrelated with load weight but the rotation error increases as the load increases. In general forklift unit two (the newer one) performs better than unit one in both traverse and rotation accuracy. The small bias in traverse position can be explained by bad calibration of the traversing encoder.
Table 5.2: Traverse positioning result for the two different forklift units during auto-rotation to 90°.

<table>
<thead>
<tr>
<th>Load [kg]</th>
<th>Pos. 1 [mm]</th>
<th>Error 1 [mm]</th>
<th>Pos. 2 [mm]</th>
<th>Error 2 [mm]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>668.6</td>
<td>3.6</td>
<td>673.1</td>
<td>1.9</td>
</tr>
<tr>
<td>0</td>
<td>668.8</td>
<td>3.8</td>
<td>672.6</td>
<td>2.4</td>
</tr>
<tr>
<td>500</td>
<td>670.0</td>
<td>5.0</td>
<td>672.0</td>
<td>3.0</td>
</tr>
<tr>
<td>500</td>
<td>669.1</td>
<td>5.9</td>
<td>673.6</td>
<td>1.4</td>
</tr>
<tr>
<td>1000</td>
<td>671.1</td>
<td>4.9</td>
<td>674.5</td>
<td>0.5</td>
</tr>
<tr>
<td>1000</td>
<td>670.8</td>
<td>5.2</td>
<td>673.3</td>
<td>1.7</td>
</tr>
<tr>
<td>1500</td>
<td>671.5</td>
<td>4.5</td>
<td>675.3</td>
<td>0.3</td>
</tr>
<tr>
<td>1500</td>
<td>672.4</td>
<td>2.6</td>
<td>675.8</td>
<td>0.8</td>
</tr>
</tbody>
</table>

Table 5.3: Rotation positioning results for forklift unit one during auto-rotation to 90°. The error is calculated using the method seen in Figure 5.8 with \( b = 1200 \) mm.

<table>
<thead>
<tr>
<th>Load [kg]</th>
<th>( d_{left} ) [mm]</th>
<th>( d_{right} ) [mm]</th>
<th>( \theta_{error} ) [mrad]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>928</td>
<td>925</td>
<td>3</td>
</tr>
<tr>
<td>0</td>
<td>928</td>
<td>926</td>
<td>2</td>
</tr>
<tr>
<td>500</td>
<td>927</td>
<td>928</td>
<td>1</td>
</tr>
<tr>
<td>500</td>
<td>927</td>
<td>928</td>
<td>1</td>
</tr>
<tr>
<td>1000</td>
<td>925</td>
<td>928</td>
<td>3</td>
</tr>
<tr>
<td>1000</td>
<td>926</td>
<td>928</td>
<td>2</td>
</tr>
<tr>
<td>1500</td>
<td>923</td>
<td>931</td>
<td>7</td>
</tr>
<tr>
<td>1500</td>
<td>923</td>
<td>930</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 5.4: Rotation positioning results for forklift unit two during auto-rotation to 90°. The error is calculated using the method seen in Figure 5.8 with \( b = 800 \) mm.

<table>
<thead>
<tr>
<th>Load [kg]</th>
<th>( d_{left} ) [mm]</th>
<th>( d_{right} ) [mm]</th>
<th>( \theta_{error} ) [mrad]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1002</td>
<td>1002</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1003</td>
<td>1001</td>
<td>3</td>
</tr>
<tr>
<td>500</td>
<td>1002</td>
<td>1002</td>
<td>0</td>
</tr>
<tr>
<td>500</td>
<td>1002</td>
<td>1002</td>
<td>0</td>
</tr>
<tr>
<td>1000</td>
<td>1001</td>
<td>1002</td>
<td>1</td>
</tr>
<tr>
<td>1000</td>
<td>1000</td>
<td>1002</td>
<td>3</td>
</tr>
<tr>
<td>1500</td>
<td>999</td>
<td>1003</td>
<td>5</td>
</tr>
<tr>
<td>1500</td>
<td>999</td>
<td>1004</td>
<td>6</td>
</tr>
</tbody>
</table>

5.3 Validation of Traversing Accuracy

In order to validate that the traverse control system is capable of tracking a sequence of target positions, typically a requirement in order to develop an automatic version of the forklift, a test program is implemented in the forklift. The
test cycle is as follows:

1. Start in left home position
2. Traverse right 600mm
3. Traverse left 30mm
4. Traverse right 10mm three times
5. Traverse back to left home position

This test cycle is performed five times for three different loads (0 kg, 500 kg, 1500 kg) to ensure repeatability and proper functionality in different working conditions. An external laser measurement system is used to monitor the traversing position with an accuracy of $\pm 0.1$ mm. The results can be seen in Figure 5.9. It is clear that the accuracy requirement is met, even though some offset exist. This is probably due to an incorrect internal zero-positioning which will not affect the accuracy of a change in position.
Figure 5.9: The traversing motion test performed for three different weights, five times each. The red dashed line represent the set point. It can clearly be seen that the accuracy requirement of ±10 mm is fulfilled.
The problem formulation in Section 1.3 was to investigate if it is possible to control the turret head to achieve a traversing accuracy of ±10 mm and a rotational accuracy of ±0.5°. As shown in Chapter 5 this is possible without adding new hardware to the system.

6.1 Model Accuracy and Complexity

The model validation responses did not match as well as they did in the estimation plots but it was considered to be good enough to use for control development purposes. This can partly be motivated with the big differences between forklifts. A perfect model would only validate the system against a single forklift but the control software must work on all machines.

The resulting model turned out to be quite complex. SIMSCAPE reports that the model after simplification has 121 variables of which 53 are differential and 68 are algebraic. Differential variables are variables whose time derivative appears in the system equations. Hence, they add dynamics to the model. In contrast, algebraic variables are variables whose time derivative does not appear in the system equations. They do not add dynamics and are typically due to mass and energy conservation laws.

The complexity of the resulting model is something one should keep in mind when using modelling tools such as SIMSCAPE. The modelling job is quite easy as blocks for many physical things are readily available. You do not need to "care" about the equations, just drag and drop the block and connect it to the model. However, complexity quickly grows as it is tempting to add every little detail to the model.

It is possible that similar model performance could be achieved with a much simpler model but this was not investigated since the model could simulate quite
fast anyway. Closed-loop simulations were on average performed at about one to three times slower than real-time when using a modern laptop.

### 6.2 Control System

The closed loop control system developed in this thesis performs well, but in many cases similar to the existing open loop control system. In manual control or when the motions is unsynchronized, open loop control performs better than the proposed closed loop configuration in terms of general handling quality. This is probably due to the delays in the hydraulic system which tend to cause oscillations if the lever commands are not heavily filtered.

When performing auto-rotation, the existing system features a semi-closed loop control system, but without feedforward part. This is probably the main reason why the auto-rotation error is decreased with the new control system.

Worth noting is that the control system has not gone through a thorough test and verification procedure. It was proven to work well in the test cases performed in the results section but in order to implement it in a production environment further verification has to be done. A few examples: How will the system perform on a different forklift? How will it perform during continuous operation in different environments? What kind of parameter variations can it handle?

### 6.3 Model-Based Design

The concept of MBD has been used throughout the development of the Fork Control System presented in this thesis. The design process started with the development of a model based on drawings, schematics, datasheets and measurements of the forklift. The input-output behaviour of the model was then compared to that of the forklift and changes were made as necessary.

When the model was found to work sufficiently well the development of the control system started. The control system was developed incrementally and as new features were implemented the whole system was continuously simulated using the model previously developed (“Model-in-the-Loop”) in order to verify intended performance and behaviour. Using this approach bugs and logical errors in the software were continuously discovered, identified and fixed on a very short timescale.

At a certain point the control system was good enough to be tested on the real forklift. Automatic code generation from the control system developed in MATLAB, SIMULINK and STATEFLOW was used to output C-code implementable on the microcontroller of the forklift. The generated code was then integrated into the existing code-base of the microcontroller, essentially just replacing a certain function controlling the fork movements.

As the system was tested on the real forklift several new bugs and errors were continuously discovered. In all of these cases the bugs were reproducible in simulation. The cause was then quickly identified, fixed and verified, often within
minutes of discovery. New code could then be generated and implemented on
the forklift removing the bug.

From the authors point of view the concept of MBD shows many advantages
over more traditional system development methods. Especially the agility of the
method vastly outperforms traditional methods. The time from an idea of a new
feature to design, implementation, test, verification and deployment is often or-
ders of magnitudes less than what could be achieved traditionally. Worth men-
tioning is that the end-result, i.e. the quality of finished system developed using
MBD might not perform better than a system developed traditionally. However
the time, resources and ultimately money needed to produce equivalent results
will in many cases be much less using MBD.

6.4 Future Work

The main conclusion of the work is that the system is hard to tame, even with a
fairly sophisticated control system. Because of this, it is probably wise to look for
different mechanical designs with servo-class hydraulic valves or even electrome-
chanical actuators. This would probably decrease the engineering effort for the
design of a control system and make the system more reliable.


