Aalborg University

Active safety in collaborative robots

Electronic & IT:
Control & Automation

Group:
CA10-1031

Student report

June 7, 2018

Fourth year of MSc study
Electronic and IT
Fredrik Bajers Vej 7
DK-9220 Aalborg East, Denmark
http://www.es.aau.dk

AALBORG UNIVERSITY
STUDENT REPORT

Topic:
Active safety in collaborative robots

Project:
P10-project

Project time:
February 2018 - June 2018

Projectgroup:
CA10-1031

Participants:
Simon Bjerre Krogh

Supervisors:
Henrik Schi∅ler

Synopsis:

This report covers the derivation of a collision detection scheme based on a dynamic model of a robotic structure with three degrees of freedom. First, a PID controller is derived with the purpose of controlling the system, then an extended Kalman filter with the purpose of collision detection. The extended Kalman filter did not work and was replaced with a linear stationary Kalman filter which is capable of estimating the system states. The collision detection is based on external force estimator, which is made by an open loop approach.
The open loop approach was deemed insufficient for collision detection, as it showed a high correlation between force estimations and inputs. Due to this correlation, it was not possible to detect if a collision had occurred.
In the end a discussion reflecting the problems of the project and how further research may solve these are made.

Number of pages: 89
Appendix: 30
Completed June 7, 2018

Preface

This report has been created by Simon Bjerre Krogh. The project has been made as part of the fourth semester of the Control & Automation master program at Aalborg University.

The report is intended for people with background knowledge corresponding to fourth semester at Control & Automation, Aalborg University.

Aalborg University, 7^{th} of June 2018

Simon Bjerre Krogh
skrogh13@student.aau.dk

Nomenclature

Acronyms

HRC	Human-Robot Collaboration
SEAs	Series Elastic Actuators
VSEAs	Variable Series Elastic Actuators
FD	Fault Detection
CDI	Collision Detection and Isolation
FDI	Fault Detection and Isolation
EKF	Extended Kalman Filter

Glossary of mathematical notation

This section sums up the mathematical notation and terminology used in this report.

Time derivative

$$\begin{aligned} \frac{d}{dt}Q &= \dot{Q} \\ \frac{d}{dt}\dot{Q} &= \ddot{Q} \end{aligned} \tag{1}$$

Where ($\dot{}$) is used for the first time derivative of the function Q, ($\ddot{}$) the second time derivative etc.

Estimations and error Estimations are denoted with a hat $\hat{q}$ and error with a tilde $\tilde{q}$.

Cosine and sinus subscripts

$$\begin{aligned} sin(\theta_1) &= s_1 \quad \wedge \quad cos(\theta_1) = c_1 \\ sin(\theta_1 + \theta_2) &= s_{(1,2)} \quad \wedge \quad cos(\theta_1 + \theta_2) = c_{(1,2)} \end{aligned} \tag{2}$$

where the subscript defines the angle of references.

Matrix and vector

Matrices are denoted with upper case letters e.g. Q, where vectors are denoted with lower case letters e.g. q.

Symbols

Symbol	Description	Unit
c	Cosine	·
m	Mass	kg
s	Sinus	·
$\mathcal{L}$	Lagrangian	$\frac{kg \cdot m^2}{s^2}$
Q	Joint angle	rad
T	Kinetic energy	$\frac{kg \cdot m^2}{s^2}$
U	Potential energy	$\frac{kg \cdot m^2}{s^2}$
q	Joint angle	rad
$\dot{q}$	Joint velocity	$\frac{rad}{s}$
$\ddot{q}$	Joint acceleration	$\frac{rad}{s^2}$
P	Joint position is Cartesian space	·
$\dot{P}$	Joint velocity is Cartesian space	·
l_i	Link length i	m
t_i	Kinetic energy contribution of link i	J
ρ	Link length density	$\frac{kg}{l_i}$
g	Gravity	$\frac{m}{s^2}$
τ	Generalized torque	Nm
B	Inertia Matrix	·
C	Centrifugal and Coriolis matrix	·
F_s	Coulomb friction	$\frac{Nm}{s}$
F_v	Viscous friction	$\frac{Nm}{s}$
G	Gravity matrix	·
f_e	External forces	N
J	Jacobian of end-effector	·
$sign$	Signum function	·
Θ	State vector	·
f	Model function	·
Y	Measured output	·
h	Output function	·
V	Lyapunov function	·
K_P	Proportional gain	·
K_I	Integral gain	·
K_D	Derivative gain	·
Ts	Step size	s
Q	System covariance	·
R	Sensor noise covariance	·
L	Kalman gain	·

Contents

1 Introduction

In 1960 robots where introduced in the industry for handling heavy and dangerous tasks for humans. In 1980 the technology was expanded to the assembly line of handling small part assembly and material handling[1]. Since then, the interest of robots in the industry, see *Figure 1.1*, has continued to grow and statistics show that the annual average sale of robots has increased with 84% from 2005 to 2016 [2].

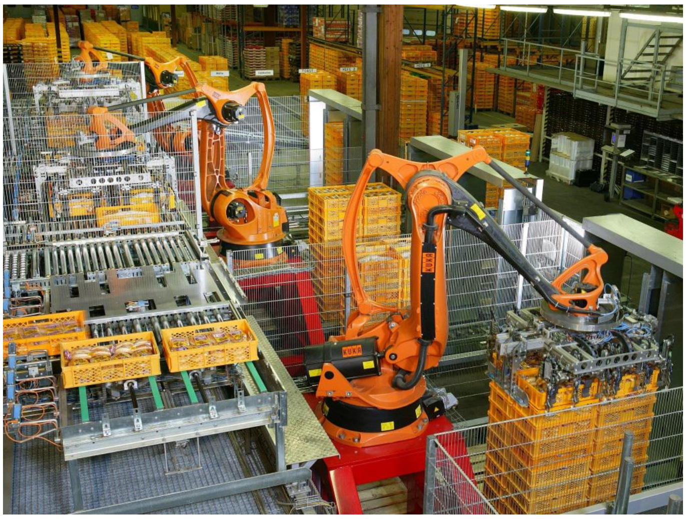

Figure 1.1: Automation with the use of robots for palletizing food products[3]. Note the large safety fences surrounding each robot that prevent contact between humans and robots.

There are several reasons for replacing human workers with robots of which some are listed below:[4]

- Robots can increase the production rate due to increased performance.
- In hazardous environments robots are better suited as they are expendable in contrast to human life.
- Human strength can be a limiting factor, e.g., lifting several hundred kilogrammes is nearly impossible for humans.

The advantages of robots in the assembly line, lies in the fact that they can be highly superior to human workers, especially when it comes to strength, endurance and precision. These advantages gives the opportunity to increase the productivity, which usually have economic benefit for the company.

Even though robots can be superior to humans, tasks requiring manual dexterity or alternating tasks which require high flexibility is not necessarily beneficial to be handled by robots due to implementation challenges[4, 5]. However a combination of both humans and robots in collaboration, enables the opportunity of combining the advantages of both parties. This field is referred to as Human-robot collaboration (HRC)[6].

HRC builds on humans and robots collaborating in a close physical proximity to each other, thereby unifying the workspace between the two instead of replacing one with the other, see *Figure 1.2*. HRC utilizes the flexibility and judgement of human workers, which can be hard to implement, combined with the strength and endurance of robots, which humans often lack. This leads to a more flexible and efficient production line as the two parties complements the abilities of each other[5, 7].

Figure 1.2: Human-robot collaboration: the human is controlling the movement of an object, while the robot is doing the heavy lifting[8].

A primary issue in HRC is to ensure human safety as the workspace between humans and robots are shared with a following higher risk of injuries due to collision between the two. In the simple case shown on *Figure 1.1*, collaboration between humans and robots is not required. Safety is therefore implemented using a large fence around the robots. The fence minimizes the risk of a human physically intersecting the workspace of a robot and thus minimizes the risk of a dangerous situation that could lead to injuries or death. Unfortunately, this is not a possible solution when it comes to HRC, as humans and robots have to collaborate in a shared workspace, see *Figure 1.2*. Consequently, more complex solutions have to be designed.

1.1 Safety: Human-robot collaboration

As mentioned, HRC requires a more sophisticated safety strategy to ensure human safety. Due to the increased risk of collision between human and robot, the different phases pre-impact, impact, and post impact have to be taken into consideration[9].

1.1.1 Pre-impact phase

The primary focus for this phase is to minimize the risk of the robot colliding with the environment, i.e. collision avoidance. Collision avoidance requires knowledge of the current environment to determine a safe motion with low risk of collision. The environment can be divided into two groups when HRC is required, one covering the static parts and another covering the moving parts i.e. humans. Taking these into consideration when designing the collision avoidance scheme can increase the protection of human workers and the robotic structure. Regardless of the importance of both group of environmental factors, human safety is obviously the main priority.

When humans are a part of the environment, an intuitive and simple solution is to stop the movement of the robot temporarily when humans are within close proximity of the robot. A *light curtain* is one possible solution for detecting humans within close proximity, see *Figure 1.3*.

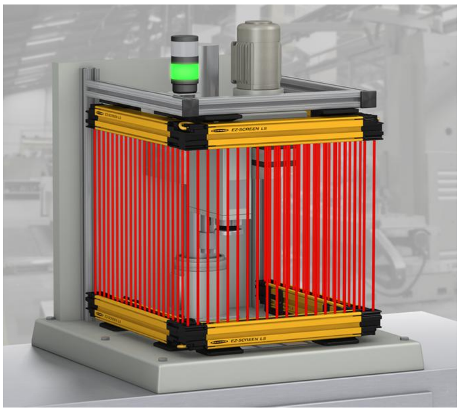

Figure 1.3: Illustration of light curtain usage. If the light beams (red lines) are crossed by any object, the robot can respond immediately[10].

These are opto-electronic devices that works as safety guards by the use of light. They work by an emitter sending out a modulated beam of light and a receiver that receives the light. If an object intersects the light beam and the receiver is cut off, a detection of interference is said to be present and the movement of the robot can be stopped until the interference disappears.
The benefits to this approach is that fast and safe interaction with the robot can be made without having to physically push a switch off button as this is fully automated.

Another approach is to make the robot aware of its surroundings, and from that determine a safe motion that avoids collision while preserving a high performance, see *Figure 1.4*.

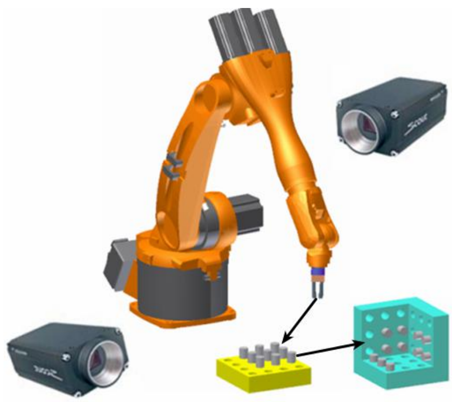

Figure 1.4: Illustration of a case where robot vision is used to detect the surroundings. Two cameras are placed around a robot structure. The task is then to get the gray bricks from the yellow plate to the green plate, without colliding with the walls of the green one [11]

Information of the surroundings and how to react on it, can be handled by *robot vision*[12].

Robot vision works by one or more vision sensors capturing images of the surroundings. The images are then analysed for the surrounding obstacles' position, with respect to the robotic structure. The optimal path can then be determined from the obstacles positions in the surroundings and the task of the robot[13, 14].

1.1.2 Impact phase

If the collision avoidance fails and collision is taking place, the impact of collision with the robot should be minimized.

The impact can minimized by decreasing the inertia and kinetic energy[15]. This can be achieved by pursuing a light weight of the robotic structure[15]. Heavy actuators and link material can be replaced with lighter products. Actuators can be placed at the base of the robot and robotic links can instead be cable driven.
Another approach is to limit the velocity of the robot, thereby limiting the amount of kinetic energy. Such changes result in a lower impact force, at the cost of a more expensive design and sacrificing some of the performance as the robot is restricted from moving at full velocity.

Using actuators which increase the compliance behavior of the robot, such as *series elastic actuators* (SEAs), can improve the shock absorption under collision as some of the energy is stored in the actuator[16]. This leads to a smaller direct energy transfer between robot and environment, thus less damaging effect. The SEAs work by including a spring between the output of the actuator and the robotic link, which limits the stiffness of the robot to the elastic coupling[17], see *Figure 1.5*.

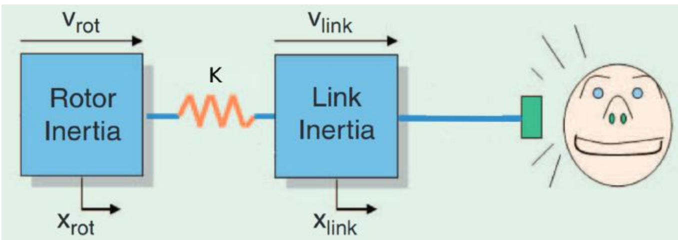

Figure 1.5: Block diagram of the spring placement between actuator and robotic link. The arrows at the top and bottom represent the velocity direction and displacement of the position respectively. At the right side a human is colliding with the robot.[17]

Because stiffness of the robot influence the rate of precision, *variable* SEAs (VSEAs) may be considered[7]. VSEAs enable the opportunity to variate the stiffness of the robotic structure, see *Figure 1.6*. The stiffness could therefore be chosen after the velocity of the robotic movement, i.e. high velocity, low stiffness and vice versa[18].

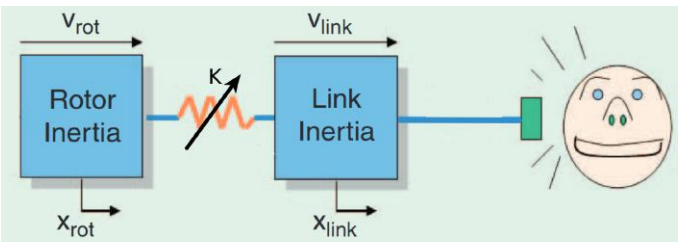

Figure 1.6: Block diagram of the variable spring placement between actuator and robotic link. The arrows at the top and bottom represent the velocity direction and displacement of the position respectively. At the right side a human is colliding with the robot.[17]

1.1.3 Post impact phase

The main goal of the post impact phase is to chose a correct reaction strategy for the robot after a collision has occurred. A crucial part of this phase is therefore to detect when a collision have happened i.e. collision detection, after which a suitable reaction strategy can be selected.

Detection

Collision detection can be done by adding external sensors to the robotic structure, such as; sensitive skin, strain gauges, or force load cells[19, 20]. In many cases it is, however, most cost-efficient to detect the collision without using additional sensors. In addition, such solution provides the opportunity of implementing it on already installed robots that did not include sensors for collision detection.

To detect a collision without additional sensors, one approach is to compare the current consumption of the system to the estimated current consumption of the model and look for inconsistencies between the two[21, 22]. This approach is however a difficult scheme to utilize due to varying command torque dynamics. Furthermore, online torque computations are based on inverse dynamics that require acceleration measurements, which are highly receptive to noise[23].

Another approach is to see a collision as a faulty behavior of the robots actuating system. A collision will affect the current draw to the actuators, since a higher torque is required, thus a *fault detection* (FD) scheme can be used on the actuators, with the purpose of collision detection. The benefits of FD is that is does not require acceleration measurements or inversions of the inertia matrix[23]. Approaching an FD scheme for collision detection generates a vector of residuals to determine if a collision has occurred or not. Ideally, this vector should be non-zero in the event of collision and zero otherwise, however due to imperfection in, e.g., modelling this is not achievable and statistical thresholds are set for residuals. Therefore, if the residual is exceeding the predefined thresholds, a collision is said to be present.

Collision scenarios

Before any accommodation strategies can be designed, it is necessary to understand the different collision scenarios that could occur for the specific robot and its environment.

In *Figure 1.7*, four different collision scenarios are presented, and a small explanation to these are given below. It should be noted that *Figure 1.7* does not contain any sharp or blunt objects and an analysis upon such scenarios are not made.

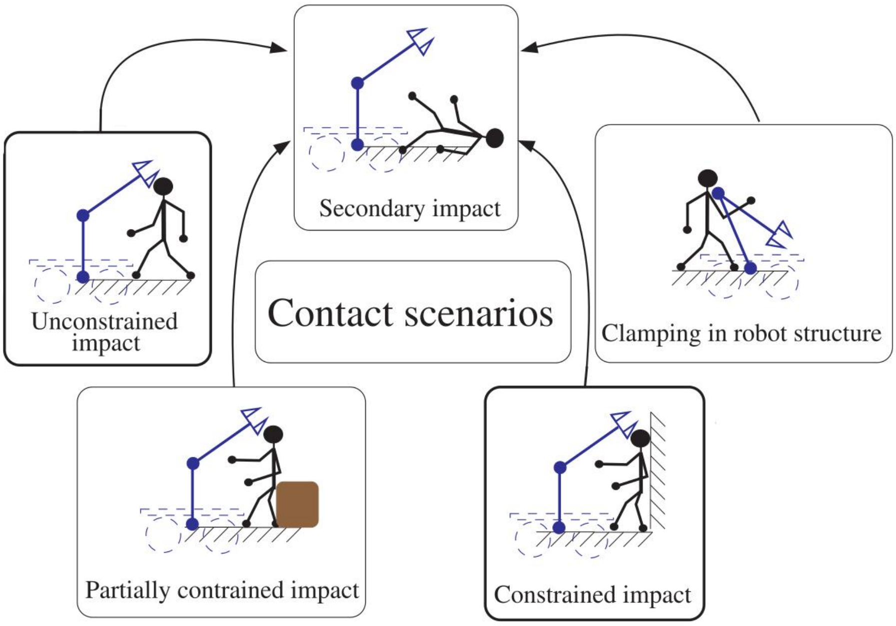

Figure 1.7: Collision scenarios between human and robot[24].

Unconstrained: If a collision between human and robot is happens in an unconstrained environment, the result will be the human getting punched/pushed once by one direct hit and no additional force is applied by the robot afterwards.

Constrained: Here the human is limited by the surroundings. A collision between robot and human can result in a jammed situation, where the human can be exposed to additional forces generated by the robot.

Partially constrained: Here, some parts of the human body is limited in movement by the surroundings. In the case seeing in *Figure 1.7*, the robot may push the human in the direction of the box with the result of the human falling. If the collision was directed at the lower parts of the human body, the result will be the same as a constrained situation.

Clamping in robot structure: This is a slightly changed case of a constrained situation. Here the human is stuck in between the links of the robotic structure, where the force of impact can increase over time.

Secondary impact: After a collision, secondary impact covers what happens afterwards e.g. the human falls, secondary collision, is being pinched, etc.

Accommodation

After a collision is detected it is important to chose a correct reaction strategy to ensure protection of the environment (viz. humans) and the robotic structure if possible. To do so it is necessary to isolate where the collision have happened. From Section *1.1.3: Detection* two types of detection schemes are briefly described; *collision* and *fault* detection. Including isolation to these, result in *collision detection and isolation* (CDI) or *fault detection and isolation* (FDI) scheme. Having one of these schemes implemented on the robot, should enable the opportunity of accommodating the collision in the most suitable way, thus switching to another control strategy.

An intuitive strategy after collision is to stop the movement of the robot. However, this approach is not necessarily the most suitable option. In Section *1.1.3: Collision* *scenarios* four different scenarios of collision is described. In the case of an unconstrained collision, stopping the movement of the robot ensures that no further force is added to the environment. Unfortunately, this also increases the risk of a clamped situation which can be dangerous, especially if the detection of collision is delayed and a large force is applied.

If it is possible to determine the direction of collision, a reaction scheme could be to force the robot in an opposite direction of the impact, thus the clamped situation could be alleviated. However if the force in the opposite direction is too large or uncontrolled, the possibility of it colliding with another part of the environment is increasing.
If the direction is known, a control strategy could be to switch to a controller with compliance. In doing so, the robot should respond as if it were a spring being compressed when interacting with the environment. In a clamped situation, it should be possible to move the robotic structure when force is applied opposite to the robot.

1.2 Project focus

The overall focus of this project is on human-robot collaboration with a perspective on human safety. To limit the project extent the main goal is on deriving a collision detection and isolation scheme with the ability to detect when a collision happens and the ability of isolating the location of collision. A secondary task is on how to accommodate the collision, such that the impact will be as small as possible.

Summarizing, a problem formulation can be written as;

How to actively detect, isolate and accommodate a collision of a robot, in a human-robot collaborative environment without violating human safety?

System description 2

This chapter provides an analysis of the system available at Aalborg university, Denmark. The analysis includes a description of the physical structure of the robot, specifications related to the different components, and its interface.

2.1 Physical structure

The robot available for this project is a uStepper, shown in *Figure 2.1*. The uStepper is a robotic arm that is comprised of four rotational joints of which three can be controlled by actuators and the fourth is passive, i.e., only manually movable. At the current state, the end-effector is fixed to a horizontal position with respect to ground.
The actuatable joints are numerated as joint one to three, see *Figure 2.1*.

The connection between joint one and two is made with a direct gear transmission, whereas the connections between joints two and three are made with parallelogram linkages.

Figure 2.1: The uStepper robotic arm. Each actuatable joint is marked with numbers from 1 to 3 [25].

The main specification of the uStepper robot can be seen in *Table: 2.1*

Total weight	2.87kg
Material	PLA plastic and aluminum tubes
No. of axes	3
Gear ratios	Main gear: 4.09:1 Second gear: 4.09:1 Base gear: 2.09:1

Table 2.1: Main specifications for the uStepper arm.

2.1.1 Actuator

Three *LM42 NEMA 17 - Lexium MDrive* motor solutions are implemented For actuation of the uStepper. Each motor solution contains a stepper motor with matching motor controller that is interfaced by CAN bus, see *Figure 2.2*. The motor controller uses an internal encoder that gives this solution a closed loop performance regarding position and force control.

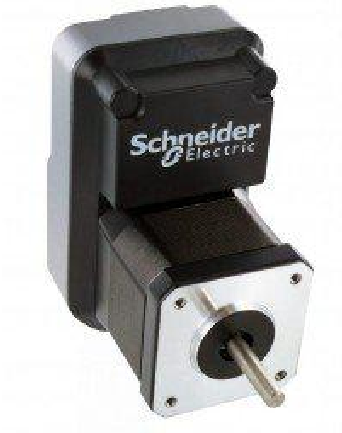

Figure 2.2: The LM42 NEMA 17 - Lexium MDrive[26].

The advantages of this solution is that a high precision control can be achieved and information of position and torque can be retrieved. Further, stepper motors yields an increased performance when it comes to start/stop and reverse response[27]. The latter advantage is likely a beneficial factor when it comes to the mitigation of a collision.

More technical information regarding the Lexium MDrive can be found in [28].

2.1.2 Micro controller

To communicate with the Lexium MDrive a *Teensy 3.6* micro-controller is used, see *Figure 2.3*.

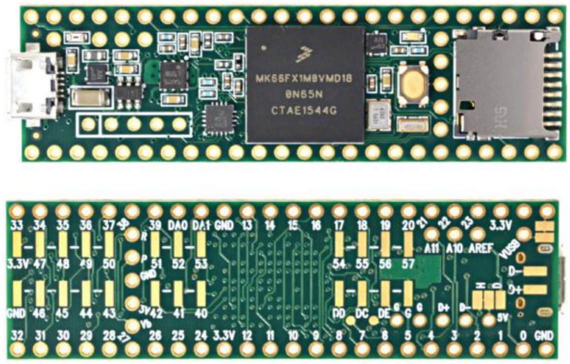

Figure 2.3: Top and bottom view of the Teensy micro-controller.

The Teensy is a micro-controller with a 180 MHz, 32 bit, ARM Cortex-M4 processor. The micro-controller has 1024 kbytes Flash, 256 kbytes RAM, and 4096 bytes EEPROM. This micro-controller has two USB hosts. One host is high speed (480 Mbit/s) and one host is full speed (12 Mbit/s). The full speed host is used for serial communication.
The micro-controller supports communication over CAN bus, thus making it compatible with the Lexium MDrivers. In addition, it is possible to use the Arduino CAN library through the add-on Teensyduino. As such, the CAN bus protocol is easily implemented.

3 Project limitations and requirements

Human-robot collaboration and its concerns regarding safety is a major problem to be solved and can be separated into three categories:

1. Collision detection,
2. Isolation of the collision,
3. Collision accommodation.

Each of these problems requires comprehensive analysis to determine which strategy is the most efficient to the specific system.
Since this project is limited in recourses, the project extent will be limited to only solve the first problem concerning the detection of a collision.

The system available has three degrees of freedom where a collision can occur anywhere on the physical structure. The collision coverage is limited to only be for the tool center point i.e collision can only occur with the tool center point. Furthermore, the human part which collides with the robot is assumed to be a hand.

Requirements

The requirements for this project is based on the ISO standard [ISO 15066:2016] *Robots and robotic devices, Collaborative robots*[29]. This standard covers the aspects of human safety in a HRC environment, its considerations, and the restrictions of the robotic system.

It is assumed that collision can only happen between the tool center point and the hands of a human. In the ISO standard, it is specified, that the maximum allowed transfer of energy between a robot and human hands is 0.49 Joules. To ensure this the standard suggests limiting the velocity of the system such that the kinetic energy is not exceeding the requirement. This approach, however, decreases the performance of the system, due to the limiting of the velocity.
Therefore, the following proposal to the requirement is made, which instead is based on the detection time of the collision. The requirement derivation is based on a general form, for which the requirement for this project is stated at the end of this section.

The kinetic energy of a system is giving in *Equation: (3.1)*, where the speed of the energy transaction between human and robot is found. The energy transfer is found by taking the time derivative of the kinetic energy, which is shown in *Equation: (3.2)*.

$$E = \frac{1}{2}mv^2 \tag{3.1}$$

Where

	E	is the kinetic energy,	[J]
	m	is the mass of the system,	[kg]
and	v	is the velocity of the system.	$\left[\frac{\mathrm{m}}{\mathrm{s}}\right]$

$$\dot{E} = m\dot{v}v \tag{3.2}$$

Where

$\dot{v}$	is the acceleration of the system,	$\left[\frac{\mathrm{m}}{\mathrm{s}^2}\right]$

Since force can be denoted as mass times acceleration, *Equation: (3.2)* is rewritten to *Equation: (3.3)*, where $F = m\dot{v}$ and denote the force.

$$\dot{E} = Fv \tag{3.3}$$

The time derivative of the kinetic energy is then set as an inequality constraint with the derivative of the maximum allowed energy transfer, see *Equation: (3.4)*. Here the energy transferred to a human is denoted as E_{req}.

$$\dot{E} < \dot{E}_{req} \tag{3.4}$$

Substituting *Equation: (3.2)* into *Equation: (3.4)* and solving for the time derivative leads to *Equation: (3.5)*.

$$\Delta t < \frac{E_{req}}{F \cdot v} \tag{3.5}$$

Where

Δt	specifies the collision detection time,	[s]

In *Equation: (3.5)* the inequality constraint for the detection time is shown.
It can be seen that the detection time is dependent on the amount of force the system is capable of delivering and the velocity of it.

For this project, the parameters, F and v, is found through the datasheet and the software-based velocity limit of the motors. In this project, the motor velocity limit is set to 5.72 rad/s.
The highest velocity possible for the available system is made by extending the robotic arm and actuating joint one see *Figure 2.1* for the joint position. With the gearing for joint one, this leads to a total velocity of 1.23 m/s at the end-effector.
The force is based on the maximum torque the motors can deliver. Which is found to be 0.31 N/m. This equals to a force at the end-effector of 1.44 N.

With the found parameters, the detection time inequality can be found to be

$$\Delta t < \frac{0.49}{1.44 \cdot 1.23} = 0.27 \ \ [\mathrm{s}] \tag{3.6}$$

If the detection time is meeting the minimum requirement of $\Delta t = 0.27$ it would require that the accommodation of collision is instantaneous, which is impossible. Therefore to make the accommodation possible the detection time is further limited to be below 100 milliseconds.

Dynamic derivation 4

A Lagrangian approach, see [30, p. 247], has been chosen for modelling the robotic dynamic of the uStepper described in Section *2.1: Physical structure*. This approach has the benefits of modelling the dynamics in a compact analytical form, where the inertia, centrifugal, Coriolis and gravitational forces are included. In this section the derivation of the Lagrangian dynamics is made.

4.1 Lagrangian

To describe the dynamics of the system, the Lagrangian has to be found. This is defined as seen in *Equation: (4.1)*. As can be seen it describes the difference between kinetic and potential energy.

$$\mathcal{L}(q,\dot{q}) = T(q,\dot{q}) - U(q) \tag{4.1}$$

Where

	$\mathcal{L}$	is the Lagrangian,	$\left[\frac{\text{kg}\cdot\text{m}^2}{\text{s}^2}\right]$
	q	is the joint angle,	$[\text{rad}]$
	$\dot{q}$	is the joint velocity	$\left[\frac{\text{rad}}{\text{s}}\right]$
	T	is the kinetic energy,	$\left[\frac{\text{kg}\cdot\text{m}^2}{\text{s}^2}\right]$
and	U	is the potential energy.	$\left[\frac{\text{kg}\cdot\text{m}^2}{\text{s}^2}\right]$

In *Figure 4.1* a simple 2D schematic representation of the uStepper is shown.

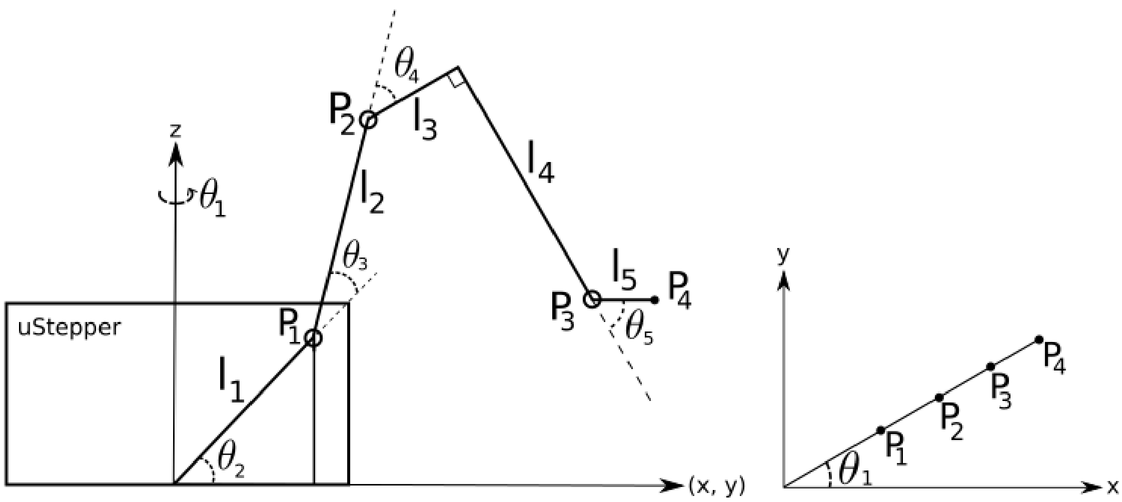

Figure 4.1: An illustration of the link position with respect to the global workspace of the robot. The leftmost figure shows a cross sectional side view of the robot and the rightmost figure shows the cross sectional view from above. Position P_4 is the center of the end-effector.

By placing the origin of the global coordinate system at the center of the uStepper, the forward kinematics of the joints positions, P_i, can be determined using *Equation: (4.2)*. In *Equation: (4.2)*, the position is described using a vector of $[x, y, z]^T$ coordinates. To keep the equations short, c is short for cos, s is short for sin, and the subscript number denotes the angle number in *Figure 4.1*. For instance, c_1 denotes the cosine of the angle θ_1 in *Figure 4.1*. Multiple subscripts are used as shorthand for addition of angles. For instance, $c_{2,3}$ is the cosine of the angle $\theta_2 + \theta_3$.

$$
\begin{aligned}
P_1(q) &= l_1 \cdot \begin{pmatrix} c_1c_2 \\ s_1c_2 \\ s_2 \end{pmatrix}, \quad P_2(q) = P_1 + l_2 \cdot \begin{pmatrix} c_1c_{(2,3)} \\ s_1c_{(2,3)} \\ s_{(2,3)} \end{pmatrix} \\
P_3(q) &= P_2 + l_3 \cdot \begin{pmatrix} c_1c_{(2,3,4)} \\ s_1c_{(2,3,4)} \\ s_{(234)} \end{pmatrix} + l_4 \cdot \begin{pmatrix} c_1c_{(2,3,4-\frac{\pi}{2})} \\ s_1c_{(2,3,4-\frac{\pi}{2})} \\ s_{(2,3,4-\frac{\pi}{2})} \end{pmatrix} \\
P_4(q) &= P_3 + l_5 \cdot \begin{pmatrix} c_1c_{(2,3,4,5-\frac{\pi}{2})} \\ s_1c_{(2,3,4,5-\frac{\pi}{2})} \\ s_{(2,3,4,5-\frac{\pi}{2})} \end{pmatrix}
\end{aligned} \tag{4.2}
$$

Where

	l_i	is the length of link i,
and	P_i	is the position of joint i.

Note that θ_2 is constant and cannot be changed without redesigning the physical structure of the robot. The same is true for the right angle between links l_3 and l_4. The equation for the position of P_4 can be simplified somewhat because the position of the end-effector is always parallel to x,y plane (ground plane). That is, the angles θ_2 to θ_5 adds up to zero, thus, P_4 can be rewritten as seen in *Equation: (4.3)*.

$$
P_4 = P_3 + l_5 \cdot \begin{pmatrix} c_1c_{(2,3,4,5-\frac{\pi}{2})} \\ s_1c_{(2,3,4,5-\frac{\pi}{2})} \\ s_{(2,3,4,5-\frac{\pi}{2})} \end{pmatrix} \implies P_3 + l_5 \cdot \begin{pmatrix} c_1 \\ s_1 \\ 0 \end{pmatrix} \tag{4.3}
$$

Kinetic energy

The kinetic energy contribution of each link of the uStepper can be found as an integration of velocities over the link. This is shown in *Equation: (4.4)*. The derivation of this can be found in *Appendix: C*.

$$
\begin{aligned}
t_i &= \rho_i \frac{1}{2} \cdot \int_0^{L_i} [\dot{P}_{i-1} + \frac{1}{L_i}(\dot{P}_i - \dot{P}_{i-1}) \cdot l]^T [\dot{P}_{i-1} + \frac{1}{L_i}(\dot{P}_i - \dot{P}_{i-1}) \cdot l] \mathrm{d}l \\
&= m_i \frac{1}{6} \left(\dot{P}_{i-1}^T \dot{P}_{i-1} + \dot{P}_i^T \dot{P}_i + \dot{P}_{i-1}^T \dot{P}_i \right)
\end{aligned} \tag{4.4}
$$

Where

	t_i	is the kinetic energy contribution of link i,	$\left[\frac{\mathrm{kg \cdot m^2}}{\mathrm{s^2}}\right]$
	ρ_i	is the link density of link i	$\left[\frac{\mathrm{kg}}{\mathrm{L}}\right]$
	m_i	is the mass of link i,	[kg]
	L_i	is the length of link i,	[m]
	l	is the integration length,	[m]
and	$\dot{P}_i$	is the velocity of joint position i	$\left[\frac{\mathrm{m}}{\mathrm{s}}\right]$

Potential energy

Similar to the kinetic energy, the potential energy contributed by each link can be described as the integration of potential energy over the individual link. This is shown in *Equation: (4.5)*.

$$
\begin{aligned}
p_i &= \rho g \int_0^L \left[P_{i-1} + l \frac{1}{L_i} [P_i - P_{i-1}] \right]^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \mathrm{d}l \\
&= mg \frac{1}{2} \left[P_{i-1} + P_i \right]^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}
\end{aligned}
\tag{4.5}
$$

Where

g	is the gravity.	$\left[\frac{\mathrm{m}}{\mathrm{s}^2}\right]$

Since the potential energy is only affected by the height difference between the link position and the base frame of the uStepper, the link position with respect to the z axis of the base frame is of interest. Therefore, the vector $[0, 0, 1]^T$ is included in *Equation: (4.5)*.

The total amount of kinetic and potential energy stored in the uStepper is described by *Equation: (4.6)*.

$$
\begin{aligned}
T &= \sum_{i=1}^{n} t_i \\
U &= \sum_{i=1}^{n} u_i
\end{aligned}
\tag{4.6}
$$

Now the Lagrangian is described, the next step is to find the equation of motion.

4.2 Equation of motion

The Lagrangian defined in Section *4.1: Lagrangian*, is used to describe the energy stored in the uStepper manipulator. Describing the dynamic equation, or equation of motion, the Lagrangian is used in *Equation: (4.7)*.

$$
\tau = \frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial \mathcal{L}}{\partial \dot{q}} - \frac{\partial \mathcal{L}}{\partial q} \tag{4.7}
$$

Where

τ	is the generalized force acting on q.	$[\mathrm{N} \cdot \mathrm{m}]$

The result of taking the derivatives of the Lagrangian, as shown in *Equation: (4.7)*, is shown in *Equation: (4.8)*.

$$
\tau = B(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) \tag{4.8}
$$

Where

	B	is a $[n \times n]$ positive and symmetric matrix containing the inertia,
	C	is a $[n \times n]$ matrix containing the centrifugal and coriolis forces
and	G	is a $[n \times 1]$ matrix containing the gravitational forces.

The matrices B, C and G, can be found in the attached files, [Dir: Attachments/Model].

In *Equation: (4.9)*, friction and external forces are added to the dynamic equation. This is done since mechanical systems are rarely frictionless and in the case of HRC external forces are affecting the system.

$$\tau - J^T(q)f_e = B(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F_s \cdot sign(\dot{q}) + F_v\dot{q} \tag{4.9}$$

Where

	J	is the Jacobian for the end-effector,
	f_e	is the [1xn] vector of external forces affecting the end-effector,
	F_s	is the diagonal [nxn] matrix of coulomb friction,
and	F_v	is the diagonal [nxn] matrix of viscous friction.

4.3 State space representation

To simulate and test the system model, *Equation: (4.9)* is written into a general state space notation as shown in *Equation: (4.10)*

$$\begin{aligned} \dot{\Theta} &= f(\Theta(t), \tau(t), f_e(t)) \\ Y &= h(\Theta) \end{aligned} \tag{4.10}$$

Where

	f	is the system model,
	Θ	is the state vector,
	Y	is the measured outputs,
and	h	is the output function.

Isolating *Equation: (4.9)* for $\ddot{Q}$, *Equation: (4.11)* can be defined.

$$\ddot{q} = B(q)^{-1}[\tau - J^T(q)f_e - C(q,\dot{q})\dot{q} - G(q) - F_v\dot{q} - F_s sign(\dot{q})] \tag{4.11}$$

Denoting the state vector as Θ=$[q \;\; \dot{q}]^T$, the state space system can be described as *Equation: (4.12)*.

$$\dot{\Theta} = \begin{bmatrix} 0_{3x3} & I_3 \\ 0_{3x3} & -B^{-1}(C+F_v) \end{bmatrix} \Theta + \begin{bmatrix} 0_{3x3} \\ B^{-1} \end{bmatrix} \tau - \begin{bmatrix} 0_{3x1} \\ B^{-1}(G + F_s sign(\dot{q})) \end{bmatrix} - \begin{bmatrix} 0_{3x3} \\ B^{-1}J^T \end{bmatrix} f_e \tag{4.12}$$

As a result of *Equation: (4.12)*, the state space can be written in the general form as seen in *Equation: (4.10)*.

4.4 Model validation

In this section, the model described in Section *4.3: State space representation* is validated. The data used herein and how it is gathered is described in *Appendix: A*. The link lengths and weights can be found in *Appendix: B* and the following simulations can be found in [Dir: Attachments/Simulation].
The simulations are made by the ODE45 solver in Matlab, which requires that the model is continuously differentiable. As the model in *Equation: (4.12)* is not continuous due to the $sign(\dot{q})$ function, an approximation of this is made by $tanh(\dot{q})$.

The validation of the model is performed using two approaches. The first test is without friction components and is performed to see if the model reacts in a similar manner as the physical system. The second validation contains hand tuned friction components, to fit the model response to the system response. The model is simulated such that only one joint is allowed to move at a time and therefore require three simulations for each validation step.

4.4.1 First validation step

In the first validation step, the friction components, external forces and actuator inputs in *Equation: (4.12)* are discarded. The simulations are shown in a chronological order from the actuatable joints one to three, see *Figure 4.2*.

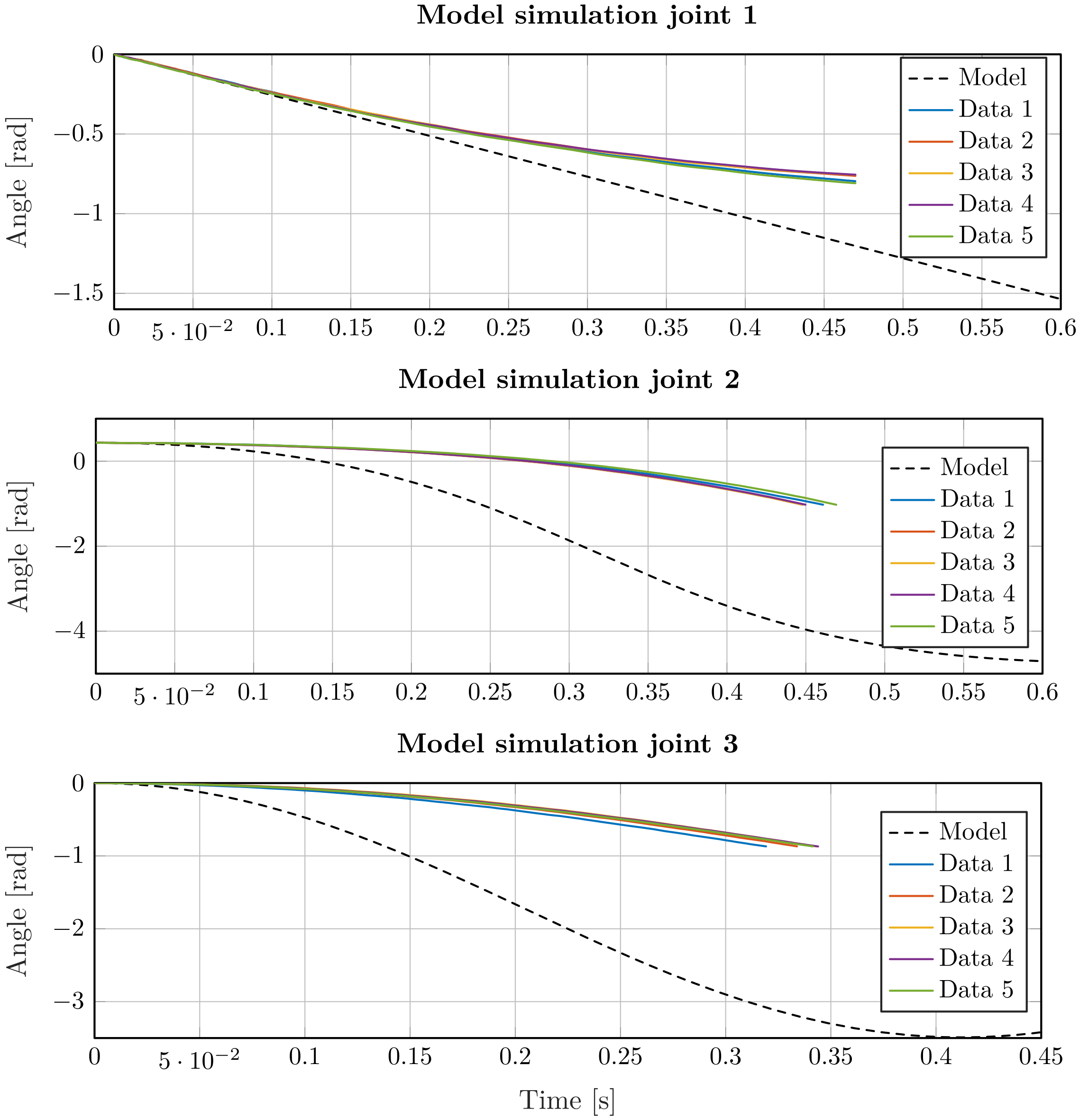

Figure 4.2: Simulation of the model without friction components. These figures are to illustrate the similar behavior between the model and the data measurements.

In *Figure 4.2*, a simulation with the model and its respective data can be seen, where the initial condition for the model is set to the same as the gathered data. As the robot is limited in movement due to physical constraints, i.e., in an upward and downward manner, the data for joint two and three seems to be cut.

In *Figure 4.2*, it can be seen that the model has a faster response compared to the data. This is expected as the model do not include any friction components. Furthermore it should be noted that the model trajectory is in a downwards direction similar to the data. Since the model react similar to the data, the initial validation of the model is accepted.

4.4.2 Second validation step

In the second validation step, friction coefficients are added to the model. This is done to get a better fit between the model and the gathered data. The model in *Equation: (4.12)* contains two types of friction; viscous and coulomb friction. Due to time constraints and that the focus is on collision detection, these parameters are estimated by hand until a suitable response of the model was reached. The friction coefficients can be found in *Appendix: B*.

In *Figure 4.3*, the model with included friction components is shown.

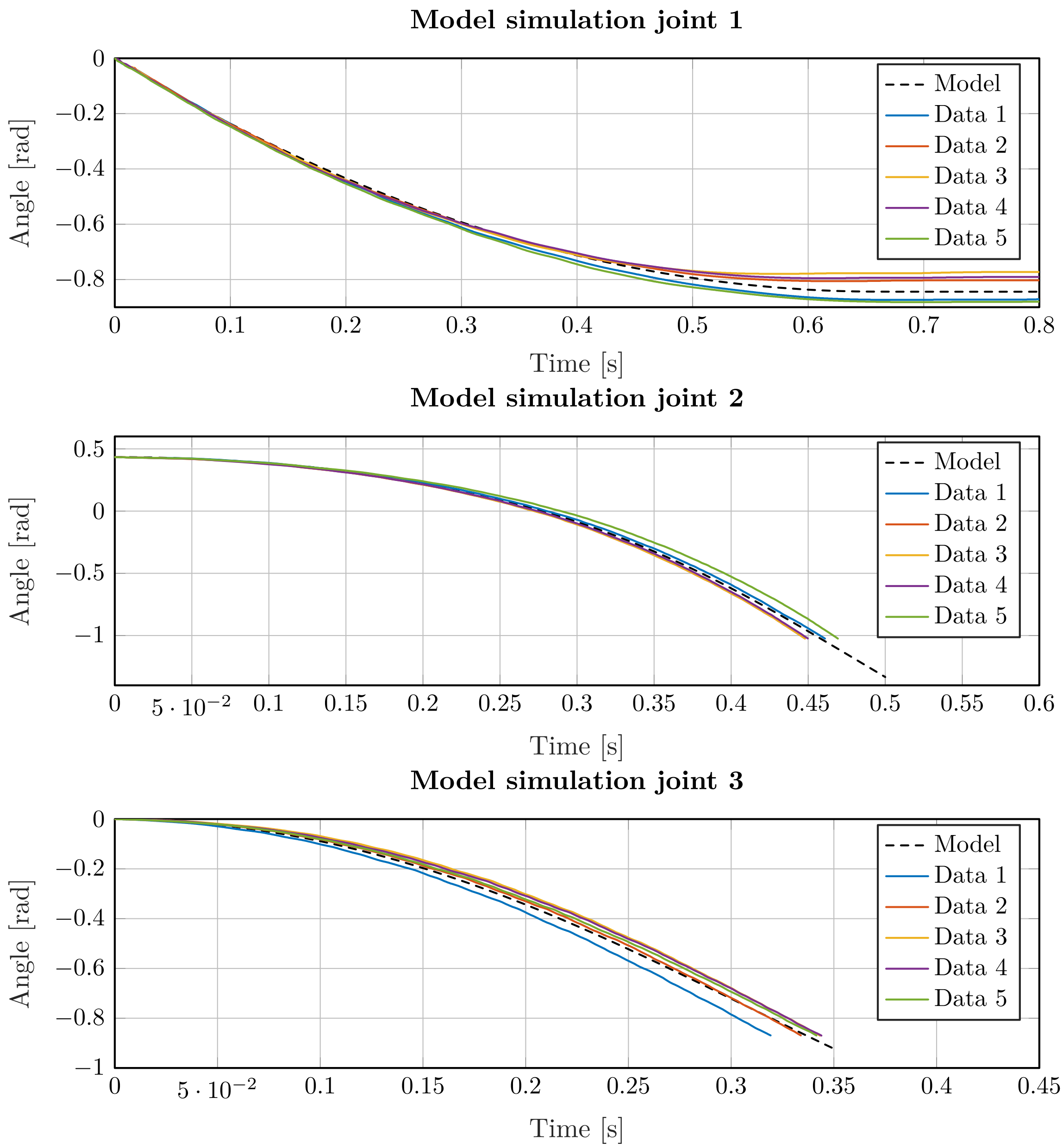

Figure 4.3: Simulation of the model with friction components.

It can be seen that the added friction gives the model an almost identical response to the data. Again, because the uStepper is limited in its physical movement, it is not possible to conclude whether the model stops at the correct position with regards to joints two and three. However, since the trajectory of the model follows the trajectory of the data it is deemed acceptable compared to joints two and three. With regards to joint one, it can be seen that the model follows the trajectory of the data and also stops at a position in between the data sets.

With the added friction to the model, the response is almost identical to the data. From these simulations the model is deemed acceptable for control and detection purposes.
In the next chapter a controller and observer for the uStepper is derived.

Controller and observer 5

This chapter elaborates on the controller and observer derived for this project in greater detail.
The chapter will be focusing on deriving a PID controller with gravity compensation and an attempt on making an extended Kalman filter for the system. An additional observer strategy is introduced at the end of this chapter.

5.1 PD control with gravity compensation

In this section some design consideration concerning the stability of a PD controller will be elaborated. The stability analysis of the controller will be based on Lyapunov direct method [30, p. 596], which requires that *Equation: (5.1)* is fulfilled.

$$\begin{aligned} &V(q) = 0, && \text{for} \quad q \equiv 0 \\ &V(q) > 0, && \text{for} \quad q \neq 0 \\ &\dot{V}(q) \leq 0 && \text{for} \quad q \neq 0 \end{aligned} \tag{5.1}$$

Where
V is a Lyapunov function.

If the criteria in *Equation: (5.1)* is achieved the system is said to be stable in the sense of Lyapunov at the equilibrium point set by the user.

To obtain the first and second criteria for stability, a Lyapunov candidate as seen in *Equation: (5.2)* is chosen.

$$V(\dot{q}, \tilde{q}) = \frac{1}{2}\dot{q}^T B(q)\dot{q} + \frac{1}{2}\tilde{q}^T K_P \tilde{q} \quad > \quad 0, \quad \forall\, [\dot{q}, \tilde{q}] \neq 0 \tag{5.2}$$

Where
K_P is a symmetric and positive definite matrix,
and $\tilde{q}$ is the error between the desired and actual position, $(q_d - q)$.

Since the inertia matrix is positive definite, this candidate will always be positive definite regardless of the states values in q due to its quadratic form. It will only become zero when the states takes a value of zero, thus, fulfilling the first and second requirement.

Taking the time derivative of *Equation: (5.1)* yields *Equation: (5.3)*.

$$\dot{V} = \dot{q}^T B(q)\ddot{q} + \frac{1}{2}\dot{q}^T \dot{B}(q)\dot{q} - \dot{q}^T K_P \tilde{q} \tag{5.3}$$

Solving for $B(q)\ddot{q}$ in *Equation: (4.11)* and substituting the result into *Equation: (5.3)*, results in *Equation: (5.4)*.

$$\dot{V} = \frac{1}{2}\dot{q}^T(\dot{B}(q) - 2C(q,\dot{q}))\dot{q} - \dot{q}^T F_v \dot{q} - F_s sign(\dot{q}) + \dot{q}^t(u - G(q) - K_P\tilde{q}) \tag{5.4}$$

Equation: (5.4) can be simplified as the first term in the right hand side is zero due to the principle of energy conservation[30, p. 259], which simplifies it to *Equation: (5.5)*.

$$\dot{V} = -\dot{q}^T F_v \dot{q} - \dot{q} F_s sign(\dot{q}) + \dot{q}^T(u - G(q) - K_P\tilde{q}) \tag{5.5}$$

Choosing a control input u as seen in *Equation: (5.6)* results in a controller with gravity compensation with proportional action.

$$u = G(q) + K_P\tilde{q} - K_D\dot{q} \tag{5.6}$$

Where
K_D is a positive definite matrix.

result in a controller with gravity compensation with proportional action.

A block diagram of the controller can be seen in *Figure 5.1*.

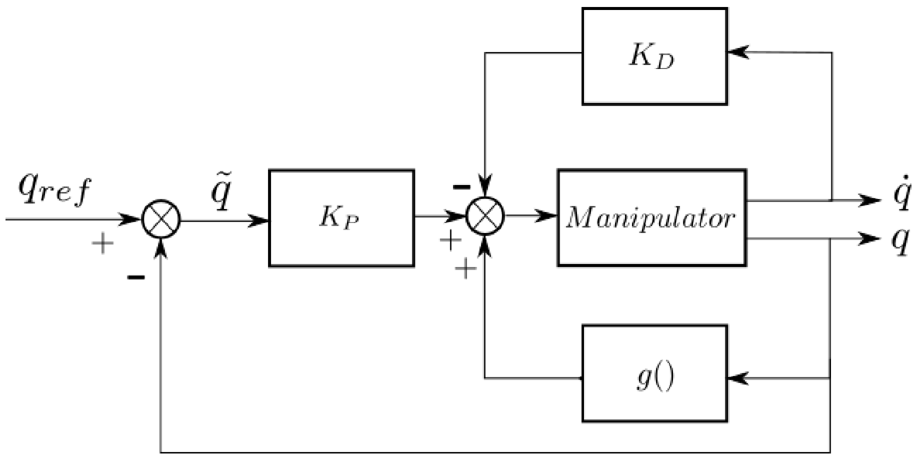

Figure 5.1: Block diagram of the PD controller with gravity compensation.

This controller depends on model parameters for the compensation. As a consequence, the model has to be ideal, i.e., the parameters of the model have to be identical to those of the actual system. An ideal model is virtually impossible to establish, especially because parameters may vary over time, e.g., due to wear. For this reason, integral action should be applied to the control. This ensures that the system reaches the desired trajectory points and guarantees a steady state error of zero. Including an integral action to the above controller results in a PID controller with gravitational compensation, see *Figure 5.2* for block diagram.

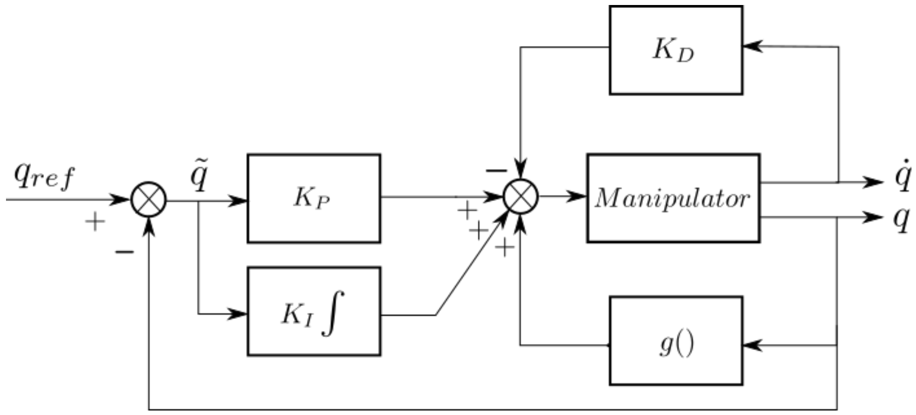

Figure 5.2: Block diagram of the PID controller with gravity compensation.

5.1.1 Control simulation

A simulation were made to find initial values for the PID gains before implementing the controller. The simulation was given the angular reference of [0.3 0.5 0.8] and initialized at an angular position of zero for each joint. Furthermore, because the motor is limited in how much torque it can deliver in the actual system, a saturation of the control input was implemented for this simulation.

The simulation of the controller can be seen in *Figure 5.3*.

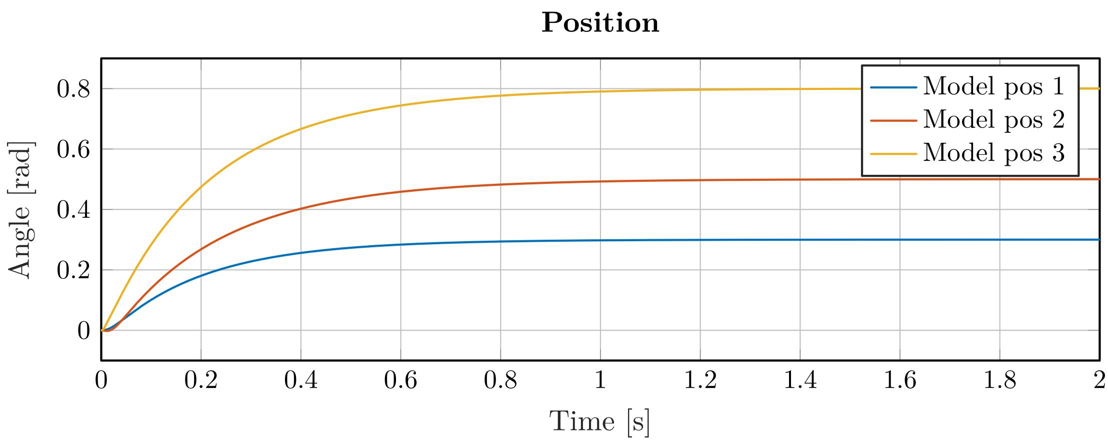

Figure 5.3: Simulation of the PID controller.

The desired angular reference is reached by the controller a bit after one second. As the model is imperfect due to the simplifications of the model, further tuning of the controller gains are not made since these would have to be further tuned after implementation.

Because the controller depends on angular velocities and the system only provides positions, the next section will be focusing on the design of an extended Kalman filter, for estimating the angular velocities.

5.2 Extended Kalman filter

The sensors available for the system provide position and torque and are described in Section *2.1.1: Actuator*. The angular velocities can be calculated by taking the derivative of the position. However, this derivative is usually sensitive to noise. To overcome this issue, an EKF is used to increase the estimation of the velocity of each joint.

The model described in Section *4.3: State space representation*, is discretized by the method of forward Euler[31] and noise is added to the system and its output function, see *Equation: (5.7)*.

$$\begin{aligned} \Theta_{k+1} &= \Theta_k \ + \ Ts\, f(\Theta_k, \tau_k, f_{e_k}) \ + \ w_k \\ Y_k &= h(\Theta_k) \ + \ v_k \\ w_k &\in \mathrm{NID}(0, Q) \\ v_k &\in \mathrm{NID}(0, R) \end{aligned} \tag{5.7}$$

Where

	Ts	is the step size,
	w	is the model noise,
	Q	is a diagonal and positive definite matrix with system variance noise,
	v	is the sensor noise,
and	R	is a diagonal and positive definite matrix with sensor variance noise.

The sensor and model noise is modelled as white Gaussian noise. That is, the diagonal of Q and R is the noise variance and the off-diagonal elements are zero because there are no correlations. Unfortunately the datasheet[32] did not present the noise variance of the sensors for the motor controllers. Instead, these are estimated by sampling a steady state position of the system over a period of three minutes. However, this did not provide any usable results because the gathered data had zero variance. Instead, the variance is estimated using the precision of the sensors. *Equation: (5.8)* provides the final estimates of variance.

$$R = diag([12 \ \ 12 \ \ 12]) \cdot 10^{-3} \tag{5.8}$$

To capture backlashing and other unmodelled dynamics in the actual system, the variance of the system noise, Q, is used as tuning parameters for the Kalman filter.

The EKF is a recursive filter that includes the two steps; time update and measurement update. The time update step predicts the next state of the system, given the present state and input, whereas the measurement update step contains the state estimation. The EKF algorithm is shown in *Equation: (5.9)* (time update step) and *Equation: (5.10)* (measurement update). The equations use the following subscripts notations.

- k: denotes the present time step,
- $k|k-1$: denotes the present time step k given previous time $k-1$.

Time update from k to $k+1$

$$\begin{aligned} \hat{\Theta}_{k+1|k} &= f(\hat{\Theta}_{k|k}, u_k, f_{e_k}) \\ P_{k+1|k} &= F_k P_{k|k} F_k^T + Q_k \end{aligned} \tag{5.9}$$

Where

	$\hat{\Theta}$	is the state prediction,
	P	is the error covariance for the Kalman filter,
and	F	is the Jacobian of the system model.

Measurement update after receiving y_k and u_k

$$\begin{aligned}
\hat{Y}_{k|k-1} &= h(\hat{Q}_{k|k-1}, u_k) \\
\tilde{Y}_{k|k-1} &= Y_k - \hat{Y}_{k|k-1} \\
K_k &\triangleq P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1} \\
\hat{\Theta}_{k|k} &= \hat{Q}_{k|k-1} + K_k \tilde{Y}_{k|k-1} \\
P_{k|k} &= (I - K_k H_k) P_{k|k-1} (I - K_k H_k)^T + K_k R_k K_k^T
\end{aligned} \tag{5.10}$$

Where

	$\hat{Y}$	is the estimated output,
	$\tilde{Y}$	is the output error,
	K	it the Kalman gain,
and	H	is the Jacobian of the output function.

The Jacobians F and H can be found as the partial derivative of the system model and the output function with respect to the states. This is shown in *Equation: (5.11)*

$$F \triangleq \frac{\partial f}{\partial \Theta^T} \quad \wedge \quad H \triangleq \frac{\partial h}{\partial \Theta^T} \tag{5.11}$$

5.2.1 External force estimation

The model described in Section *4.3: State space representation* contains only six states that represents the positions and velocities. As a consequence, the EKF contains six states as well. However, if the model is altered with three additional states that represent the external forces applied to the system, the EKF should be capable of estimating these as well. As such, the EKF can be used to detect whether a collision has occurred.

For simplicity the continuous time model is summarized as:

$$\ddot{q} = B(q)^{-1}[\tau - J^T(q) f_e - C(q, \dot{q})\dot{q} - G(q) - F_v \dot{q} - F_s sgn(\dot{q})]$$

It can be seen that the contribution of the external forces to the system, adds an additional acceleration to the system. Therefore, extracting the term related to the external force, the acceleration contribution can be denoted as seen in *Equation: (5.12)*.

$$\ddot{q}_e = B(q)^{-1} J^T f_e \tag{5.12}$$

Where

$\ddot{q}_e$	is the external force acceleration to the system.

the external forces can then be isolated as three new states for the system, see *Equation: (5.13)*.

$$f_e = J^{-T} B(q) \ddot{q}_e \tag{5.13}$$

By including *Equation: (5.13)* to the model and following the design procedure described for the EKF algorithm, it should be possible to estimate the external force to the system using the EKF.

5.2.2 Extended Kalman filter simulation

The EKF depends on a prediction step size for the prediction step. An initial guess of the step size is found through a rough implementation of the EKF on the Teensy board, where the time for the calculation and sampling of the positions is measured. This results in an average time of 7.44 ms, which is rounded up to 8 ms, such that variance of calculation and sampling are taken into account. The C++ code for the EKF is generated by the Matlab toolbox *Matlab Coder*[33]. This toolbox requires that the function, in this case the EKF, is defined as a Matlab function and that the inputs for the code generator are specified. The toolbox then returns the C++ code and its needed libraries.

With the found step size, the initial tuning of the EKF is made. The tuning is made such the the EKF can track the discrete model seen in *Equation: (5.7)*, where the step size of the EKF is set to 8 ms and the step size of the model 10 times faster.
The initial value for the error covariance is set to a [9x9] identity matrix.

Figure 5.4, *Figure 5.5*, *Figure 5.6*, and *Figure 5.7* shows the simulation of the EKF tracking the discrete model. These simulation shows the position, velocity, residual, and the force estimation made by the EKF.

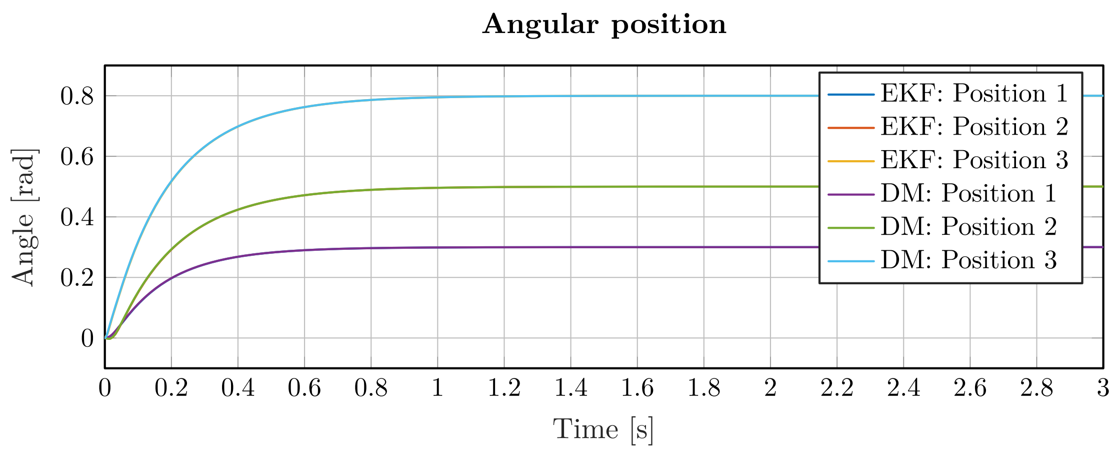

Figure 5.4: The EKF tracking the position of the discrete model. DM is short for *discrete model*

Figure 5.4 shows that the EKF follows the discrete model very closely and lies directly on top of the model. This is expected as the noise variance defined for the sensors are very small, and consequentially, the tracking is identical to the model.

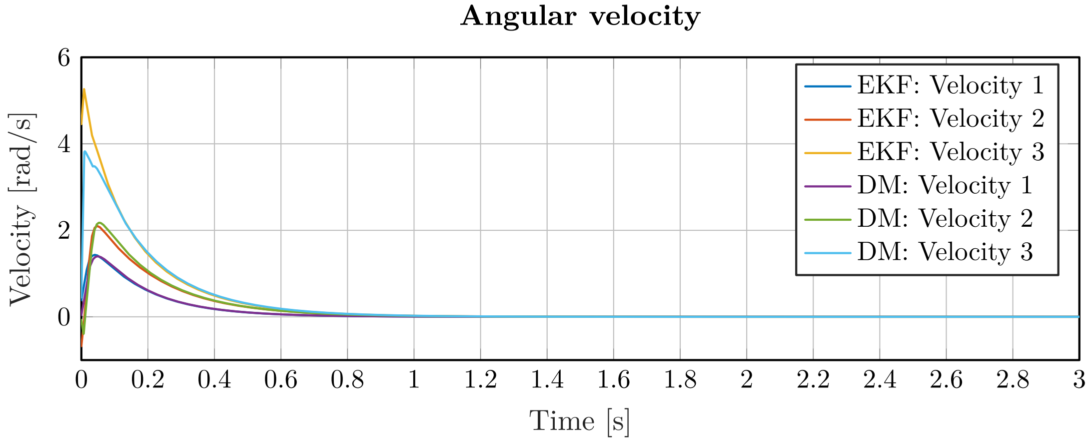

Figure 5.5: The EKF tracking the velocity of the discrete model. DM is short for *discrete model*

Figure 5.5 shows the EKF velocity estimates. The figure shows a large error for the velocity of joint three at the initialization of the EKF. However, after 0.2 seconds the EKF converged towards the actual velocity of the model. Different parameters in the model variance, Q, were tested out. It was, however, not possible to lower the overshoot of the velocity without compromising the fit of the other two estimates.

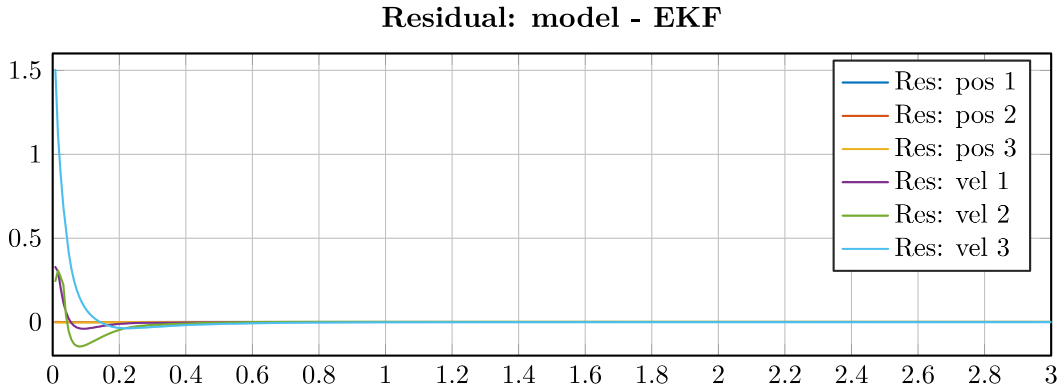

Figure 5.6: The residual of tracking the discrete model.

Figure 5.6 shows the estimation residuals for the EKF. The position estimations are identically zero for the entire estimation period while the velocities first have converged after 0.6 seconds.

In the simulation of the EKF, an external force is added to the model to see if the EKF can capture this disturbance, see *Figure 5.7*.

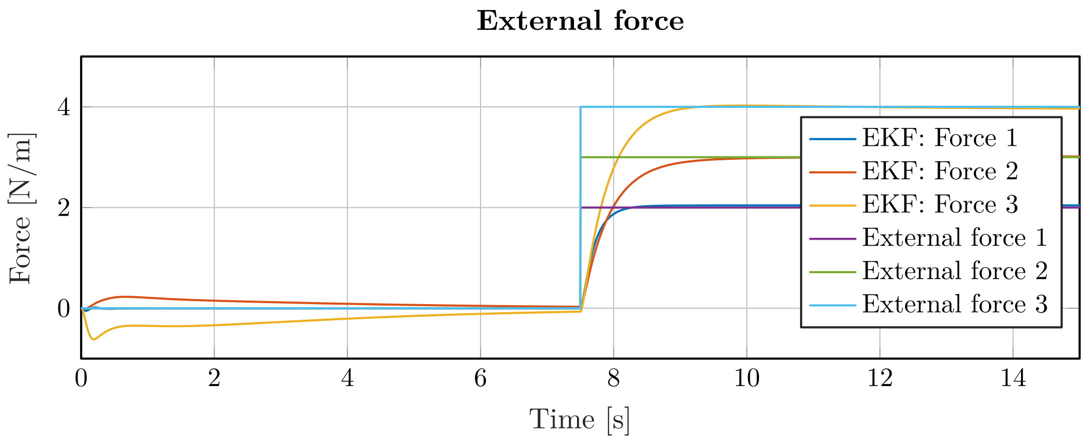

Figure 5.7: Simulation of the EKF estimating the external force from the additional acceleration to the discrete model.

It can be seen that the EKF at the beginning estimates that external forces is present even though this first is applied at time step 7.5 second. This could be because of the initial conditions set for the error covariance matrix that were set to the identity matrix. A better initial guess on this matrix might give a lower force estimation for the beginning of the EKF.
At time step 7.5 seconds, an external force is added to the discrete model, which can be seen as the straight lines on *Figure 5.7*, (External force one, two and three). When the external force is applied the EKF starts reacting to this. The convergence of the estimated external force could be further tuned by the EKF parameter Q by increasing the elements corresponding to the external forces. This would lead to a more steep transaction from no force applied to applied force. However, it was observed that the filter became much more sensitive with a larger overshoot.

In this section simulations of the EKF is made. In the next section the implementation of the controller and EKF on the Teensy board is made were test measurements is shown.

5.3 Implementation

In this section the focus is on the implementation of the designed controller and why the EKF is discarded due to large tracking error of the actual system. As the EKF is discarded an alternative solution is presented and implemented to the board.

Control implementation

The controller is implemented on the Teensy, see Section *2.1.2: Micro controller*, for initial testing. Since the controller depends on the angular velocity of the joints, and only position readings is possible, the velocities is for these tests derived from the position.

Initially the controller was implemented with the found simulation gains, however this responded in a very aggressive controller with large overcompensation and continuous oscillation around the reference. The gains were adjusted by decreasing the proportional gain and increasing the derivative gain. These were adjusted until the system had a fast and smooth response.

In the following figures, the system response with the controller is shown. The figures are shown in a chronological order from motor one to three.

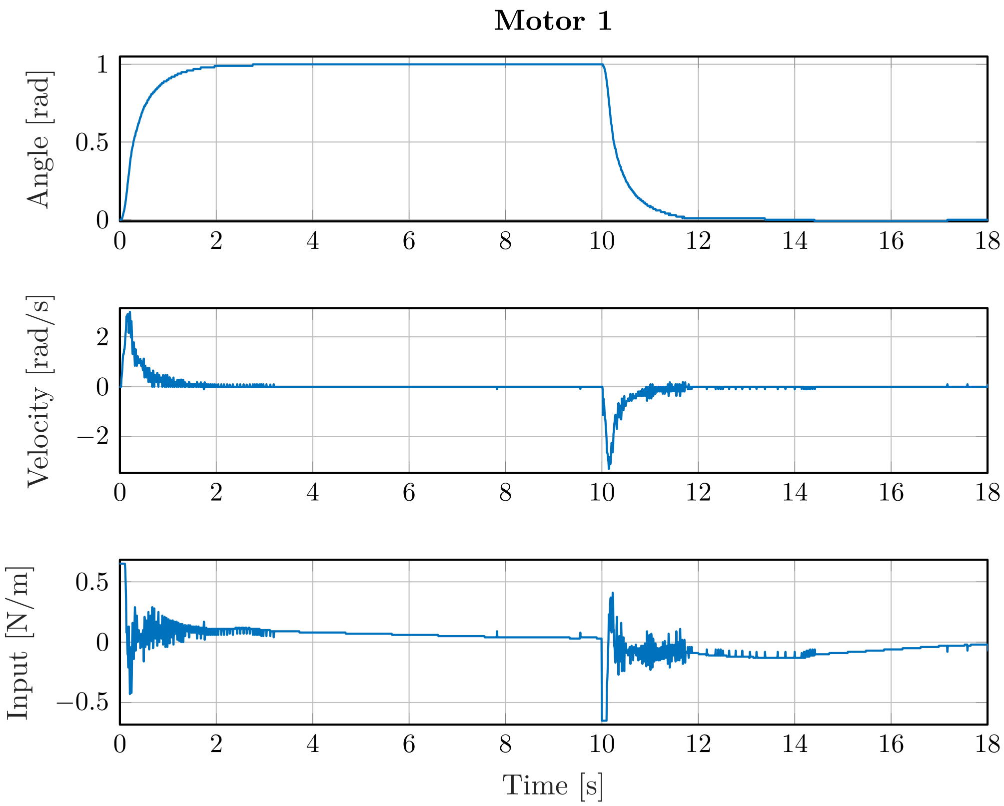

Figure 5.8: Implemented PID controller for motor 1 with angular reference 1 and 0.

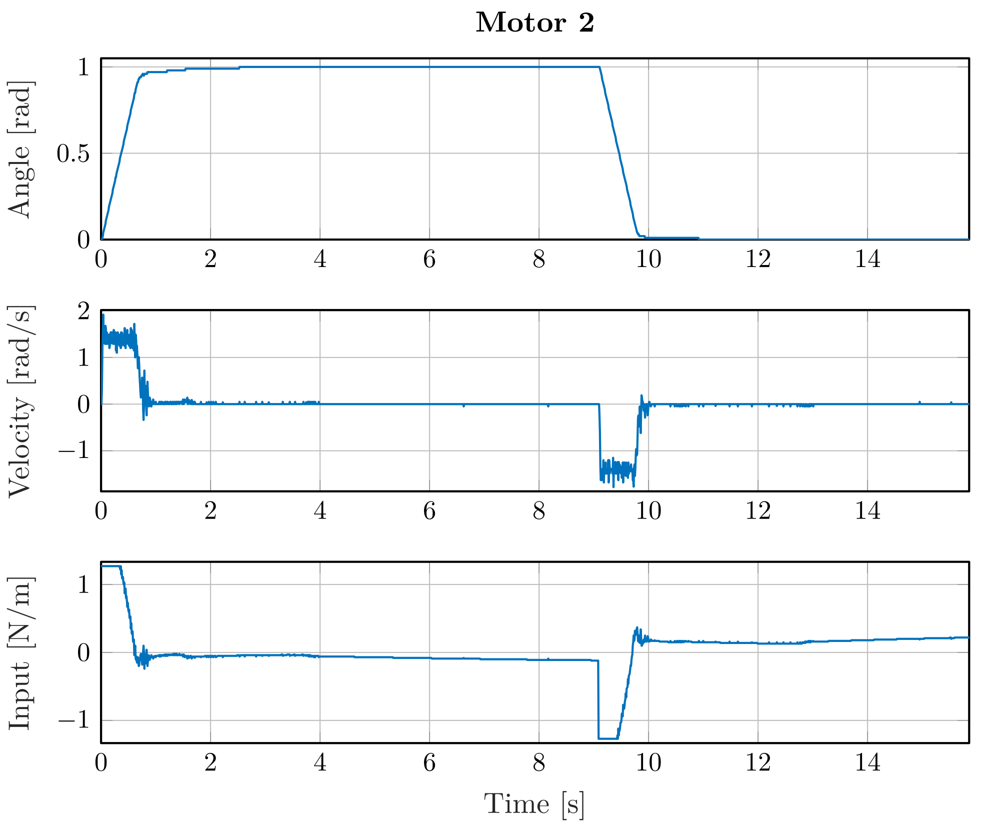

Figure 5.9: Implemented PID controller for motor 2 with angular reference 1 and 0.

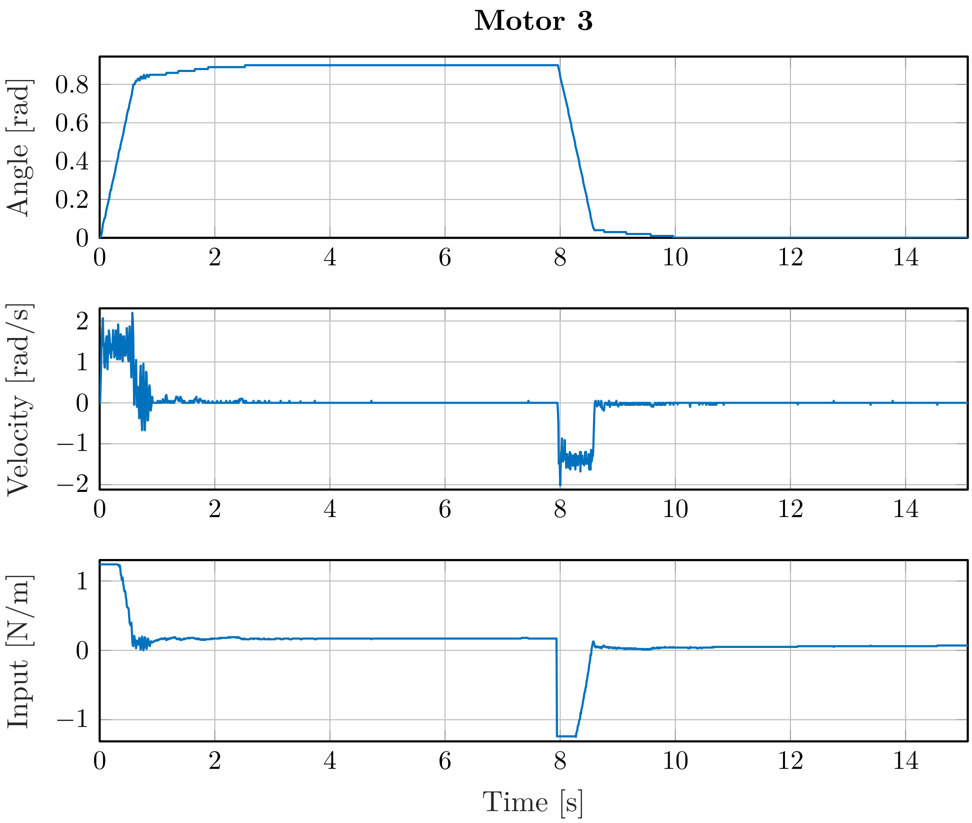

Figure 5.10: Implemented PID controller for motor 3 with angular reference 0.9 and 0.

From *Figure 5.8* to *Figure 5.10* three individual test can be seen. Each test are giving two angular references, for which the controller makes the system reach within 2.3 seconds. In comparison to the simulated controller, see *Figure 5.3*, the found gains gives the system a slower response. These gains where tuned in such a way that the system had the most stable response possible without or little vibrations. Since the system is not completely stiff in its joints, large gains resulted in the system making a whip effect when reaching the reference, thus making the system starting to shake around the reference. Lowering the gains made the system response in a more suiting manner, however with the compromise of a fast response time.

The previous figures only showed the movement of one actuator at a time. In *Figure 5.11* all three actuators are set to move simultaneously.

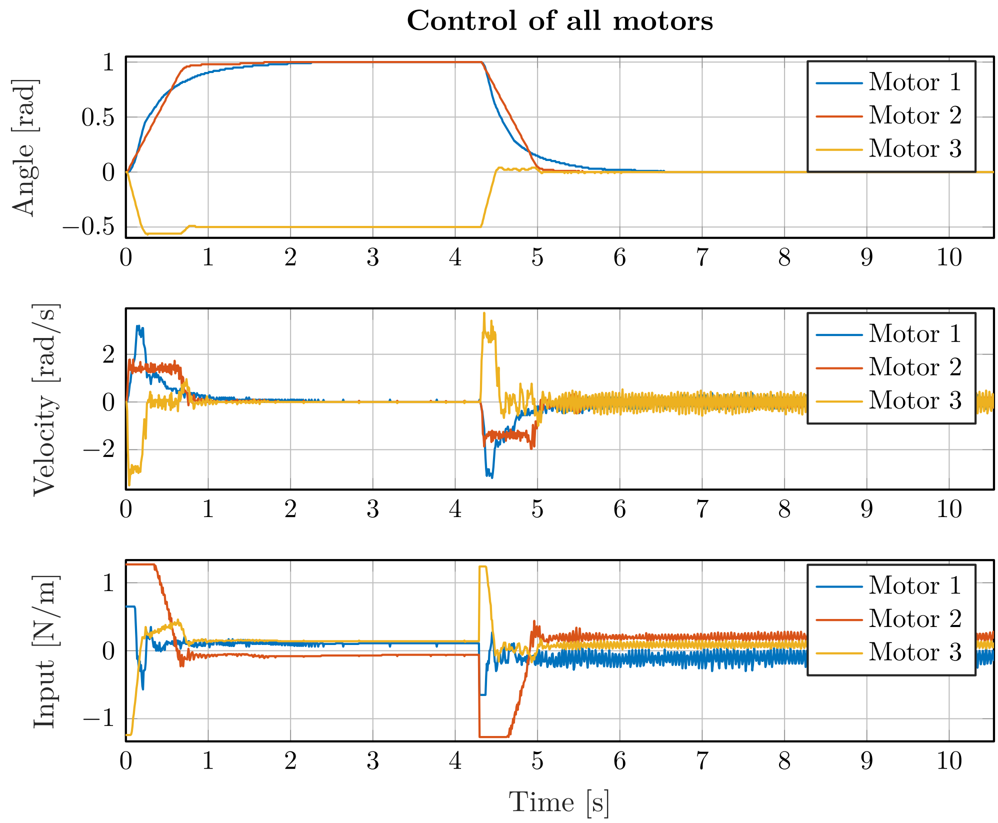

Figure 5.11: Implemented PID controller for system. References are set to 1 to 0 for motor one, 1 to 0 fro motor two and -0.5 to 0 for motor three.

It can be seen that the controller for motor three has a small overshot of -0.07 for the first reference and 0.05 for the second. This could be due to the additional energy contribution to the third link, when actuating joint one and two which result in a higher centrifugal energy. Secondly a lot of noise is present in the estimates of the velocities, which could have an influence on the input to the system and result in a higher torque. Third, joint three is not directly driving in the joint but by a parallelogram arm, see *Figure 2.1*. This result in the third joint being dependable on the position of joint two, which can be expressed by *Equation: (5.14)*.

$$q_3 = \frac{\theta_{m3}}{N} - \frac{\theta_{m2}}{N} \tag{5.14}$$

Where

	θ_{mi}	is the motor joint position,
and	N	is the gear ratio.

Due to this dependency, moving joint two will directly affect the position of joint three for which the regulator has to counter act this movement. To overcome this problem, one may modify the presented control strategy such that movement of joint two is seen as a disturbance for joint three, thus decouple the dependency of the two joints by the controller. This is however not implemented to the current strategy.

Implementation of extended Kalman filter

The implementation of the EKF were not as promising as the controller. Before including the EKF to the control scheme, the EKF was only set to estimated the position/velocities of the system without feeding the estimates to the controller i.e the controller was running independently of the EKF in the same manner as mentioned in Section *5.3: Control implementation*. The inputs to the EKF is the angular position of the joints and the control input to the system. In *Figure 5.12* too *Figure 5.14* the estimations made by the EKF is shown.

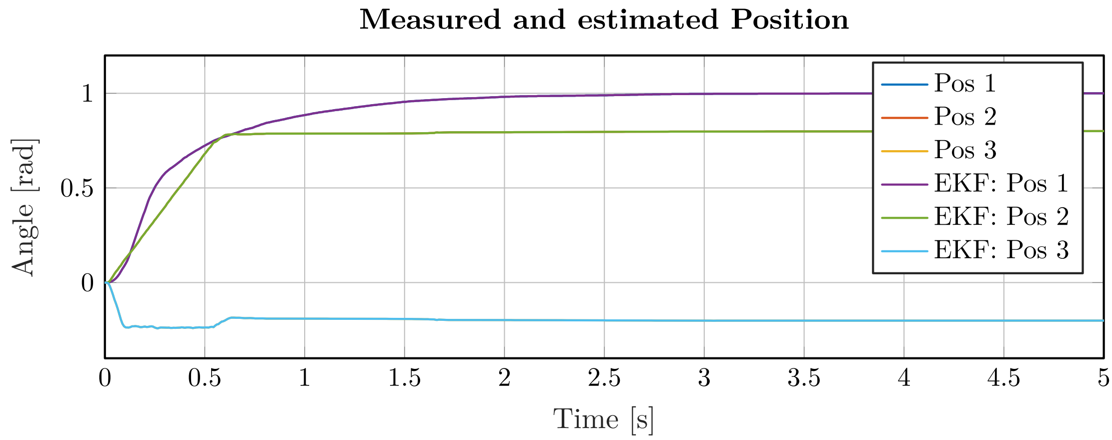

Figure 5.12: Position measurements and position estimates made by the EKF.

In *Figure 5.12* the measured positions and the EKF estimates is shown. It can be seen that the measurements and estimates are the same. This is expected as the measurement variance is set to a low value and the influence of the EKF should be very low.

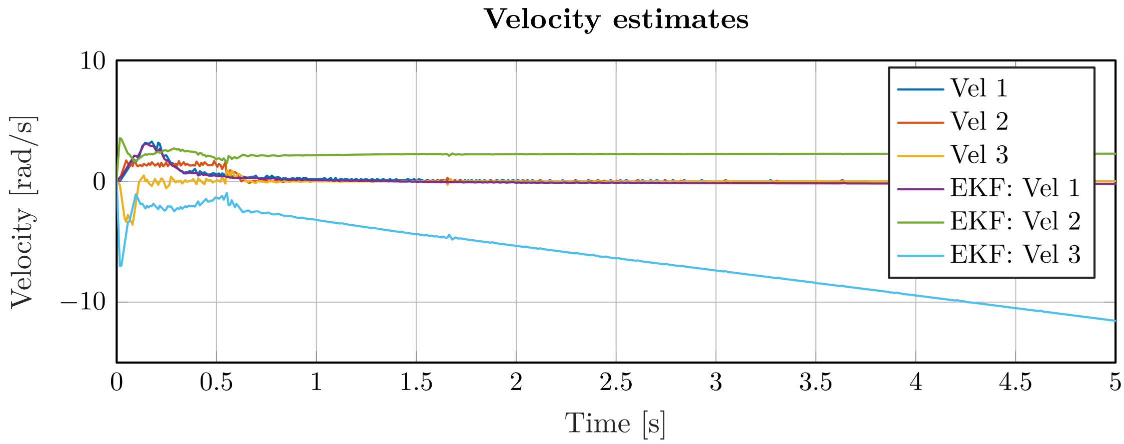

Figure 5.13: Numerical estimates of the velocities and estimations made by the EKF.

In *Figure 5.13* two velocity estimates is shown. One by taking the numerical derivative of the position and the other by the EKF. It were expected that the estimate made by the EKF followed the transient of the numerical derivative. This is however not the case for the velocity of joint two and three. Joint two goes to a static positive velocity, when the system do not move and joint three actually diverges from the numerical derivative. The only estimation made by the EKF which is following the transient of the numerical derivative is the velocity for joint one, see *Figure 5.14* for a close up.

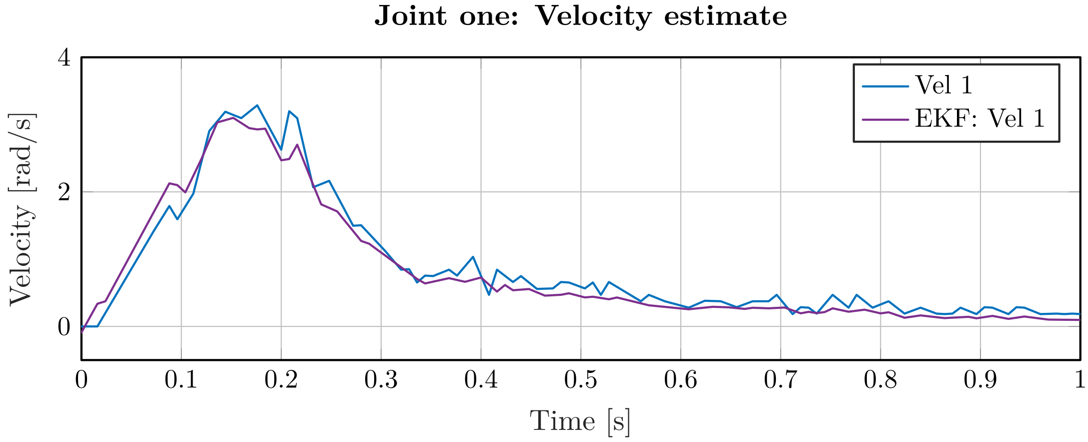

Figure 5.14: Close up of the numerical and EKF estimates of the velocity of joint one.

It can be seen that the EKF estimates the velocity of joint one and actually smooth out the velocity estimates in comparison to the numerical derivative.

In regards to increase the performance of the EKFs estimations of the velocity, increasing and decreasing the EKF model parameter, Q, did not improve the velocity estimations. An analysis of the EKf were made to see if this could be solved, however this were not possible to directly specify where the problem was, however some observations were made. Simulating the discrete model with a step size of 8 ms showed instability in the model when the velocities were going towards zero. The instability issue only showed up for joint two and three, where joint one remained stable. Lowering the static friction related to joint two and three solved the instability issue, however, doing so also made the response of the model become a decaying oscillation.
It was not possible to solve the problem within time, thus, the EKf was discarded and another observer strategy was made based on a linear Kalman filter and an open loop external force computation. This will be described in the next section.

5.4 Alternative observer strategy

The following observer strategy is based on a stationary linear Kalman filter, which is used for the estimation of position, velocity and acceleration of the system. The estimates are given as inputs to the dynamic model, where the external force, f_e, is isolated as seen in *Equation: (5.15)*. The estimation of the external force is thereby made as an open loop.

$$f_e = J^{-T}(q)[\tau - B(q)\ddot{q} - C(q,\dot{q})\dot{q} - G(q) - F_v\dot{q} - F_s sgn(\dot{q})] \quad (5.15)$$

This approach is not the best strategy for the estimation of the external force because it requires that the dynamic model is identical to the actual system. Furthermore, because the estimate of the external force is made by open loop, the estimation will not converge towards the actual external force in case of imperfections in the model.
A block diagram of the observer strategy can be seen in *Figure 5.15*.

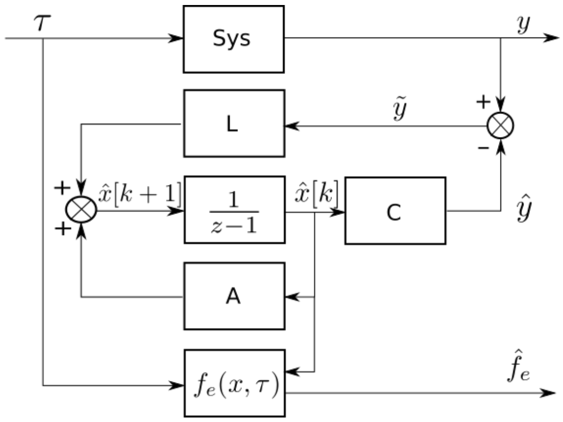

Figure 5.15: Block diagram of the linear stationary Kalman filter with the open loop estimation of the external force.

The linear Kalman filter for this project is based on the system seen in *Equation: (5.16)*.

$$\begin{aligned} \hat{x}_{k+1} &= A\hat{x}_k + L(y_k - \hat{y}_k) \\ \hat{y}_k &= C\hat{x}_k + v_k \end{aligned} \tag{5.16}$$

Where

	$\hat{x}$	is the state estimate vector, i.e $x = [q\ \dot{q}\ \ddot{q}]^T$,
	A	is the system matrix,
	L	is the stationary Kalman gain,
and	C	is the system output matrix.

The matrices A and C is designed as

$$A = \begin{bmatrix} I & Ts & 0 \\ 0 & I & Ts \\ 0 & 0 & I \end{bmatrix} \qquad C = \begin{bmatrix} I_{3x3} & 0_{3x6} \end{bmatrix}$$

The Kalman gain is found through the Matlab function Kalman, which requires the system matrix, the noise covariance for the model and the sensors[34]. As with the EKF, the model covariance is again used as a tuning parameter for the filter. This function returns the Kalman gain that minimize the steady state error covariance of the state estimate seen in *Equation: (5.17)*,

$$\begin{aligned} P &= \lim_{k\to\inf} E(e[k|k-1]e[k|k-1]^T) \\ e[k|k-1] &= x[k] - \hat{x}[k|k-1] \end{aligned} \tag{5.17}$$

Where

e	is the state estimation error.

The Kalman gain is then derived by *Equation: (5.18)*.

$$\begin{aligned} L &= (PC^T + \bar{N})\bar{R}^{-1} \\ \bar{R} &= R + CQC^T \\ \bar{N} &= G(QC^T + N) \end{aligned} \tag{5.18}$$

Simulation of linear Kalman

The following simulations are based on data gathered from the system, where the model noise covariance, Q, is tuned until a suitable response of the state estimations is found. The Kalman estimation of velocities and accelerations is set up against a first and second order numerical derivative of the positions, i.e., velocity and acceleration.
Figure 5.16, *Figure 5.17*, *Figure 5.18*, and *Figure 5.19* shows the estimates made by the linear Kalman filter.

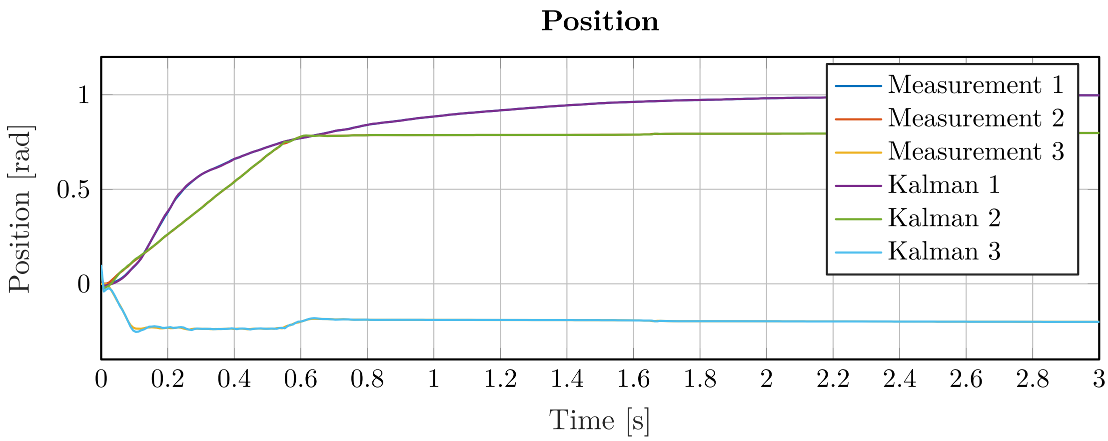

Figure 5.16: In the graph the position measurement and estimates made by the linear Kalman filter is shown.

Figure 5.16 shows the estimations and measurements from the system. The Kalman filter follows the measured positions closely and only minor errors at the beginning of the estimates are shown.

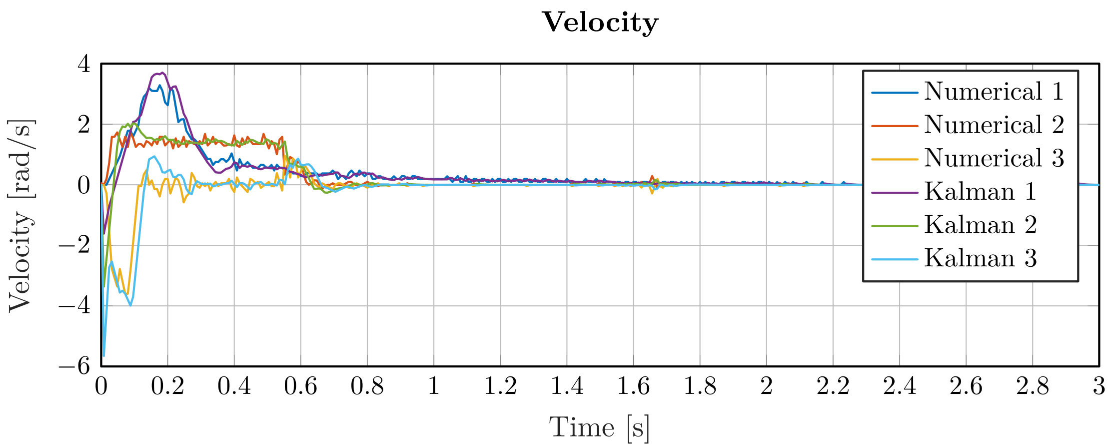

Figure 5.17: In the graph the numerical derivative and the linear Kalman estimates is shown for joint one, two and three.

The velocities are shown in *Figure 5.17*. Two estimations methods are shown: one is the numerical derivative, and the other is the Kalman filter. By comparing the numerical derivative with the Kalman estimate it is observed that the Kalman estimates follow the trajectory of numerical derivative. Furthermore, due to the low pass effect of the Kalman filter the estimates are smoother. That is, the Kalman filter has a lower frequency components than the numerical derivative.

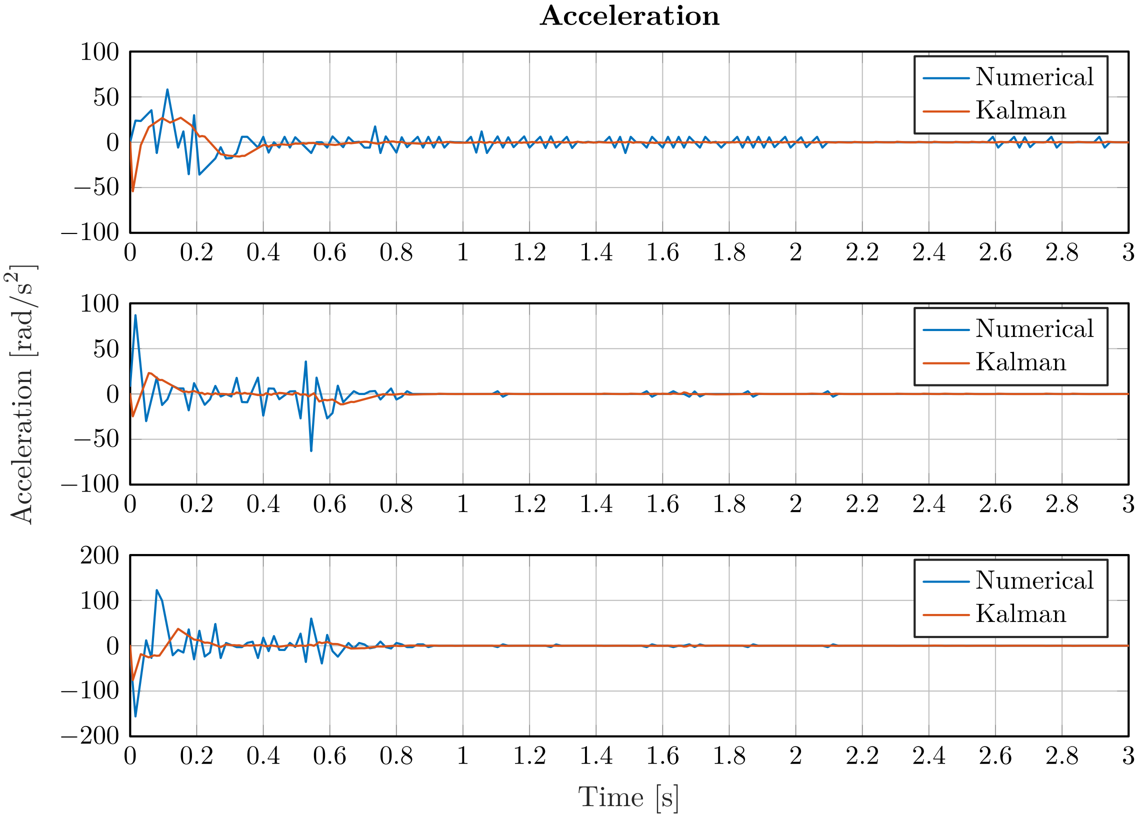

Figure 5.18: In the graphs the numerical and linear Kalman estimates are shown. Acceleration of joints one (top), two (middle), and three (bottom).

Figure 5.18 shows the acceleration estimates. The Kalman filtering is challenged in tracking the acceleration of the system but eventually converges. The Kalman filter has not been tuned to follow the numerical derivative of the acceleration as this over-estimates the acceleration.

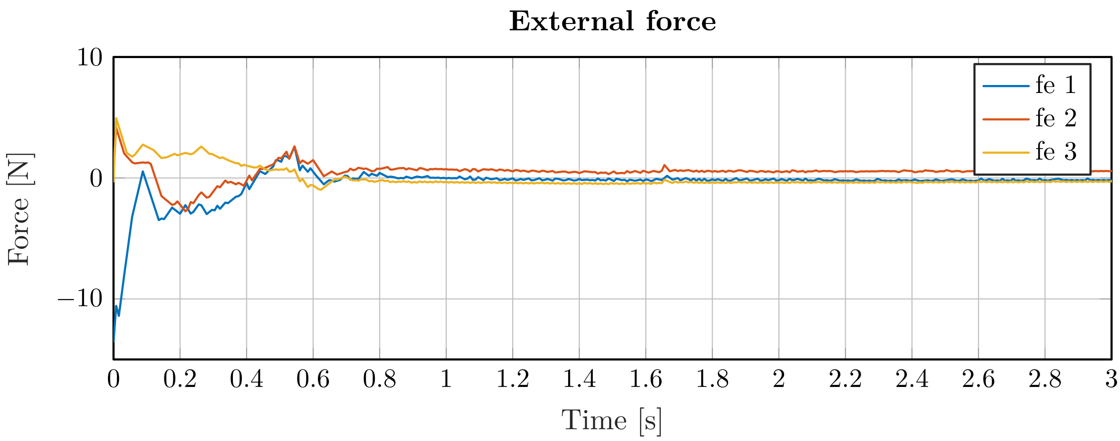

Figure 5.19: In the graph the open loop estimates are shown. Fe_i correspond to the external force being reflected to joint one to three.

Figure 5.19 shows the external force estimation. It can be seen that during initialization of the system, a large external force is estimated even though no force is applied to the system. Around 0.8 seconds the estimates have converged to a biased estimate, which could be due to simplifications in the model.

A small analysis was made in an effort to determine the cause of the large overshoot at the initialization of the Kalman filter. It was observed that the overshoot only appeared when new references were given to the system.
When the system is initialized different parameters are set for the system, e.g., maximum velocity and how fast the transient of new torque input is applied. The velocity of the motors were limited to 5.72 [rad/s]. Due to the specific gearings, this limitation results in a maximum velocity of 2.73 [rad/s] for joint one and 1.39 [rad/s] for joints two and three. These values are clearly in conflict with the numerically calculated velocities shown in *Figure 5.17*.

How the control loop of the maximum velocity is implemented is not specified in the datasheet. One guess could be that the actual input to the system is lower than the giving reference, for lowering the velocity. Therefore a test was carried out to see if increasing the limit would decrease the overshoot of the force estimates.

In *Appendix: D*, a test was made to see if increasing the velocity limit would decrease the overestimate of the external force. However, this did not seem to improve the estimation but rather decrease the performance.
While testing, the encoder of motor one was damaged. This damage resulted in corruption of the position measurements of joint one. It appears that the encoder disk was loose and would start to spin given external force, even without movement of the system. Therefore, new references was only given to motors two and three.
It was observed that actuating motor two and three reflected back to a external force applied to joint one. This was to be expected due to some off symmetry of physical structure of the system. However, in this case these correlations are deemed too high as these were the largest force estimates of the entire system.
In conclusion to this test, the model is deemed to be of poor quality with respect to capturing the actual system dynamics, which leads to large overshoots of the force estimations. If the the model was capable of capturing all the dynamics in the system and the linear Kalman filter estimating the states correctly, this would lead to a more correct force estimation.

Due to time constraints on the project, this problem is not solved, however the next chapter will cover the implementation and testing of the observer.

5.4.1 Implementing the observer

In this section the implementation and testing of the alternative observer strategy is made. This is made to see if the observer, when implemented, reacts in the same manner as in the simulated case.
It is expected that the implemented observer will react similar to the simulation described in Section *5.4: Simulation of linear Kalman* This postulate is based on the fact that the simulated case was designed from actual position data gathered from the system.

The graphs shown in *Figure 5.20*, *Figure 5.21*, *Figure 5.22*, *Figure 5.23*, and *Figure 5.24* are the observer estimates and the input to the system.

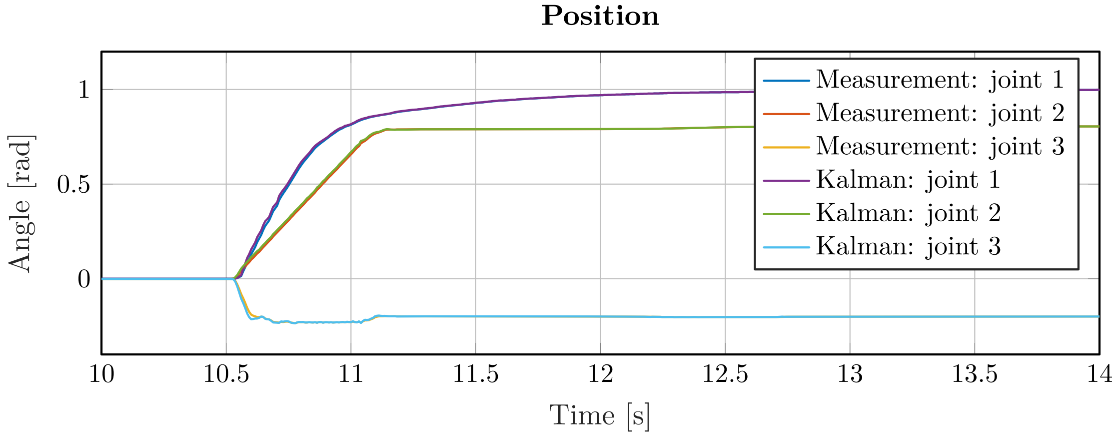

Figure 5.20: In the graph, the measurement and estimates made by the implemented linear Kalman filter is shown.

Figure 5.20 shows the position measurements and estimates. The estimates are lying on top of the measurements, which is to be expected because the noise variance for the position measurements are small.

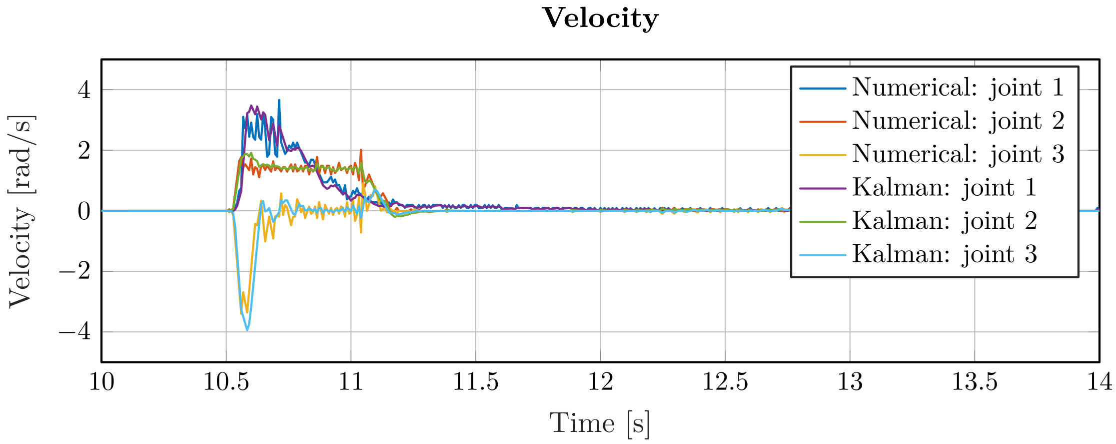

Figure 5.21: In the graph, the two estimates is shown. One being the numerical derived velocity, two being the implemented linear Kalman filter.

Figure 5.21 shows two estimates to test the reliability of these estimates. The graph shows the numerically derived velocities and the implemented Kalman estimates. It can be seen that the Kalman estimates follow the transient of the numerically derived velocities. Furthermore, it can be seen that the Kalman estimates have a low pass effect on the estimates in comparison to the numerically derived.

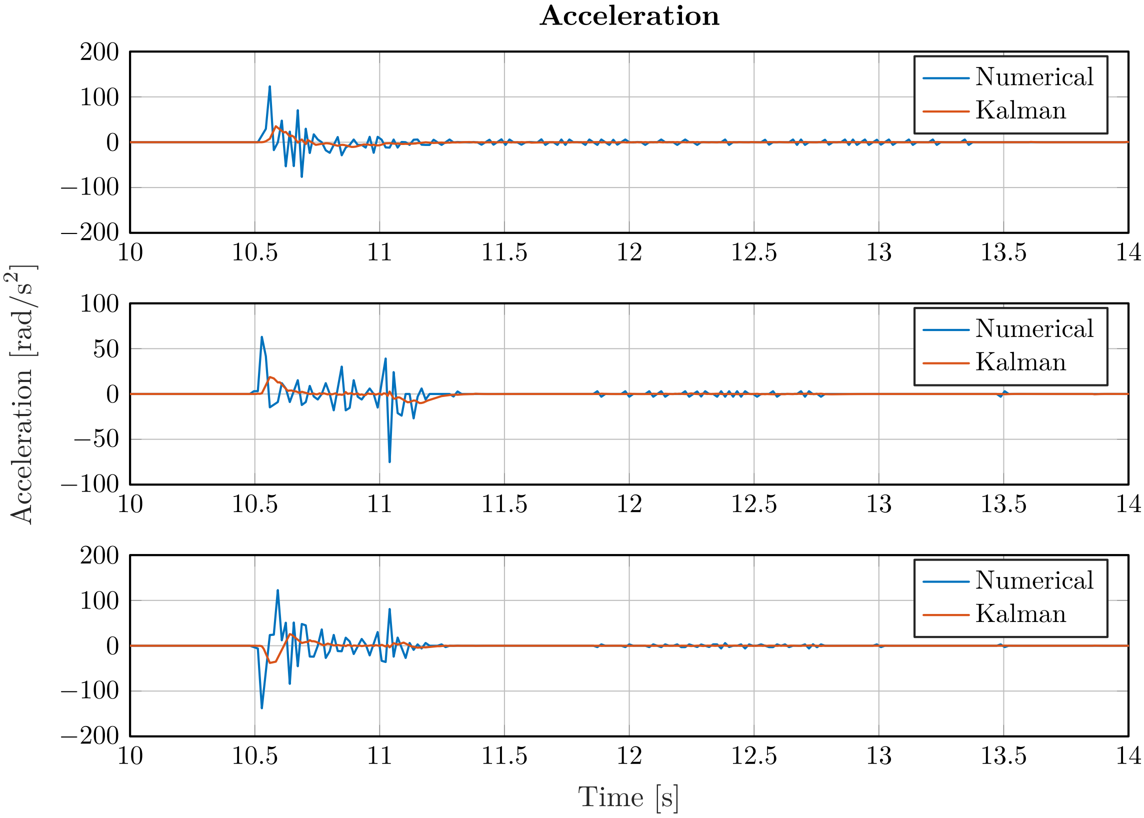

Figure 5.22: In the graphs, two acceleration estimates is shown. One being the numerical derived acceleration, two being the implemented linear Kalman filter. Acceleration of joints one (top), two (middle), and three (bottom)

Figure 5.22 shows the acceleration estimates. It can be seen that the numerical derived acceleration is rather sensitive to noise in the system as in the simulation case, see *Figure 5.18*. The high noise sensitivity results in the numerical acceleration having rather large spikes. The Kalman filter is a bit slower in comparison. The low-pass effect of the Kalman filter is smoothing the acceleration estimates in comparison with the volatile changes in the numerically derived.

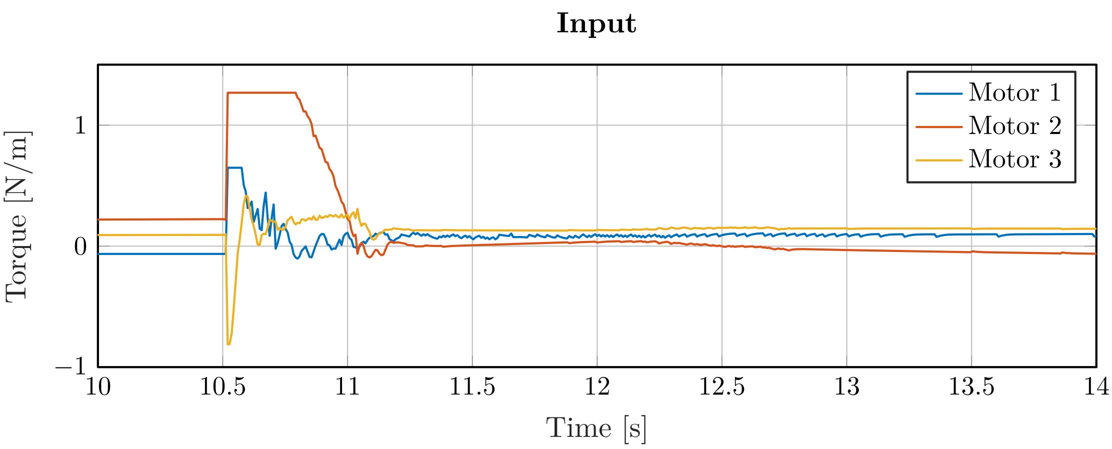

Figure 5.23: In the graph, the input to the system is shown.

Figure 5.23 shows the inputs to the system. The inputs for motors two and three are smooth throughout the graph, whereas the input to motor one contains more noisy. Inspecting the velocity estimates in the interval [10.5 < t < 11] reveals that the input to motor one is rapidly changing. This will be reflected into the calculation of the input, due to the feedback of the velocity. The problem to this could be due to poorly chosen design parameters for the filter, or the encoder problem described in Section *5.4: Simulation of linear Kalman*. However, due to time constraints this is not further investigated.

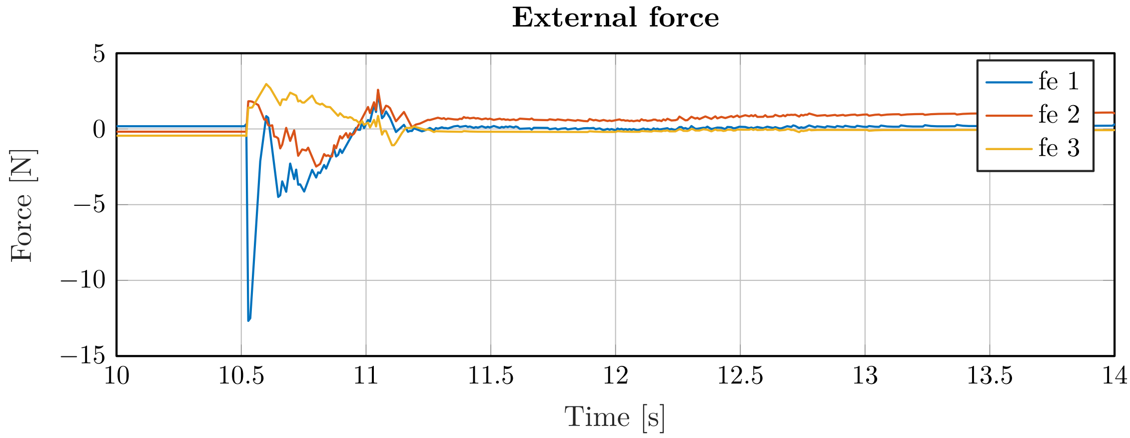

Figure 5.24: In the graph, the external force estimates made by the implemented linear Kalman filter is shown.

Figure 5.24 shows the external force estimate. When applying input to the system, a large force is estimated even though no external force is applied to the system. This is as expected and is discussed in the previous section Section *5.4: Simulation of linear Kalman*.

To sum up this section: the linear Kalman filter is implemented on the Teensy board and tested to see if it follows the trajectory of the system. It is concluded that the Kalman estimator follows the trajectory of the system, however smaller delays in the estimations of the velocity and acceleration are present. These are, however, less noisy in comparison to the numerical method of deriving these.

The next section tests the performance of collision detection. This is done, even though the model is concluded to be of poor quality, as it may still be possible to detect collision with the system.

6 Force estimation

This chapter is made for the test of the alternative observer described in Section *5.4: Alternative observer strategy.* The observer is implemented to the Teensy board described in Section *2.1.2: Micro controller.* The position measurements are given as inputs to the observer, where the position, velocity, acceleration and force estimates are returned.

The test is carried out in the following three steps:

1. The system is in a steady state position and force is applied to the end-effector.
   This test is to see how well the observer can estimate the external force applied in different directions to the end-effector.
2. An external force is applied to the system while moving.
   This test is to see if it is possible to detect a collision while moving the system.
3. An object is placed on the end-effector with a total force of 2.65 Newton.
   This test is made to see how fast the observer can detect an external force.

To limit the size of this section, only graphs with the external force estimates and in some cases position estimates are shown. However, all the data gathered under these tests can be found in the appendix, i.e position, velocity, acceleration, input and the force estimate. The reference for the specific appendix is giving in the introduction to the different tests, however, can all be found in *Appendix: E*.

Test one

The system is moved into the position seen in *Figure 6.1*, which equals an angular position of $[q_1 \ 0 \ 0]$, where q_1 is chosen arbitrary.

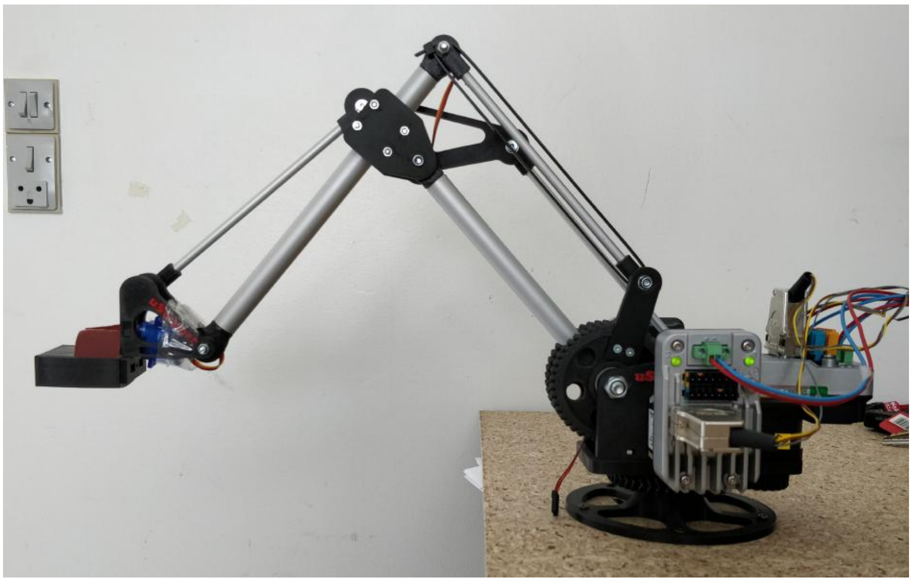

Figure 6.1: Initial position of the system before external force is applied to the system.

The applied external force is made by attaching a Newton-meter to the end-effector, and then stretching it out.
The available Newton-meter for these test is only capable of measuring a force up to 1.2 Newton. To ensure not exceeding the maximum of the meter, a force of 1.1 Newton is only applied to all the test within test one.

In *Figure 6.4* and *Figure 6.5* an external force is applied to the front and the back of the end-effector. To illustrate the direction of the force see *Figure 6.2* for force applied to the front and *Figure 6.3* for force applied to the back.
The additional data for this test can be found in *Appendix: E.3* for force applied to the front and *Appendix: E.4*, for force applied to the back.

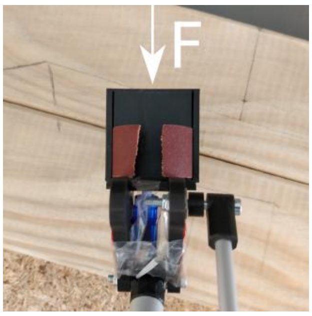

Figure 6.2: Force applied to the front of the end-effector.

Figure 6.3: Force applied to the back of the end-effector.

In *Figure 6.4* the estimates of the applied force to the front of the end-effector is shown. The force is applied at the interval [8.22 < t < 50.14]. It can be seen that the force estimate is not converging to a steady state position, however, increases to around one Newton. This is due to the fact that the applied force is made through a Newton-meter, which is held by a person. As the force is directly applied to the front of the end-effector, it was expected to see the largest force estimates being reflected to joint two and three i.e fe_2 and fe_3. However, in this case fe_1 and fe_2 are the largest. Since the force is

directly applied to the front of the end-effector the force fe_1 should not occur as the force is perpendicular to joint one. Therefore, it seems as there is some correlation between the force estimate of fe_1 and fe_2.
The largest increase in the force estimate is found to be

$$[\Delta fe_1 \quad \Delta fe_2 \quad \Delta fe_3] = [0.99 \quad 1.51 \quad 0.52] \tag{6.1}$$

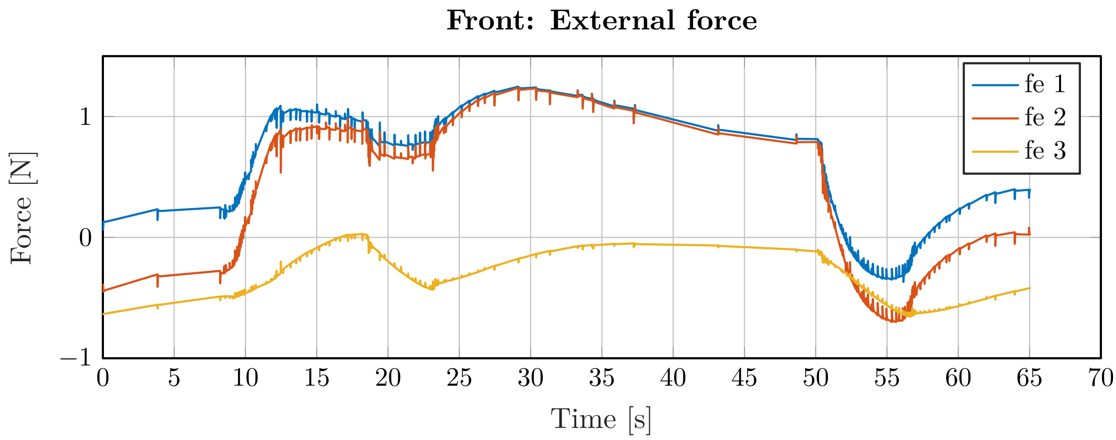

Figure 6.4: In the graph, the external force estimate for a force applied to the front of the end-effector is shown.

In *Figure 6.5*, the force is applied to the back of the end-effector in the time interval [7.04 $< t <$ 44.61]. Similar to the case where the force is applied to the front of the end-effector, the correlation between fe_1 and fe_2 is occurring.
The difference between the force estimate before applying the external force to the largest estimate is found to be

$$[\Delta fe_1 \quad \Delta fe_2 \quad \Delta fe_3] = [-1.17 \quad -1.19 \quad -0.18] \tag{6.2}$$

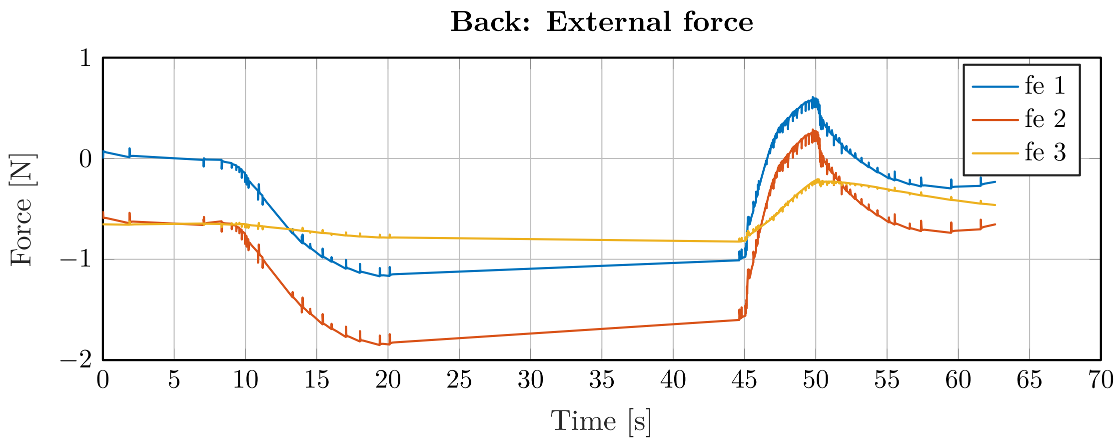

Figure 6.5: In the graph, the external force estimate for a force applied to the back of the end-effector is shown.

If *Figure 6.6* and *Figure 6.7* an illustration of force applied to the end-effector in an downwards and upwards direction, respectively, is shown. These are made for the reader to show how force is applied to the end-effector for the next two graphs.

The additional data for these tests can be found in *Appendix: E.2* for force applied in a downwards direction and *Appendix: E.1*, for force applied in an upwards direction.

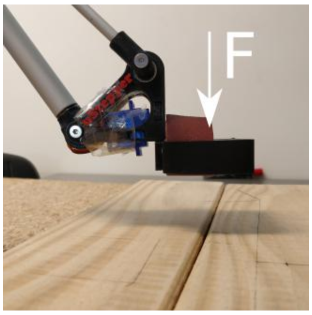

Figure 6.6: Force applied to the top of the end-effector, in a downwards direction.

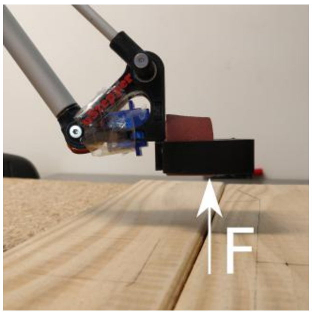

Figure 6.7: Force applied to the bottom of the end-effector, in an upwards direction.

In *Figure 6.8* the estimate of the external force applied to the end-effector in a downwards direction is shown.
It can be seen that the correlation between the force estimate of fe_1 and fe_2 still is occurring when force is applied to the system. The time interval for which the force is applied to the system is in the interval $[15.74 < t < 57.26]$. The highest increase of the force estimate, from the force being applied, is found to be

$$[\Delta fe_1 \quad \Delta fe_2 \quad \Delta fe_3] = [0.47 \quad -0.33 \quad 0.76] \tag{6.3}$$

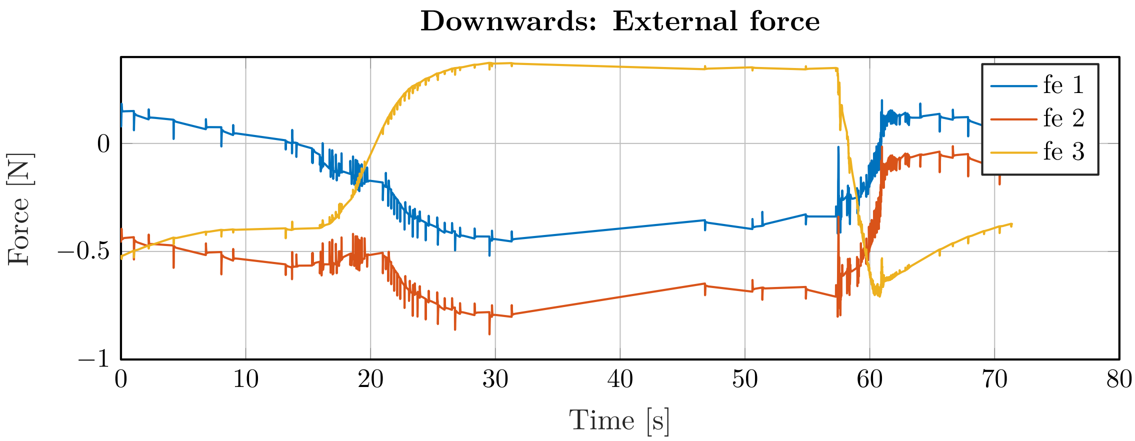

Figure 6.8: In the graph, the external force estimate for a force applied in a downwards direction to the end-effector is shown.

In *Figure 6.9*, the force is applied in an upwards direction.
It can be seen that the correlation between the force estimate fe_1 and fe_2 is still present in this test. The force is applied in the time interval $[17.42 < t < 52.85]$, where the maximum increase for fe_1 , fe_2 and fe_3 is found to be

$$[\Delta fe_1 \quad \Delta fe_2 \quad \Delta fe_3] = [0.65 \quad 0.78 \quad -0.89] \tag{6.4}$$

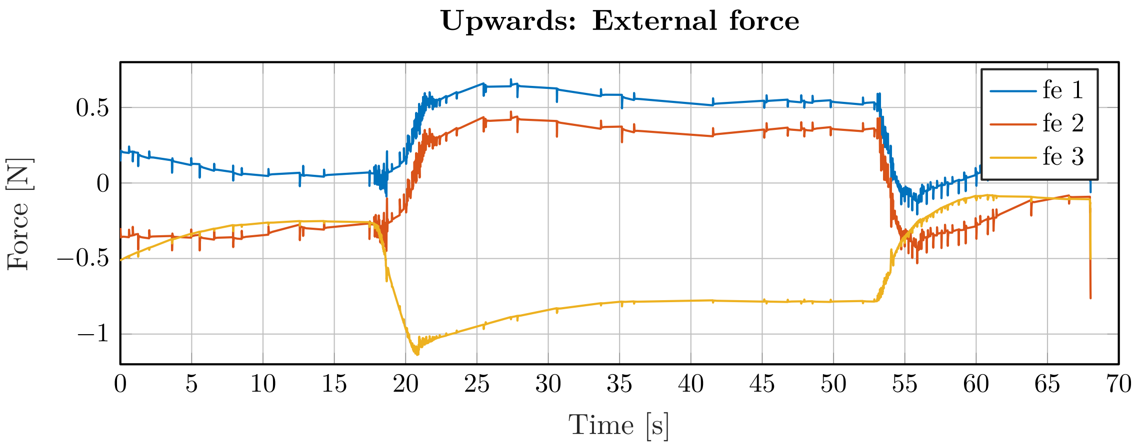

Figure 6.9: In the graph, the external force estimate for a force applied in a upwards direction to the end-effector is shown.

Conclusion test one: It is observed that there is a large correlation between the force estimates of fe_1 and fe_2. Inspecting the data in *Appendix: E.3*, it was noticed that the system input to motor two and the force estimations of fe_1 and fe_2 had a similar transient behavior. In the isolation of the external force, see *Equation: (5.15)*, the inverse Jacobian matrix of the end-effector is multiplied with input to the system. If the Jacobian is incorrectly derived, this could lead to the correlations seen in the force estimates.
With respect to the force estimations. It could be seen, when the system was exposed to external forces, that the observer reacted on these. However, none of the estimations converged close to the actual applied force. This could be due to imperfections in the model, which could be seen as the bias of the estimate, before applying a force.

Test two

In test two, the system is initialized at the position seen in *Figure 6.1*. Motor two is then giving a new reference, while motor one and three are set to keep an angular position of zero. Before actuating the system a blockage is set for the system, such that movement is only possible for a short angular distance before the blockage. The data for this test, can be found in *Appendix: E.6* for blockage in an upwards direction and *Appendix: E.7* for blockage in a downwards direction.

In *Figure 6.10* the position and force estimate data for the upwards direction is shown. The system is initialized to move at time 4.64, where after a blockage is introduced at time 4.84. The previous stated problem with the observer estimating a large external force under actuation is still present.
Before this test was made, it was assumed that the observer would make a large peak when a blockage was introduced. This is however not the case and it is not possible to disgusting between the actuating problem and the actual collision.
The increasing force estimate at the end of the test is due to the integral action in the controller, see *Appendix: E.6*, which tries to force the system to the reference point.
Furthermore it can be seen that the correlation between the force estimates fe_1 and fe_2 is gone, however correlations between fe_1 and fe_3 is present.

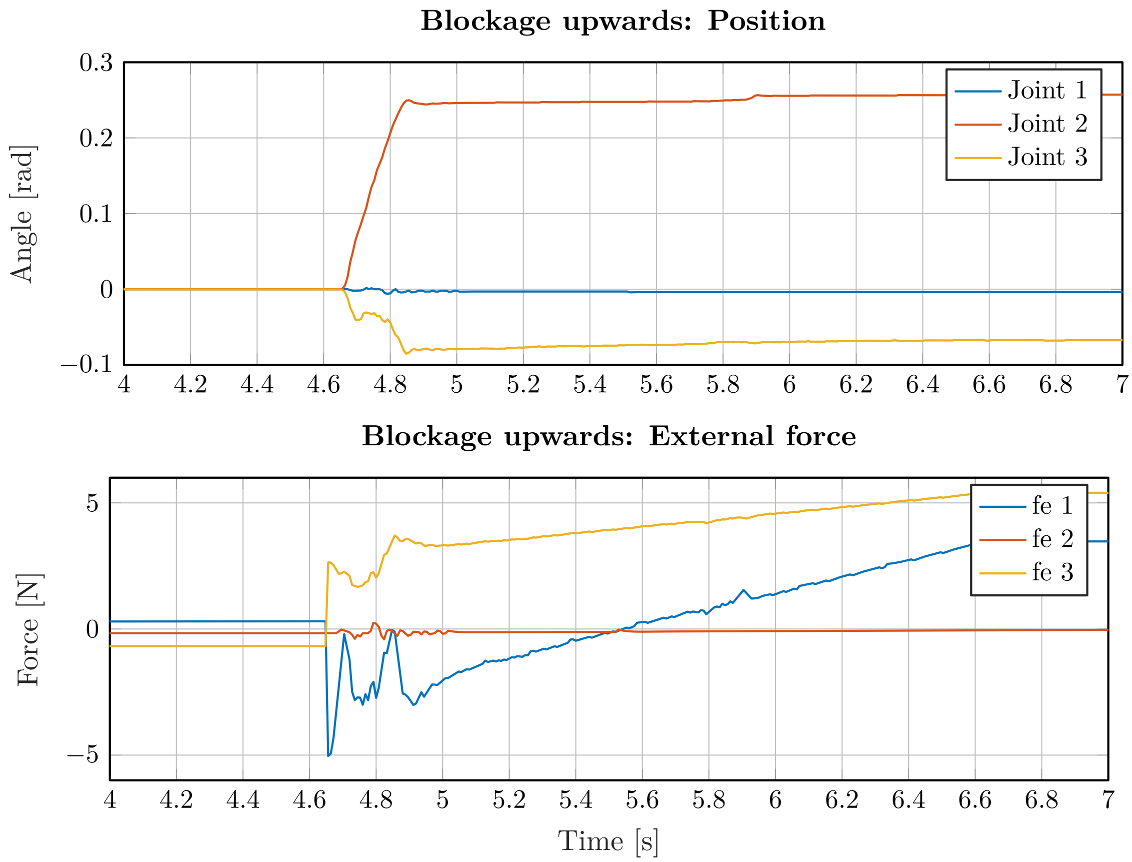

Figure 6.10: In the graphs, the angle and force estimates is shown for the system. The graphs shows actuation in an upwards direction for the system, for then being blocked by an external object.

In *Figure 6.11*, the position and force estimate data for the downwards direction is shown. The system is initialized to begin movement at time 2.64 where after a blockage is introduced at time 2.94. Similar to the upward case, it is not possible to isolate the collision from the actuation of the system.

One thing to be noted is however that the estimated force for fe_1 and fe_2 is no longer correlated as in the case of test one. However, this time it seems that fe_1 and fe_3 is somewhat correlated. Inspecting the data in *Appendix: E.7*, it is found that the input to motor one increases to a static value under actuation, while the input for motor three is slowly increasing due to the integral action of the controller. The transient between the force estimate fe_1 and the input to motor three has the same behavior and both becomes static when the input saturates.

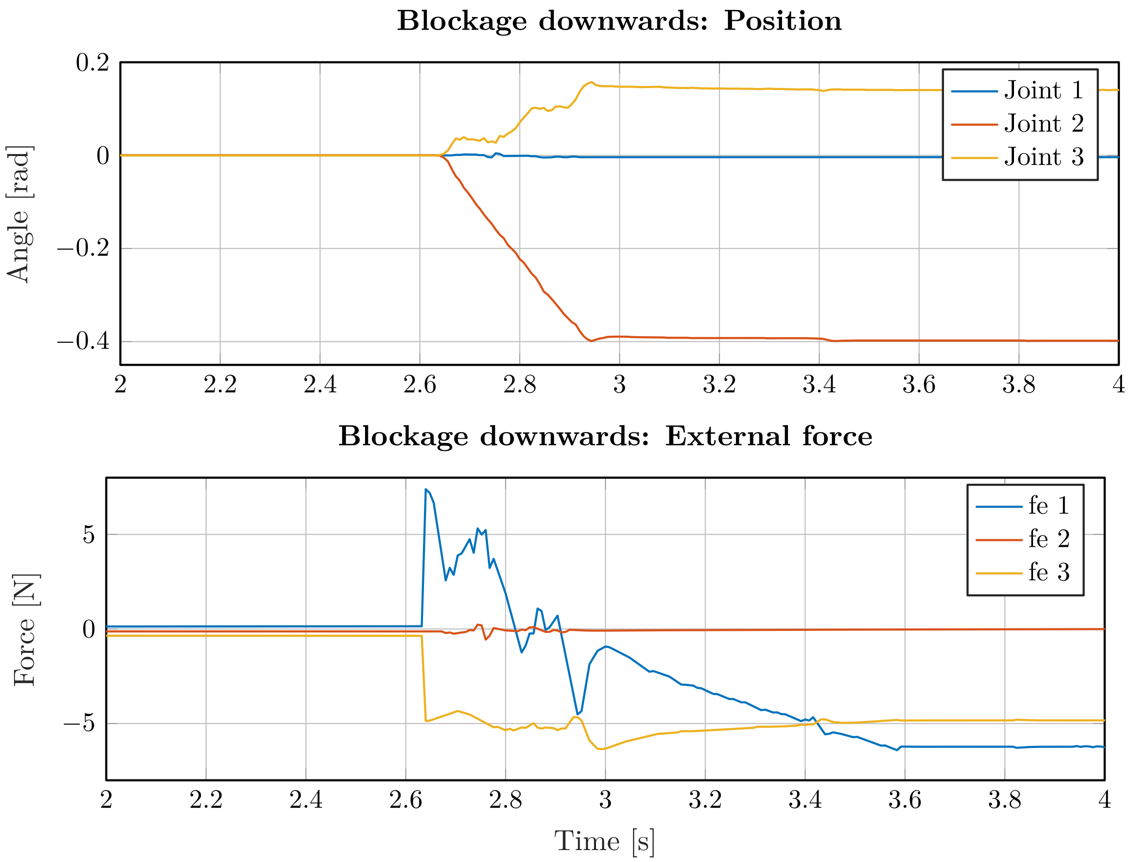

Figure 6.11: In the graphs, the angle and force estimates is shown for the system. The graphs shows actuation in an downwards direction for the system, for then being blocked by an external object.

In conclusion to test two: It is not possible to differentiate between the system being actuated or a collision is present for the system. This could be due to the estimation of the friction parameters found in Section *4.4.2: Second validation step*. If these do not capture the actual friction dynamic of the system, it would lead to an increase in the force estimations, thereby making it difficult to differentiate between collision and applied external forces.
With regards to the force estimation fe_1 it was observed that large correlations between the input to the system and this estimate, were correlated. This was also observed in test one. This could be related to the Jacobian matrix of the end-effector being derived wrongly and thereby making correlations between the inputs and force estimations.

Test three

This test is made to see how fast the observer is capable of observing an external collision. The system is brought to the initial state seen in *Figure 6.1*, where after an object of 270 grams is placed at the end-effector, which equals to a force of 2.65 N.
The data gathered for this test can be found in *Appendix: E.5*.

In *Figure 6.12* the external force estimate for test three is shown.
The object is placed on the end-effector at time 20.10, whereafter it can be seen that a large force estimate is made by the observer. The force estimates start reacting to the placed object after a period of eight milliseconds.
After a period of 100 milliseconds from the object being placed, the force estimates have

increased with

$$[\Delta fe_1 \quad \Delta fe_2 \quad \Delta fe_3] = [0.65 \quad 0.78 \quad -0.89] \tag{6.5}$$

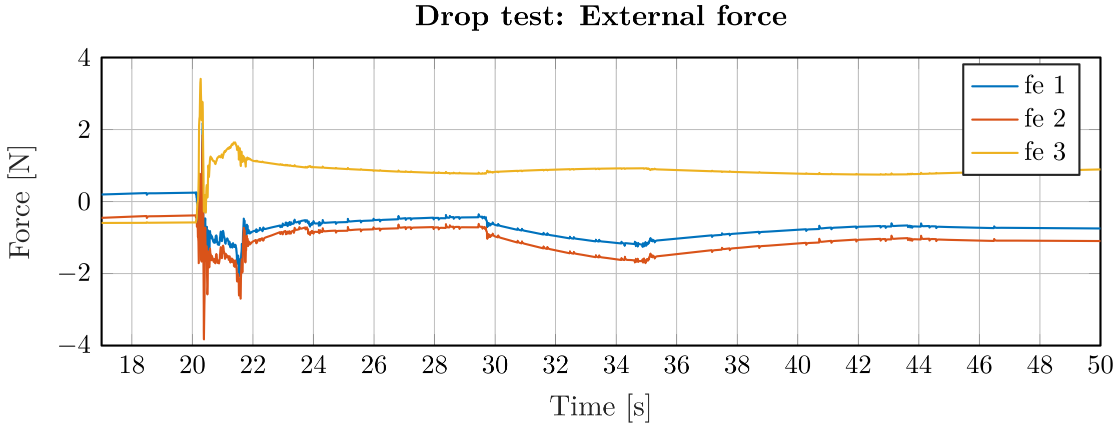

Figure 6.12: In the graph, the force estimates for an objected, equal to 2.65 Newton of force, being placed on the end-effector is shown.

The conclusion of test three is that the observer starts estimating the external force after a period of eight milliseconds.
With respect to the time requirement in Chapter *3: Project limitations and requirements*: Collision detection is usually threshold based, due to model imperfection and sensor noise. In this case, it is necessary to set the threshold with respect to the bias in the open loop estimate and within the increase of the estimates when the external force is applied. However, setting the threshold within the increase of the force estimates do not work in this case. This is due to problem lying in the actuation of the system, where the estimations increases above an external force of 5 N see *Figure 6.10* and *Figure 6.11*.

6.0.1 Conclusion of collision detection

This section was made to test the alternative observer strategy and see how well it performed. To summon up the three tests, it can be concluded that the observer reacts to the different external forces being applied to the system.
Unfortunately, the observer is deemed insufficient for application usage as it is not possible to isolate if the system only being actuated or colliding. Furthermore, it seems as if there is a large correlation between the force estimates and the inputs to the system. This could be a result of a wrongly derived Jacobian matrix for the end-effector, which is used in the open loop estimation of the external forces.

In test three it was measured that the observer starts reacting on the external force after a period of eight milliseconds. It is however not possible to verify if the observer fulfills the time requirement as a threshold needs to be found. The threshold should be found with respect to the bias of the estimator. As the force estimate is highly sensitive to the system being actuated, the threshold should be placed higher than the largest force estimate, while being actuated. However, test regarding larger forces than five Newtons have not been carried out. Therefore it is not possible to verify if a collision can be detected at higher forces.

Before introducing external forces to the system, it could be seen that the observer is biased in the estimates of the external forces. This could be due to the friction parameters found in Section *4.4.2: Second validation step*. As a result of this, the error in the external force estimate would increase.

Discussion 7

In the giving time of the project, a model based on Lagrangian dynamics was derived, with the purpose of deriving an extended Kalman filter for the system. Hand tuned parameters were found regarding the friction coefficients of the system and were tuned until a suiting response of the model was made. This gave promising results as it was possible to get an almost identical fit between model and measured data from the system. The model was used to derive an extended Kalman filter with the purpose of collision detection, which showed promising results regarding capturing the system dynamics and estimation of external forces in the simulation environment. Unfortunately, implementation of the extended Kalman filter to the system, showed poor results as the filter did not capture all system dynamics. This could be due to the estimation method regarding the friction parameters for the model which were hand tuned. Better approximations could be found through optimized identification algorithms, which could lead to a better fit of the model to the real system. The friction parameters were tuned such that the model gave a suiting response compared to the measurement gathered from the system. As these measurements were gathered over a short period of time, less dynamics of the system would be contained in these. Therefore it is proposed to redo the identification of friction parameters, however, this time based on longer measurements. This could lead to an increase of capturing system dynamics, for which the model could be further improved through identification of parameters.

An alternative observer method was derived with the purpose of detecting if a collision had occurred. This was based on a stationary linear Kalman filter and an open loop force estimator for the collision detection. This showed promising results regarding tracking of the states of the system, however, the force estimations were biased and highly influenced by actuation of the system. The linear Kalman filter was used for the state estimation of the system, where the estimates were given as input to the open loop force estimation. As the force estimation is made through an open loop, the possibility of it being biased increases as it requires that the model used is identical to the actual system, if not, a decrease in accuracy is to be expected.
The force estimators correlation to the system input should be investigated with the focus on solving this problem. Since the force estimator requires the inverse of the Jacobian matrix for the end-effector, and this is multiplied with the input vector, a guess could be a wrong implementation of the Jacobian or derivations of it.

Conclusion 8

In this project, the main focus has been on deriving a collision detection scheme for a robotic system with three degrees of freedom. The system available was a uStepper robot provided by Aalborg University.

A model has been derived from the method of Lagrangian dynamics.
Through drop test of the individual links and a velocity test, angular position data was gathered with the perspective of tuning friction parameters for the model. The friction parameters were hand tuned until the simulated model gave the same response as the gathered data. This approach led to promising results as it were possible to get an almost identical fit between the dynamic model and the gathered data.
With the found model, a PID controller with gravity compensation was designed. At the implementation of the controller, new control gains had to be found, due to an aggressive control of the system.

For estimating the states of the system, an extended Kalman filter was derived with the dynamic model. The extended Kalman filter is capable of tracking the derived model, and furthermore estimating when external forces were applied in the simulation environment. Unfortunately, at the implementation, the extended Kalman filter had to be discarded as it was unable to track all the states of the system. It was capable of estimating the angular positions and one of the joint velocities.
An alternative approach of the state estimations was derived based on a stationary linear Kalman filter and an open loop estimator for the estimation of external force. The estimated states were giving as input to the open loop estimation of the external force. This led to some promising result regarding tracking of the system states, however, the external force estimation showed large errors when the system is actuated. Furthermore, due to the open loop estimation of the force, the estimates were biased, when the system being in a static position.

Tests were carried out for different scenarios where external force was applied to the system. These were made for testing the performance of the external force estimates in different cases. When the system was in a static position and external forces were applied slowly to the system, the observer reacted to this and an increase in the force estimations was observed. This however also led to the observation that the force estimations was highly correlated to the input, even though no correlation should be seen.
When the system was actuated and a blockage for the system was introduced it was not possible to distinguish between the system being actuated or colliding.
The last test was to find the reaction time for a sudden force being applied to the system. This gave a result of eight milliseconds from the force being applied to the force estimate started increasing.
After the tests, it was decided that the observer strategy was not applicable for force estimations nor collision detection. This was based on the poor estimation of external forces when forces being applied to the system.

Bibliography

[1] B. Carlisle, 2017. Last visited 19-02-2018, http://www.robobusiness.com/about/event-news/pick-and-place-for-profit-using-robot-labor-to-save-money/.

[2] International Federation of Robotics, "Executive summary: World robotics 2017 industrial robots," 2017. Last visited 05-02-2018, https://ifr.org/downloads/press/Executive_Summary_WR_2017_Industrial_Robots.pdf.

[3] KUKA Roboter GmbH, Bachmann. Last visited 05-02-2018, https://commons.wikimedia.org/wiki/File:Factory_Automation_Robotics_Palettizing_Bread.jpg.

[4] F. Lamb, *Industrial automation: hands on*. McGraw Hill Professional, 2013.

[5] B. Matthias, "Risk assessment for human-robot collaborative applications," 2015. Last visited 05-02-2018, https://www.idiap.ch/~scalinon/IROS2015WS/BjoernMatthias-slides.pdf.

[6] KUKA Roboter GmbH, "Human-robot collaboration." Last visited 05-02-2018, https://www.kuka.com/en-de/technologies/human-robot-collaboration.

[7] N. Lauzier and C. Gosselin, "3-dof cartesian force limiting device based on the delta architecture for safe physical human-robot interaction," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 3420–3425, IEEE, 2010.

[8] L. Calderone, "Collaborative robots working in manufacturing," 2016. Last visited 06-02-2018, https://www.manufacturingtomorrow.com/article/2016/02/collaborative-robots-working-in-manufacturing/7672/.

[9] A. De Luca, A. Albu-Schaffer, S. Haddadin, and G. Hirzinger, "Collision detection and safe reaction with the dlr-iii lightweight manipulator arm," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pp. 1623–1630, IEEE, 2006.

[10] Bannerengineering. Last visited 13-02-2018, https://www.bannerengineering.com/us/en/products/machine-safety.html.

[11] "Computer vision based control of industrial robots." Last visited 14-02-2018, http://robotics.iitd.ac.in/ARL/?q=vision-based-robot-control.

[12] A. Owen-Hill, 2016. Last visited 13-02-2018, https://blog.robotiq.com/robot-vision-vs-computer-vision-whats-the-difference.

[13] D. Ebert, T. Komuro, A. Namiki, and M. Ishikawa, "Safe human-robot-coexistence: emergency-stop using a high-speed vision-chip," in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pp. 2923–2928, IEEE, 2005.

[14] M. Rashidan, Y. Mustafah, S. Hamid, Y. Shawgi, and N. Rashid, "Vision aided path planning for mobile robot," in *Computer and Communication Engineering (ICCCE), 2014 International Conference on*, pp. 5–8, IEEE, 2014.

[15] G. Hirzinger, A. Albu-Schaffer, M. Hahnle, I. Schaefer, and N. Sporer, "On a new generation of torque controlled light-weight robots," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 4, pp. 3356–3363, IEEE, 2001.

[16] "Compliant robots." Last visited 16-02-2018, http://www.eucognition.org/eucog-wiki/Compliant_robots.

[17] M. Zinn, O. Khatib, B. Roth, and J. K. Salisbury, "Playing it safe [human-friendly robots]," *IEEE Robotics & Automation Magazine*, vol. 11, no. 2, pp. 12–21, 2004.

[18] A. Bicchi and G. Tonietti, "Fast and" soft-arm" tactics [robot arm design]," *IEEE Robotics & Automation Magazine*, vol. 11, no. 2, pp. 22–33, 2004.

[19] A. Garcia, V. Feliu, and J. Somolinos, "Experimental testing of a gauge based collision detection mechanism for a new three-degree-of-freedom flexible robot," *Journal of Field Robotics*, vol. 20, no. 6, pp. 271–284, 2003.

[20] V. J. Lumelsky and E. Cheung, "Real-time collision avoidance in teleoperated whole-sensitive robot arm manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 1, pp. 194–203, 1993.

[21] S. Takakura, T. Murakami, and K. Ohnishi, "An approach to collision detection and recovery motion in industrial robot," in *Industrial Electronics Society, 1989. IECON'89., 15th Annual Conference of IEEE*, pp. 421–426, IEEE, 1989.

[22] K. Suita, Y. Yamada, N. Tsuchida, K. Imai, H. Ikeda, and N. Sugimoto, "A failure-to-safety" kyozon" system with simple contact detection and stop capabilities for safe human-autonomous robot coexistence," in *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, vol. 3, pp. 3089–3096, IEEE, 1995.

[23] A. De Luca and R. Mattone, "Sensorless robot collision detection and hybrid force/motion control," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pp. 999–1004, IEEE, 2005.

[24] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger, "Requirements for safe robots: Measurements, analysis and new insights," *The International Journal of Robotics Research*, vol. 28, no. 11-12, pp. 1507–1527, 2009.

[25] "Ustepper robot arm." Last visited 4-06-2018, http://www.instructables.com/id/Robot-Arm-UStepper/.

[26] "Lm42 nema 17 canopen ip20." Last visited 19-02-2018, https://motion.schneider-electric.com/lexium-mdrive-produc/lm42-nema-17-canopen-ip20-2/.

[27] Autom, "About motor industry," 2012. Last visited 20-02-2018, hhttp://autom.cn/News_view_38.html.

[28] Motion Schneider Electric, "Lm42 nema 17 canopen ip20." Last visited 20-02-2018, https://motion.schneider-electric.com/lexium-mdrive-produc/lm42-nema-17-canopen-ip20-2/#1483118489716-9041bf69-f156.

[29] B.-B. Mathieu, "Iso/ts 15066 and collaborative robot safety," vol. 63, pp. 20–23, 2016.

[30] Siciliano, B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics*. 2009.

[31] B. Carlisle, 1998. Last visited 5-06-2018, http://web.mit.edu/10.001/Web/Course_Notes/Differential_Equations_Notes/node3.html.

[32] Schneider Electric, "Simplifying machine building with compact integrated motors." Last visited 05-02-2018, https://motion.schneider-electric.com/lmd/downloads/literature/LMD_ethernet_datasheet.pdf.

[33] "Generate c and c++ code from matlab code." Last visited 19-05-2018, https://se.mathworks.com/products/matlab-coder.html.

[34] "Kalman." Last visited 30-05-2018, https://se.mathworks.com/help/control/ref/kalman.html.

Part I

Appendices

Measurements A

In this chapter an elaboration of how measurements for model verification and parameter estimation is giving.

Test equipment

- uStepper robot described in Section *2.1: Physical structure*,
- Teensy 3.6,
- Computer.

Procedure

The following three procedures for gathering the data was made and the initial states for the link positions can be seen in *Figure A.3*. The initial set up for procedure one and three are the same.

Procedure 1

1. Lock joint 2 such that link 2 is in a vertical position with respect to ground,
2. Lift Link 4 to a position where a 90 degree angle between link 2 and 4 is made and lock joint 3,
3. Accelerate joint 1 until an angular velocity of 2.56 [rad/s] is reached,
4. Sample position and time,
5. Transfer data to computer.
6. Repeat five times.

Procedure 2

1. Lock joint 2 such that link 2 is in a 73 degree position with respect to ground,
2. Lift Link 4 to a position where a 90 degree angle between link 2 and 4 is made and lock joint 3,
3. Unlock joint 2 and sample position and time,
4. Transfer to computer.
5. Repeat five times.

Procedure 3

1. Lock joint 2 such that link 2 is in a vertical position with respect to ground,
2. Lift Link 4 to a position where a 90 degree angle between link 2 and 4 is made and lock joint 3,
3. Unlock joint 3 and sample position and time,

4. Transfer to computer.
5. Repeat five times.

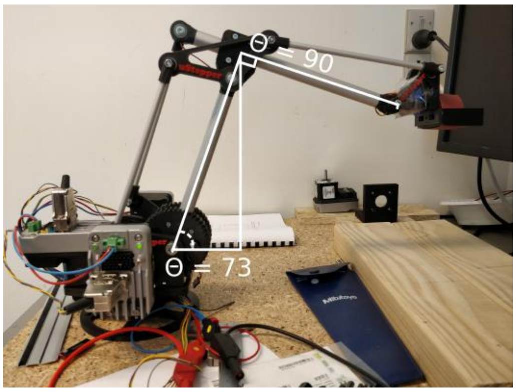

Figure A.1: Initial state of procedure two.

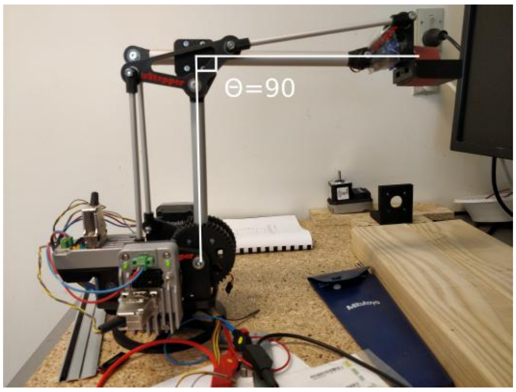

Figure A.2: Initial state of procedure one and three.

Figure A.3: Initial states for the procedures.

Data

The test data can be seen in *Figure A.4*, *Figure A.5* and *Figure A.6* and found on the attached files [Dir: Attachments/Test measurements /Position data].

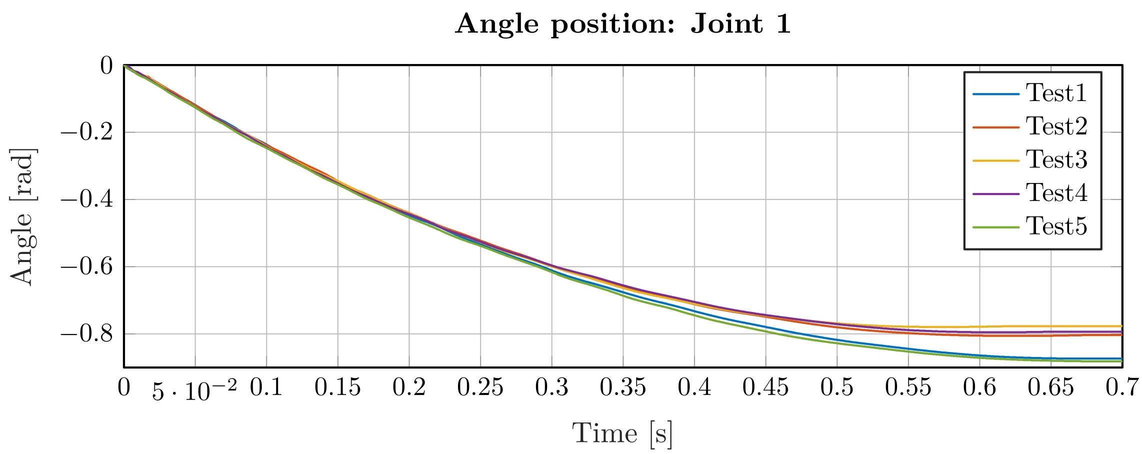

Figure A.4: Measurement from procedure one.

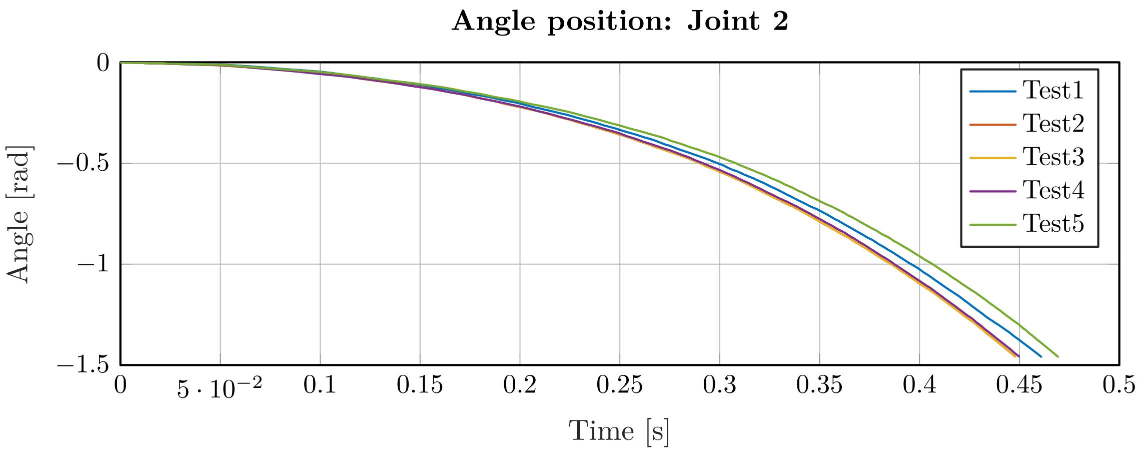

Figure A.5: Measurements from procedure two.

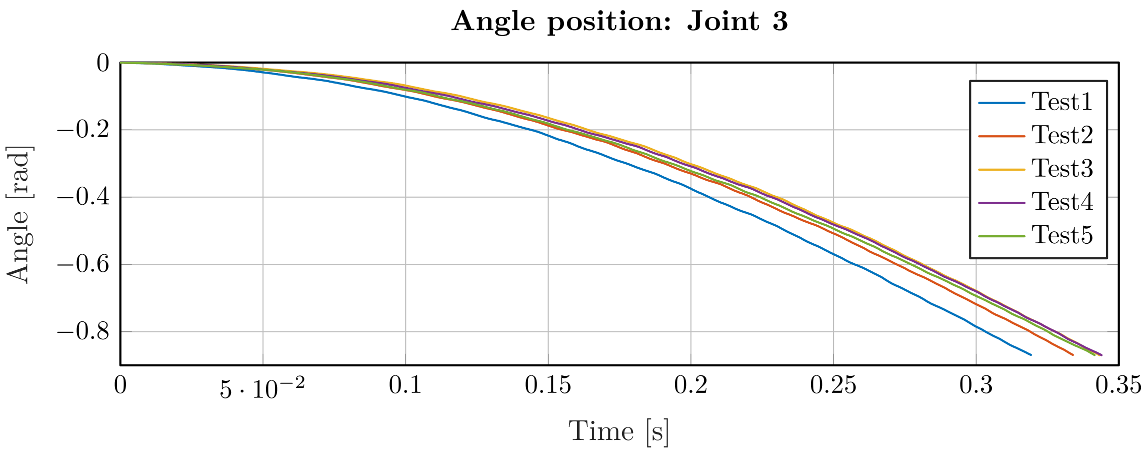

Figure A.6: Measurement from procedure three.

Uncertainties

- Initial stats of uSteppers joints/link may vary ±5 degrees between measurements.
- Varying time delay between samples.

Link weight and friction B

This chapter includes the weights and lengths of the different links of the uStepper and the end-effector.

Link number	Weight [kg]	Length [m]
1	2.7	0.06
2	0.053	0.18
3 / 4	0.091	0.03 / 0.17
End-effector	0.048	0.0475

Table B.1: The links lengths and weights. Link three and four is seen as one connected link.

Joint number	Viscous friction	Coulomb friction
1	0.0001	0.023
2	0.008	0.206
3	0.0001	0.139

Table B.2: The hand tuned friction components.

Kinetic and potential energy C

This chapter is made for the derivation of the kinetic and potential energy contribution for a link of the uStepper. It is assumed that the mass of each link is uniform distributed over the entire length L. Furthermore a length density ρ is defined as the mass over the length of the link. Lower case l on *Figure C.1* defines the integration length of interest and is bounded by $[0 \leq l \leq L]$. As the total contribution of each link is of interest the integration is over the entire length L.

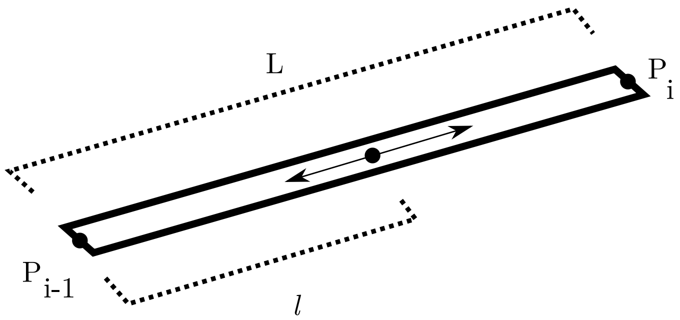

Figure C.1: Illustration of a link and its contribution of kinetic energy over the length l of the total length L of the link. P_i is the position of joint i and P_{i-1} the previous joint position.

The point on *Figure C.1* can be found as seen in *Equation: (C.1)*

$$P_l = P_{i-1} + (P_i - P_{i-1})\frac{1}{L} \cdot l \tag{C.1}$$

The kinetic energy of an object can be found as seen in *Equation: (C.2)*,

$$T = \frac{1}{2} \cdot m \cdot v^2 \tag{C.2}$$

Where

	m	is the mass of the object,	[kg]
and	v	is the velocity.	$[\frac{\mathrm{m}}{\mathrm{s}}]$

To find the velocity of the point mass of the link, see *Figure C.1*, the time derivative of *Equation: (C.1)* is made and can be seen in *Equation: (C.3)*

$$\dot{P}_l = \dot{P}_{i-1} + (\dot{P}_i - \dot{P}_{i-1})\frac{1}{L} \cdot l \tag{C.3}$$

In equation *Equation: (C.4)*, the expression for the kinetic energy contribution of a link on the uStepper is derived.

$$
\begin{aligned}
t_i &= \rho\frac{1}{2}\cdot\int_0^L [\dot{P}_{i-1} + \frac{1}{L}(\dot{P}_i - \dot{P}_{i-1})\cdot l]^T[\dot{P}_{i-1} + \frac{1}{L}(\dot{P}_i - \dot{P}_{i-1})\cdot l]\mathrm{d}l \\
&= \rho\frac{1}{2}\cdot\int_0^L \dot{P}_{i-1}^T\dot{P}_{i-1} + \left[\frac{1}{L}[\dot{P}_i - \dot{P}_{i-1}]\cdot l\right]^T\left[\frac{1}{L}[\dot{P}_i - \dot{P}_{i-1}]\cdot l\right] + 2\frac{1}{L}\dot{P}_{i-1}^T[\dot{P}_i - \dot{P}_{i-1}]\cdot l\mathrm{d}l \\
&= \rho\frac{1}{2}\cdot\int_0^L \dot{P}_{i-1}^T\dot{P}_{i-1} + \frac{1}{L^2}l^2[\dot{P}_i^T\dot{P}_i + \dot{P}_{i-1}^T\dot{P}_{i-1} - 2\dot{P}_i^T\dot{P}_{i-1}] + \frac{2}{L}l\dot{P}_{i-1}^T[\dot{P}_i - \dot{P}_{i-1}]\mathrm{d}l \\
&= \rho\frac{1}{2}\left(\dot{P}_{i-1}^T\dot{P}_{i-1}L + \frac{1}{3L^2}L^3[\dot{P}_i^T\dot{P}_i + \dot{P}_{i-1}^T\dot{P}_{i-1} - 2\dot{P}_i^T\dot{P}_{i-1}] + \frac{1}{L}L^2\dot{P}_{i-1}^T[\dot{P}_i - \dot{P}_{i-1}]\right) \\
&= m_i\frac{1}{2}\left(\dot{P}_{i-1}^T\dot{P}_{i-1} + \frac{1}{3}[\dot{P}_i^T\dot{P}_i + \dot{P}_{i-1}^T\dot{P}_{i-1} - 2\dot{P}_i^T\dot{P}_{i-1}] + \dot{P}_{i-1}^T[\dot{P}_i - \dot{P}_{i-1}]\right) \\
&= m_i\frac{1}{2}\left(\dot{P}_{i-1}^T\dot{P}_{i-1} + \frac{1}{3}\dot{P}_i^T\dot{P}_i + \frac{1}{3}\dot{P}_{i-1}^T\dot{P}_{i-1} - \frac{2}{3}\dot{P}_i^T\dot{P}_{i-1} + \dot{P}_{i-1}^T\dot{P}_i - \dot{P}_{i-1}^T\dot{P}_{i-1}\right) \\
&= m_i\frac{1}{2}\left(\frac{1}{3}\dot{P}_i^T\dot{P}_i + \frac{1}{3}\dot{P}_{i-1}^T\dot{P}_{i-1} + \frac{1}{3}\dot{P}_{i-1}^T\dot{P}_i\right) \\
&= m_i\frac{1}{6}\left(\dot{P}_{i-1}^T\dot{P}_{i-1} + \dot{P}_i^T\dot{P}_i + \dot{P}_{i-1}^T\dot{P}_i\right)
\end{aligned}
\tag{C.4}
$$

Where

t_i	is the kinetic energy contribution of link i	$\left[\frac{\mathrm{kg\cdot m^2}}{\mathrm{s^2}}\right]$
$\dot{P}_i$	is the position of joint i,	$\left[[\mathrm{x,y,z}]^\mathrm{T}\mathrm{m}\right]$
ρ	is the length density of the link,	$\left[\frac{\mathrm{kg}}{\mathrm{L}}\right]$
m_i	is the link mass,	$[\mathrm{kg}]$
l	is the integration length of interest,	$[\mathrm{m}]$
and L	is the total Length of the link.	$[\mathrm{m}]$

In *Equation: (C.5)* the expression for the potential energy contribution of each link is derived. The vector $[0\ 0\ 1]^T$ is done as the hight of the element is of interest.

$$
\begin{aligned}
p_i &= \rho g\int_0^L \left[P_{i-1} + l\frac{1}{L}[P_i - P_{i-1}]\right]^T \begin{bmatrix}0\\0\\1\end{bmatrix}\mathrm{d}l \\
&= \rho g\left[P_{i-1}L + L^2\frac{1}{2L}[P_i - P_{i-1}]\right]^T\begin{bmatrix}0\\0\\1\end{bmatrix} \\
&= mg\left[P_{i-1} + \frac{1}{2}[P_i - P_{i-1}]\right]^T\begin{bmatrix}0\\0\\1\end{bmatrix} \\
&= mg\frac{1}{2}\left[P_{i-1} + P_i\right]^T\begin{bmatrix}0\\0\\1\end{bmatrix}
\end{aligned}
\tag{C.5}
$$

Where

g	is the gravitational acceleration,	$\left[\frac{\mathrm{m}}{\mathrm{s^2}}\right]$
and p_i	is the potential energy contribution of link i.	$\left[\frac{\mathrm{kg\cdot m^2}}{\mathrm{s^2}}\right]$

Removal of velocity limite D

This chapter is made for the testing of the impact the limit of the velocity has on the external force estimation. Unfortunately, before the gathering of data, the encoder for motor/joint one got damaged, which resulted in unreliable data readings when torque is applied to motor one. Therefore, the only actuation is made for motor two and three. The measurement for joint one is however still shown in this chapter, as small changes is showing, when actuation the other joints.
The tests are initialized in a position of [0 0 0] rad for all joints. The system is then giving a reference of [0 0.8 -0.2].

Low velocity limit

In *Figure D.1* to *Figure D.4*, the testing of the initial velocity limit of 5.72 [rad/s] for the motors is shown.

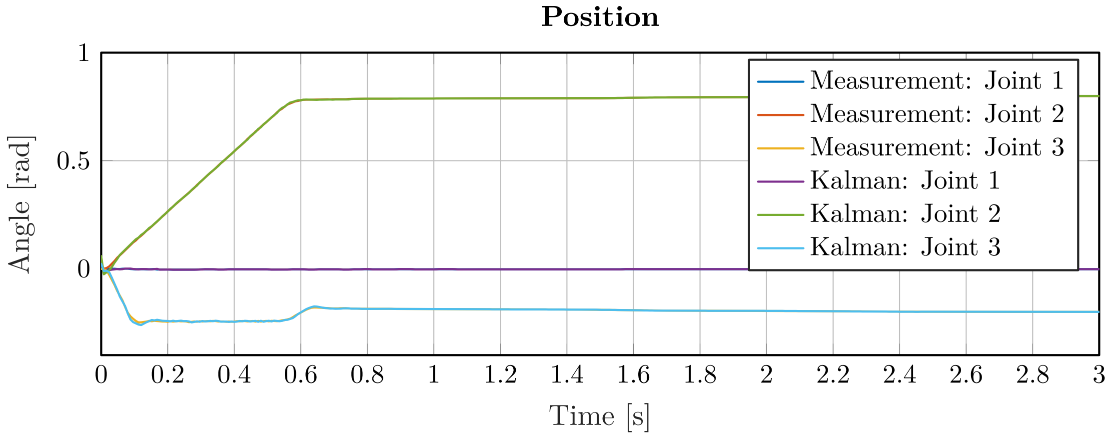

Figure D.1: The gathered position data with the Kalman estimates of the positions.

In *Figure D.1* the position estimates and measurements are shown. It can be seen that the estimation and measurements are on top of each other. The system has reached steady state position within three seconds.

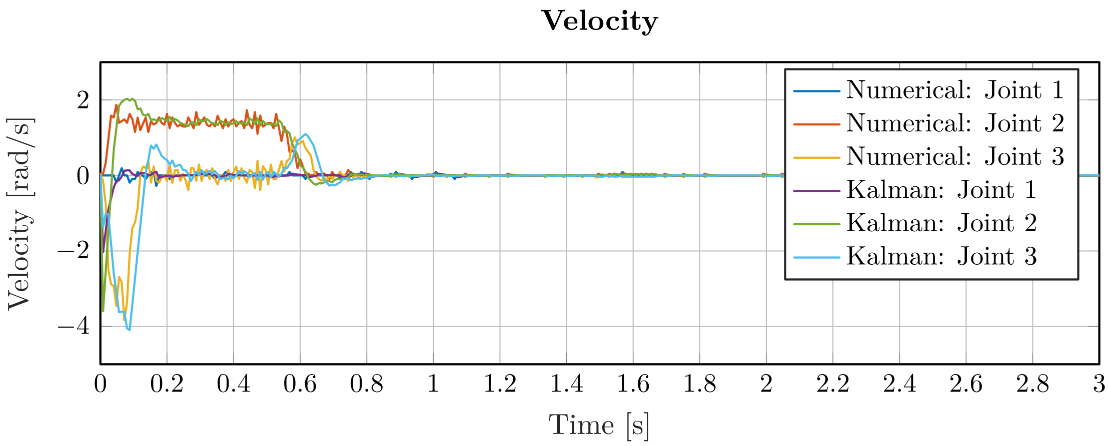

Figure D.2: This graph shows the velocity estimates for the system. The graph includes two estimates methods, one being the numerical and the other the Kalman estimates.

In *Figure D.2* the velocity estimations are shown. It can be seen that the Kalman estimates follows the transient of the numerical derived velocity, however a small overshoot can be seen.

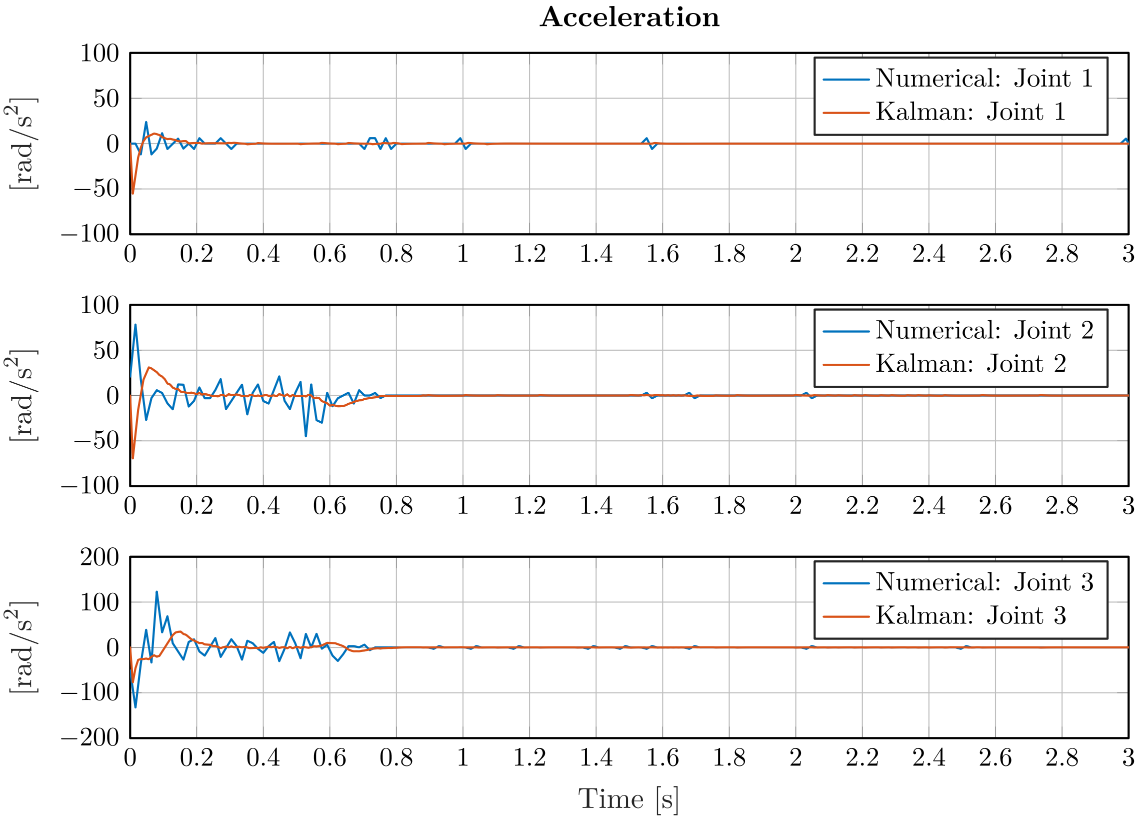

Figure D.3: In this graph the estimations of the acceleration of the system is estimated. The graph includes the two estimations methods, one being the numerical and the other Kalman estimates.

In *Figure D.3* the acceleration of the joints can be seen. It can be seen that the Kalman filter follows the numerical derived acceleration, however with a larger delay in comparison to the velocity. This could be solved by choosing another and faster pole location for the

Kalman filter. It can be seen that the acceleration for joint one is not at zero at all time,

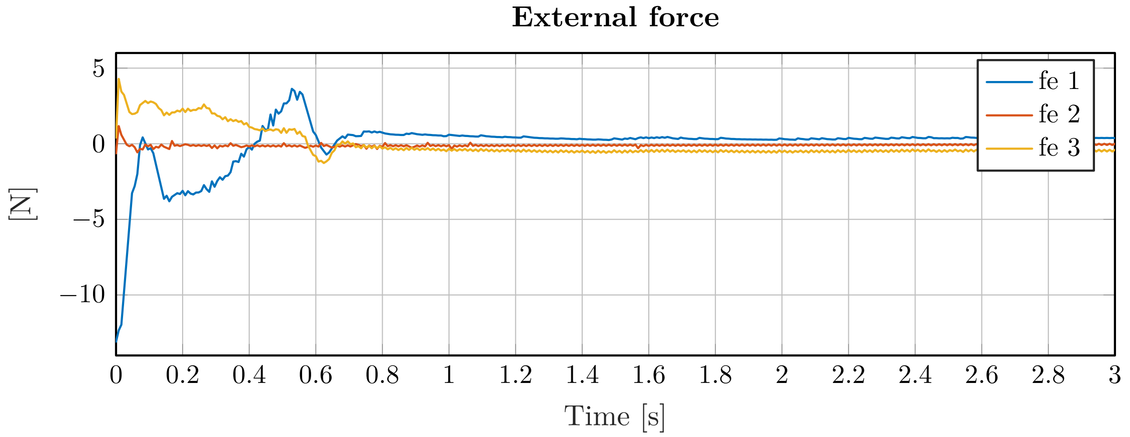

Figure D.4: In this graph the force estimations based open loop observer is shown.

In *Figure D.4* the force estimations is shown. It can be seen when the system is initialized with a large force estimate, even though no external force is applied to the system.

Increased velocity limit

In *Figure D.5* to *Figure D.8* the data for the increased velocity limit is shown. The limit is increased to a maximum velocity of 14.72 [rad/s]. The implemented controller is not re-tuned due to time constraints on the project. Therefore a performance degradation is to be expected.

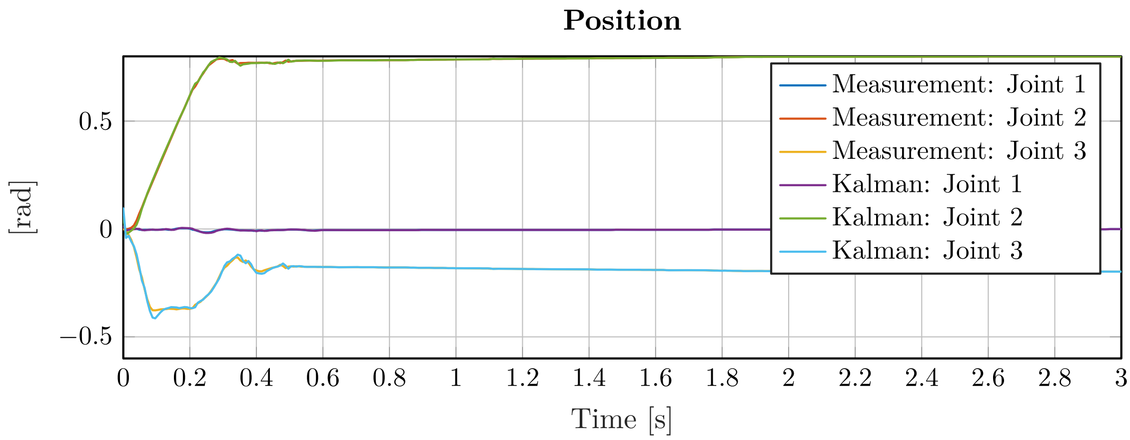

Figure D.5: In this figure the position measurements and Kalman estimates are shown.

In *Figure D.5* the position estimates and measurements are shown. Different from *Figure D.1* it can be seen that the position transaction is not as smooth when reaching the reference. This is probably due to the controller not being tuned to the increased velocity limit. It can further be seen that the increased velocity limit, makes the system reach the reference at half the time of the lower limit.

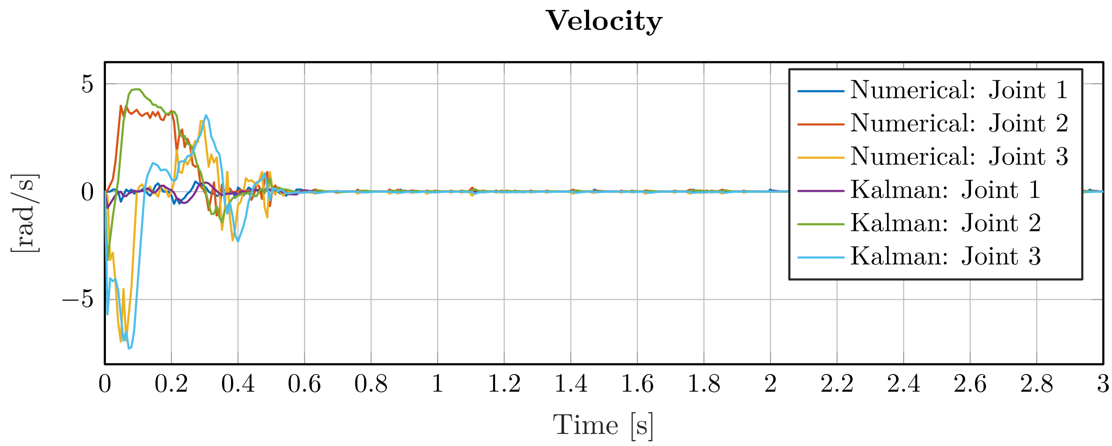

Figure D.6: This graph shows the velocity estimates for the system. The graph includes two estimates methods, one being the numerical and the other the Kalman estimates.

In *Figure D.6* the numerical and Kalman estimates for the system is shown. It can be seen that the Kalman filter still is capable of following the transient of the numerical calculated velocity.

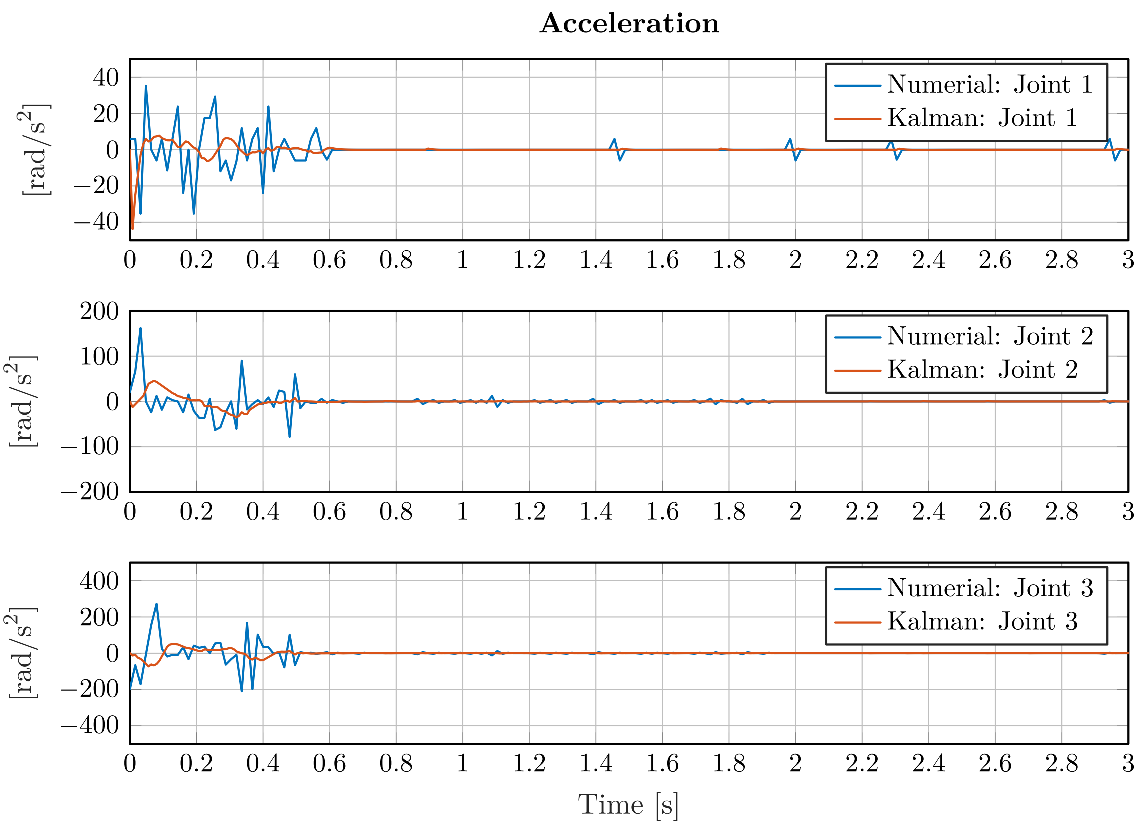

Figure D.7: In this graph the estimations of the acceleration of the system is estimated. The graph includes the two estimations methods, one being the numerical and the other Kalman estimates.

In *Figure D.7* the acceleration of the system is shown. It can be seen that the delay in the estimations of the acceleration still is present as in the case with the lower limit.

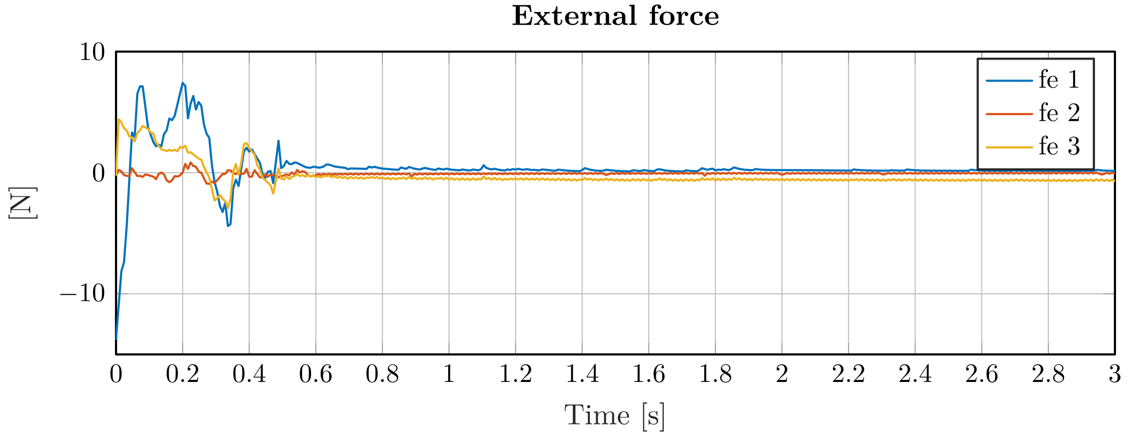

Figure D.8: In this graph the force estimations based open loop observer is shown.

In *Figure D.8* the external force estimation is shown. It can be seen that the estimates still shows large estimations of external force, when the system is initialized.

Conclusion

In *Figure D.9* the force estimation for both the high and low velocity limit is shown.

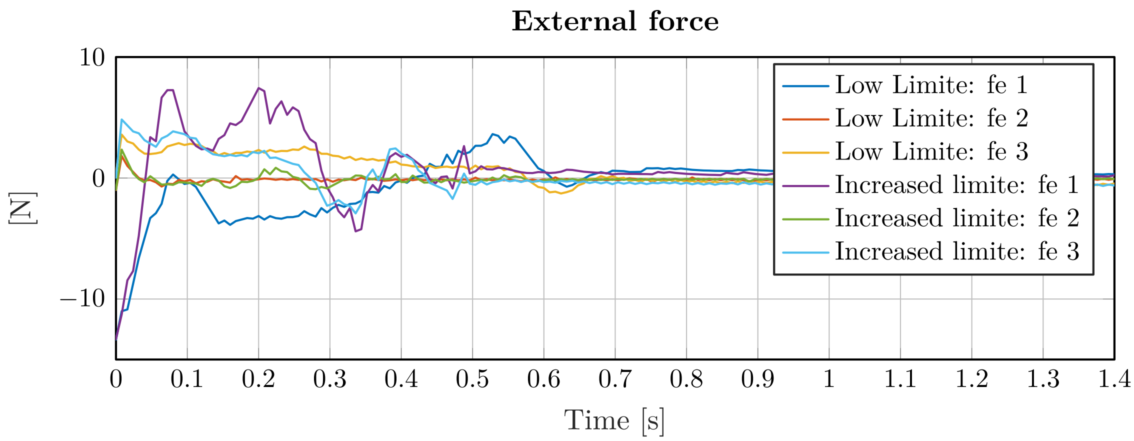

Figure D.9: In this graph the force estimations is shown. It is made for a simpler comparison of the two estimations for where the velocity limits are set.

It can be seen that increasing the velocity limit do not improve the external force estimation, but rather decrease the performance of it. One should notice that the force estimate, with regards to joint one, all ways is present even though this joint is not actuated. This could be due to some correlation between the different joint in the system, from which the model is to be discussed.
Since the first joint is rotated 90 degrees in respect to joint two and three, only a small correlation should be present due to the small off symmetric in the structure of the physical system.

External force test E

In this chapter the measurements for the testing of external force applied to the end-effector is gartered.
For the sake of the reader, this chapter has been split up over multiple pages, such that all graphs for the individual test are gathered at one and the same page.

The measurements and estimations for the different force estimation tests is shown in the following order:

1. Force applied in an upwards direction to the end-effector, [p. 76].
2. Force applied in an downwards direction to the end-effector, [p. 78].
3. Force applied to the front of the end-effector, [p. 80].
4. Force applied to the back of the end-effector, [p. 82].
5. Force impulse to the end-effector, [p. 84].
6. Blockage in upwards directions, [p. 86].
7. Blockage in downwards directions, [p. 88].

The force estimating, denote fe_1, fe_2 and fe_3, corresponds to the external force in the global Cartesian space. These estimates are based on the reflected force from Cartesian space to the torque in the joint space of the system.

E.1 External force applied in an upwards direction

In *Figure E.1* the data gathered while applying a force in an upwards direction to the end-effector is shown. The force is applied by attaching a Newton meter to the end-effector, which is being extended to a force of 1.1 N. The force applied in this test is not static as force is made by human hand.
The time steps for which the external force is applied is in the interval [17.42 < t < 52.85]. It can be seen that the position for joint one is static in the interval where the force is applied, however this is not the same for the positions of joint two and three. Therefore it can be said that the force is being reflected to joint two and three, however the force estimation for fe_1 still reacts on the applied force. Comparing the force estimate fe_1 and fe_2 there seems to be a correlation between the two estimates.
From the force being applied to the maximum estimated force an increase for the different force estimates is measured to:

- Δfe_1 = 0.646 N
- Δfe_2 = 0.781 N
- Δfe_3 = -0.888 N

whereafter somewhat converges towards

- fe_1 = 0.538 N
- fe_2 = 0.389 N
- fe_3 = -0.777 N

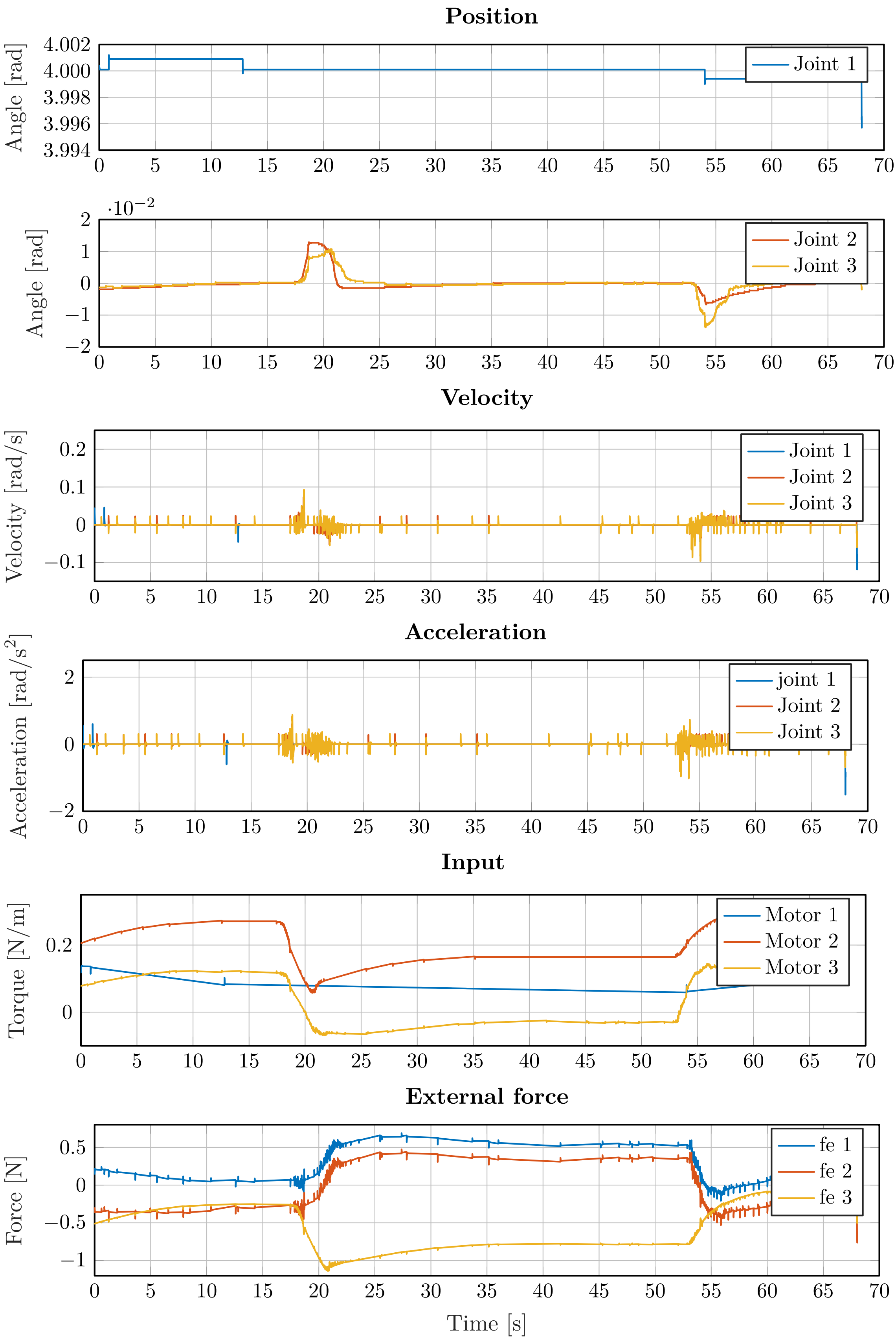

Figure E.1: This figure shows estimation data for the uStepper robot while an external force in an upwards direction is being applied to the end-effector. One and two (position), three (velocity), four (acceleration), five (input) and six (force).

E.2 External force applied in an downwards direction

In *Figure E.2* the data gathered while applying a force in an downwards direction to the end-effector is shown. The force is applied by attaching a Newton meter to the end-effector, which is being extended to a force of 1.1 N. The force applied in this test is not static as force is made by human hand.
The time interval for which the force is applied is [15.74 < t < 57.26]. It can be seen that the position of joint one is moved a bit under these tests, which could be due human error i.e the force is not directly orthogonal to the top of the end-effector. Similar to the test, where the force is applied in an upwards direction, see *Appendix: E.1*, it can be seen that the force estimates fe_1 and fe_2 have some correlations.
From the force being applied to the maximum estimated force an increase for the different estimates is measured to:

- Δfe_1 = 0.473 N
- Δfe_2 = -0.333 N
- Δfe_3 = 0.763 N

whereafter somewhat converges towards

- fe_1 = 0.538 N
- fe_2 = 0.389 N
- fe_3 = -0.777 N

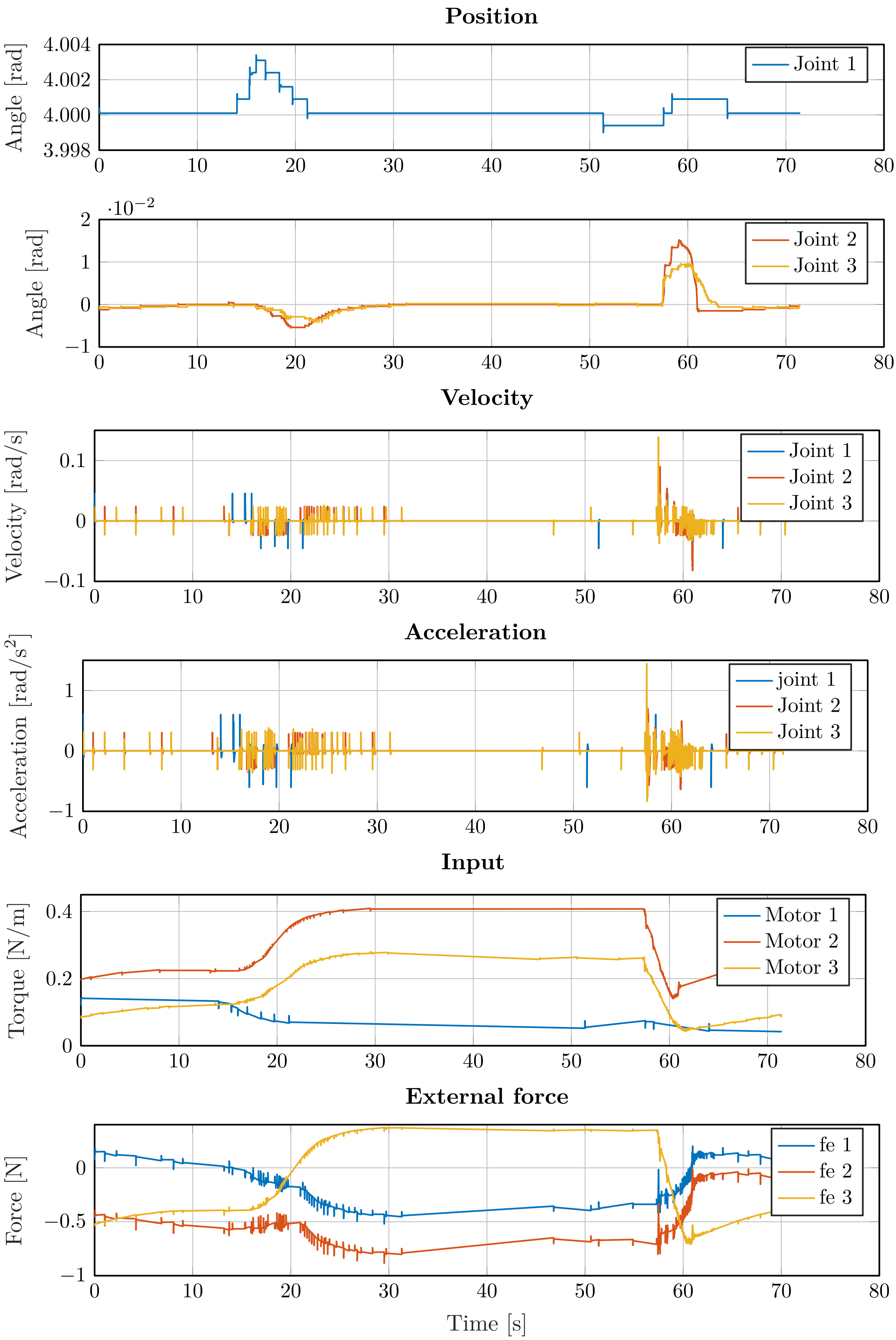

Figure E.2: This figure shows estimation data for the uStepper robot while an external force in an downward direction is being applied to the end-effector. One and two (position), three (velocity), four (acceleration), five (input) and six (force).

E.3 External force applied to the front of the end-effector

In *Figure E.3* the data gathered while applying a force to the front of the end-effector is shown. The force is applied by attaching a Newton meter to the end-effector, which is being extended to a force of 1.1 N. The force applied in this test is not static as force is made by human hand.
The time span in which the force is being applied is [8.22 < t < 50.14]. It can be seen when the force is being applied, that all angular positions are moving. However, the largest position difference is seen for joint two and three. Therefore, it can be said that most of the force is reflected to these.
Even though, the largest difference is seen for joint two and three, the force estimates have the largest increase for fe_1 and fe_2. It can also be observed that there is some correlation between these, as the transient is almost identical.
From the force being applied to the maximum estimated force an increase for the different estimates is measured to:

- Δfe_1 = 0.999 N
- Δfe_2 = 1,51 N
- Δfe_3 = 0.518 N

whereafter somewhat converges towards

- fe_1 = 0.816 N
- fe_2 = 0.789 N
- fe_3 = -0.100 N

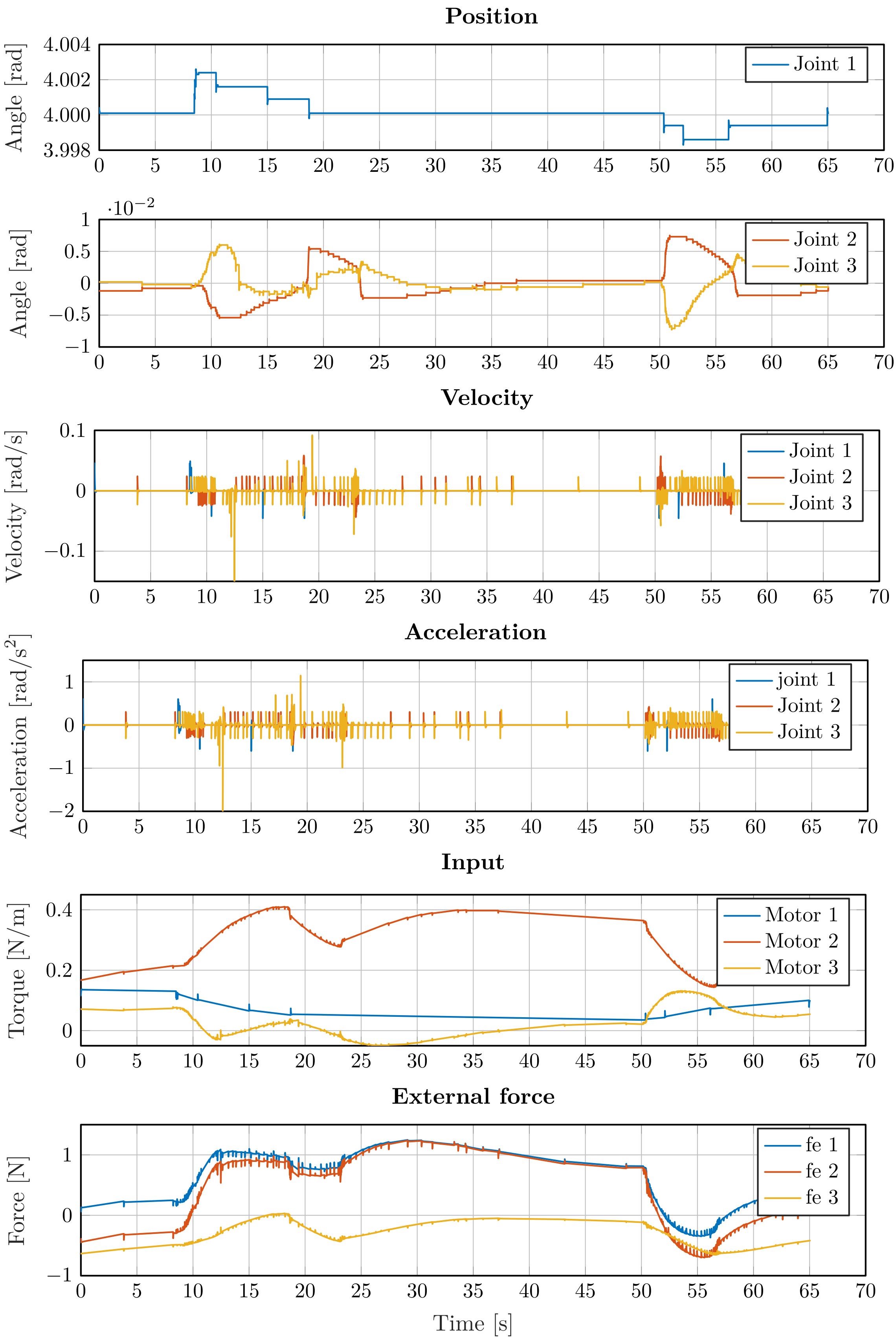

Figure E.3: This figure shows the estimation data for the uStepper robot, while an external force is being applied to the front of the end-effector.One and two (position), three (velocity), four(acceleration), five (input) and six (force)

E.4 External force applied to the back of the end-effector

In *Figure E.4* the data gathered while applying a force to the back of the end-effector is shown. The force is applied by attaching a Newton meter to the end-effector, which is extended to a force of 1.1 N. The force applied in this test is not static, as the force is made by human hand.
The time span for which the external force is applied to the end-effector is [7.04 $<$ t $<$ 44.61]
It can be seen when the external force is being applied that all angular positions are moving. However, the movement to joint one is probably due to human error. The largest position difference, can be seen for joint two and three, which is to be expensed, as these are responsible for the forward and backwards positioning of the end-effector. With regards to the force estimation, it can be seen that the force estimate fe_1 and fe_2 seems correlated as these have the same transient for the entire measurement period.
From the force being applied to the maximum estimated force an increase for the different estimates is measured to:

- Δfe_1 = -1.171 N
- Δfe_2 = -1.192 N
- Δfe_3 = -0.178 N

whereafter somewhat converges towards

- fe_1 = -1.037 N
- fe_2 = -1.637 N
- fe_3 = -0.816 N

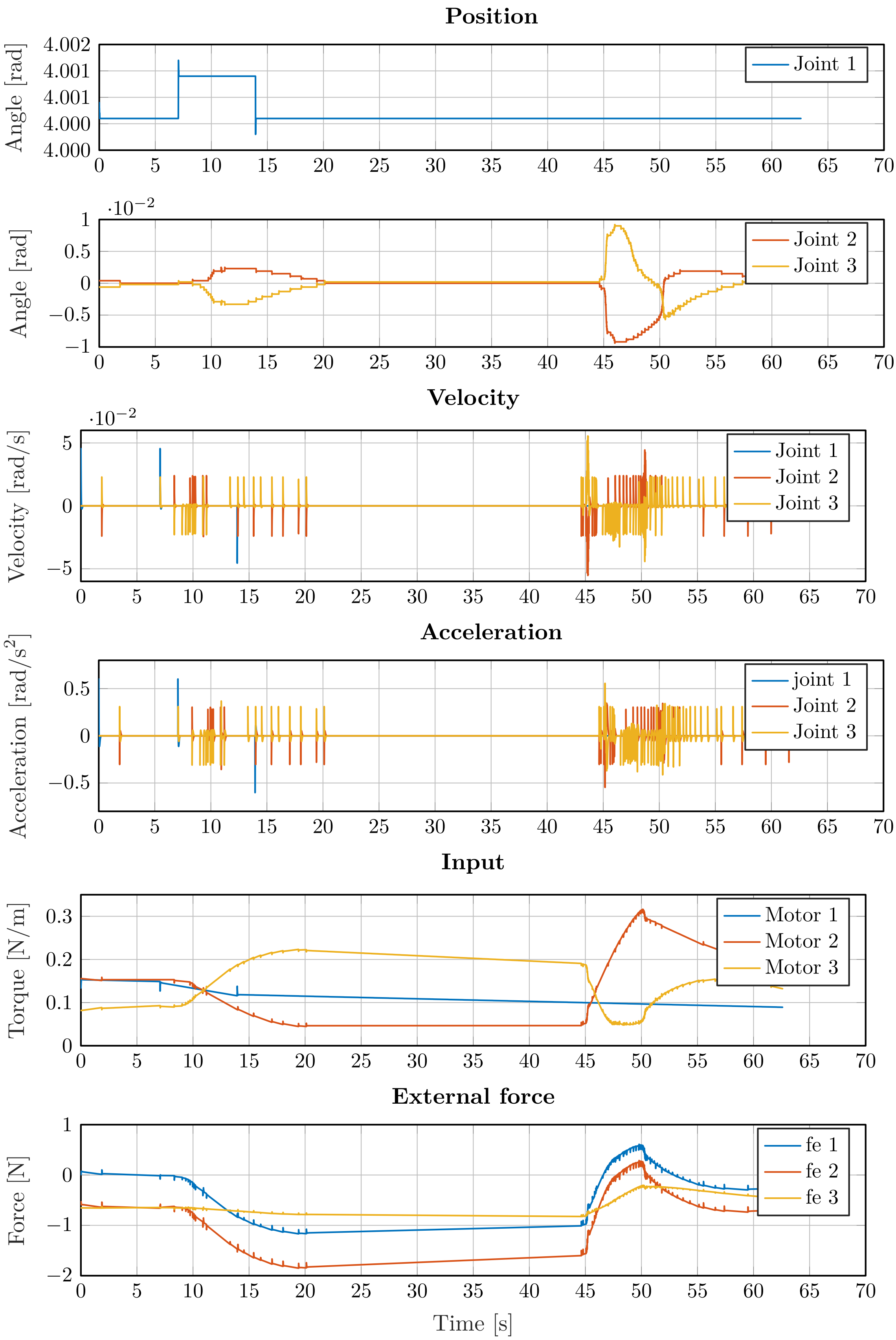

Figure E.4: This figure shows the estimation data for the uStepper robot, while an external force is being applied to the back of the end-effector.One and two (position), three (velocity), four(acceleration), five (input) and six (force)

E.5 Impulse

In *Figure E.5* the data gathered from an object being placed at the end-effector is shown. The object has a total mass of 270 g, which equals to 2.65 N. The force applied for this test is static.
At time step [t = 20.10] the object is being placed at the end-effector. This is clearly seeing in the position measurements, as these are highly affected by this.
When the object is placed, a large peak in the force estimates are shown, whereafter converging. From the object being placed the largest difference in the force estimates is measured to

- Δfe_1 = 3.147 N
- Δfe_2 = 3.444 N
- Δfe_3 = 3.986 N

whereafter somewhat converges towards

- fe_1 = -0.692 N
- fe_2 = -1.090 N
- fe_3 = 0.802 N

The time from the object being placed to the observer starts reacting is measured to eight milliseconds.
The time from the object being placed to the maximum estimated value is [$\Delta t_1 = 0.265$ $\Delta t_2 = 0.265$ $\Delta t_3 = 0.153$] milliseconds for fe_1, fe_2 and fe_3 respectively. Within the timespan of the requirement set to 100 milliseconds, the estimate difference is found to be

- Δfe_{t1} = 0.758 N
- Δfe_{t2} = 0.728 N
- Δfe_{t3} = 0.906 N

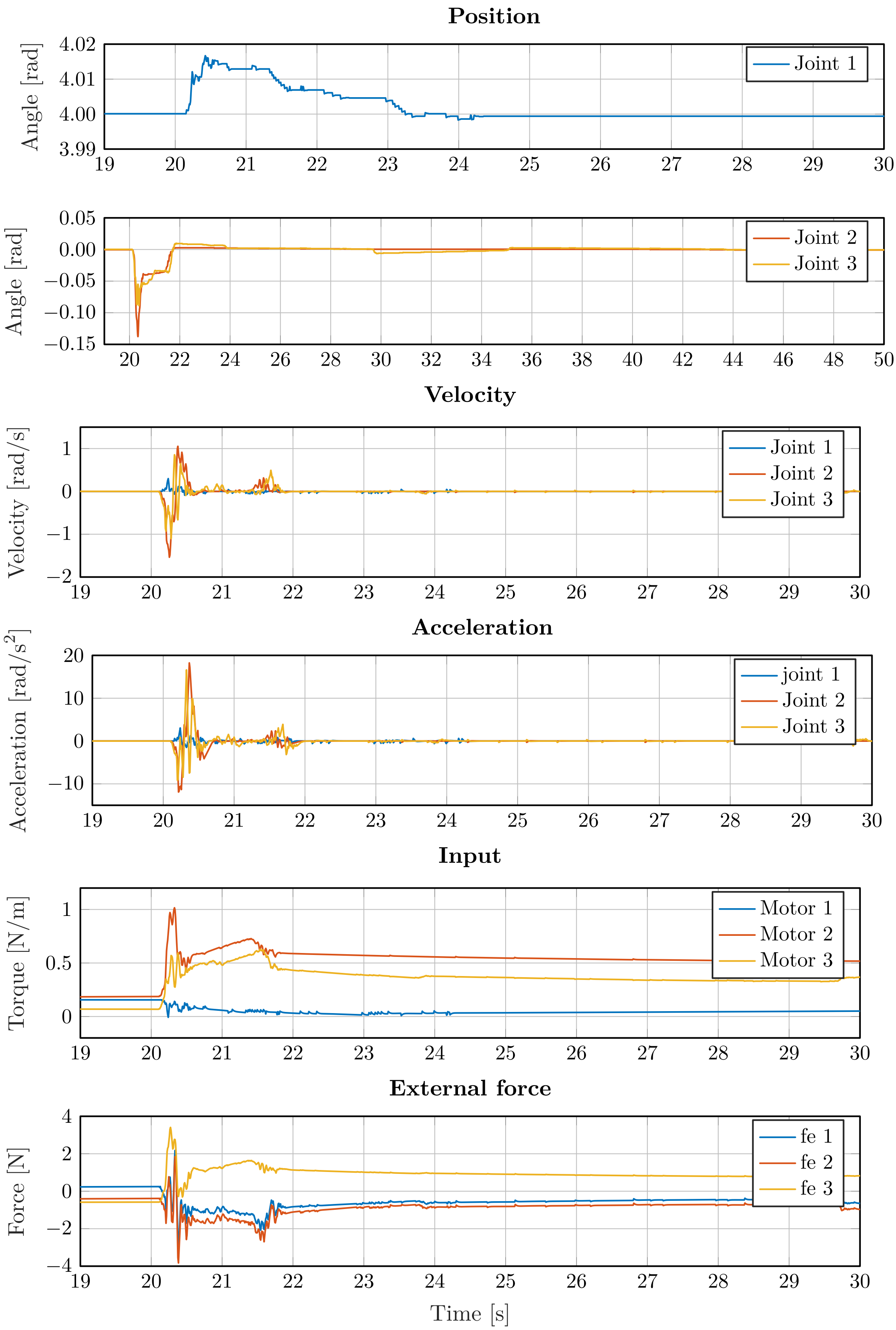

Figure E.5: This figure shows the estimation data for the uStepper robot, where an object is being placed on top of the end-effector. One and two (position), three (velocity), four(acceleration), five (input) and six (force)

E.6 Blockage for upwards directions

In *Figure E.6* the data gathered for the test, where a blockage is introduced for the system is shown. The system is being actuated at time instance 4.64 second, and the time of blockage is at 4.84 second. The actuation is however only for joint two and three as the motor for joint one is damage at the time of testing.
Under actuation of the system, it can be seen that large external forces is estimated. When the blockage is introduced, the estimates seems to stabilize, whereafter starts to increase for fe_1 and fe_3. The increase of force estimation is due to the integral action in the controller.
Different from the previous test in *Appendix: E.1*, *Appendix: E.2*, *Appendix: E.3* and *Appendix: E.6*, is that the it seems as if the correlation between fe_1 and fe_2 is gone. Looking at the input to the system it can be seen that the input to motor two is static, after actuation, however the input to motor three is not. Comparing the input to the force estimation in *Figure E.6*, could seems as the correlation lies in the dynamics of the input to the system.
with respect to the topic of Collision detection, is is not possible to separate the actuation of the system from the collision of it. This is based on the rapid changes in the force estimate under actuation.

After the collision the force estimate is measured to converge towards the following values.

- $fe_1 = 3.429$
- $fe_2 = 0.464$
- $fe_3 = 5.423$

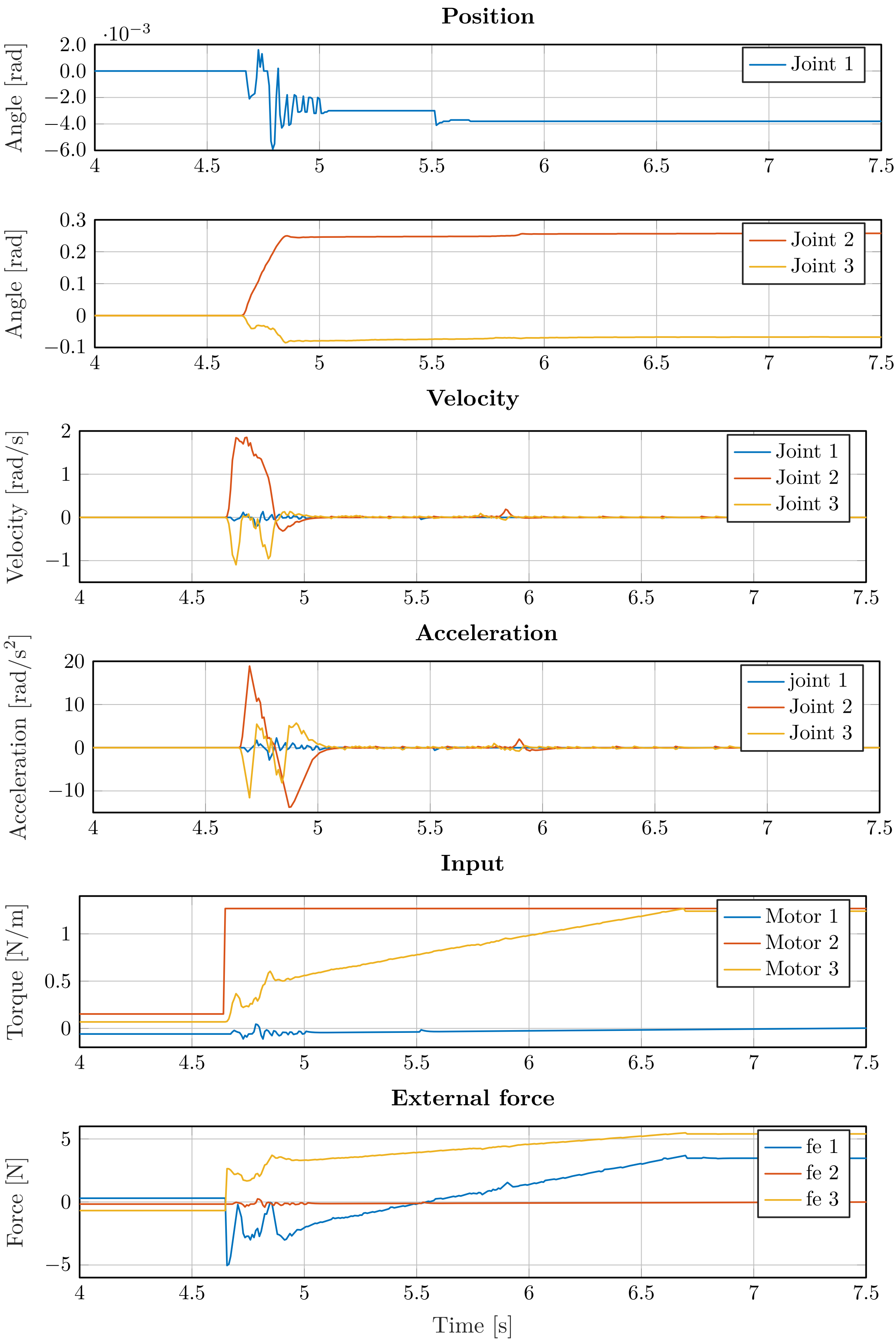

Figure E.6: This figure shows the estimation data for the uStepper robot, actuating in an upwards direction, for then being blocked in the direction. One and two (position), three (velocity), four(acceleration), five (input) and six (force)

E.7 Blockage for downwards directions

In *Figure E.7* the data gathered for the test, where a blockage is introduced for the system in a downwards direction is shown. The system is being actuated at time instance 2.64 second, and blockage at time 2.94 second. The actuation is only giving for motor two and three as the motor for joint one is damage at the time of testing.
Similar to the test in *Appendix: E.6*, a large force estimation is made when the system being actuated, and that the correlation between the force estimate fe_1 and fe_3 is present. By inspecting the input to the system, it can be seen that the force estimate fe_1 is following the input to motor three.
As there is integral action in the controller, the force is increasing due to this. When the input is saturated, it can be seen that the force estimate stops increasing.

- fe_1 = -6.227
- fe_2 = 0.588
- fe_3 = -4.829

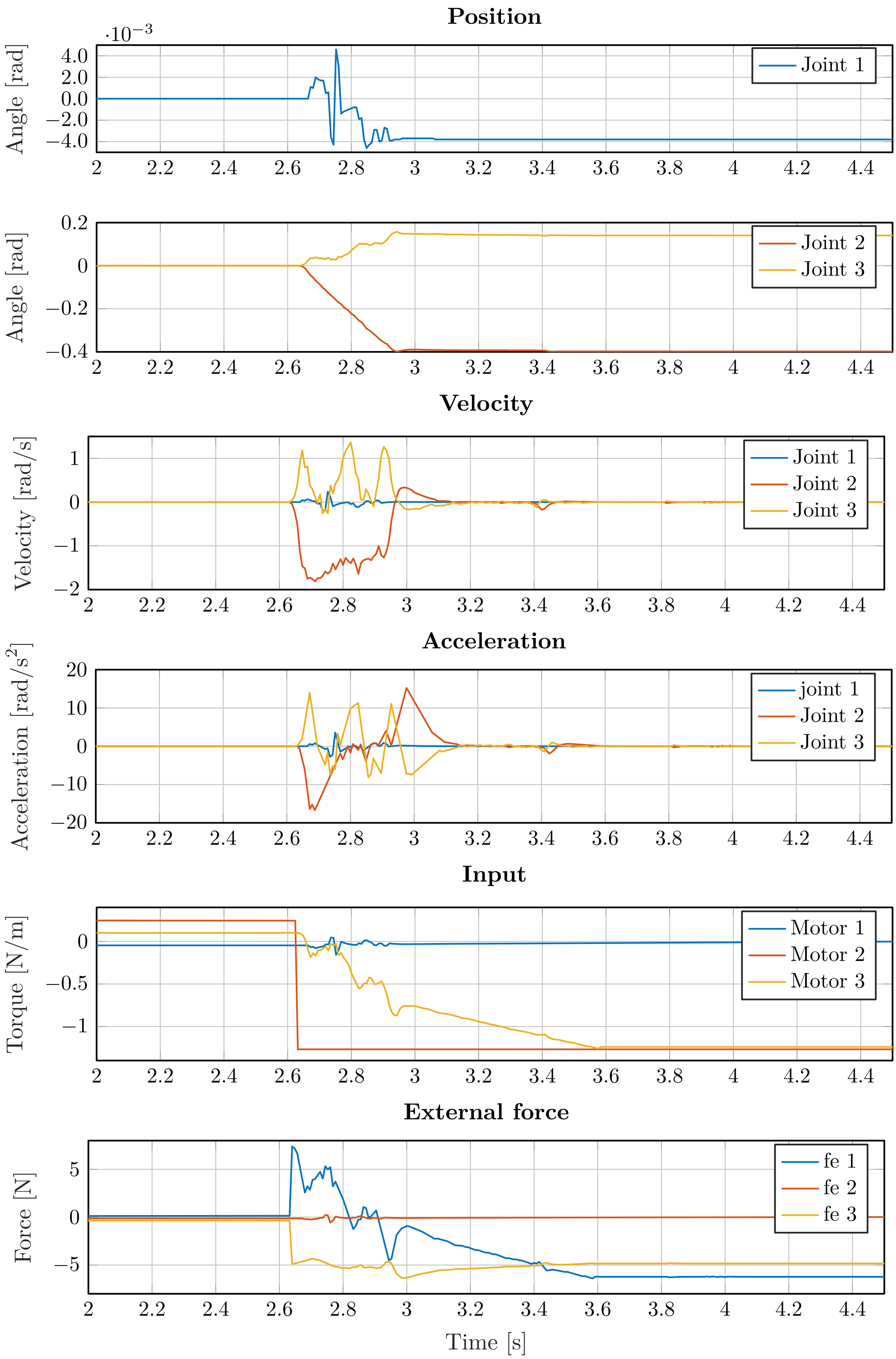

Figure E.7: This figure shows the estimation data for the uStepper robot, actuating in an downwards direction, for then being blocked in the direction. One and two (position), three (velocity), four(acceleration), five (input) and six (force)