



**CHALMERS**  
UNIVERSITY OF TECHNOLOGY

---



# **Design of a control system to an industrial lifting aid**

For movement in the horizontal plane

Master's thesis in Systems, Control and Mechatronics

**HERMAN LUNDBERG**



MASTER'S THESIS EX077/2018

# Design of a control system to an industrial lifting aid

For movement in the horizontal plane

HERMAN LUNDBERG



**CHALMERS**  
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering  
*Division of Systems and Control*  
CHALMERS UNIVERSITY OF TECHNOLOGY  
Gothenburg, Sweden 2018

Design of a control system to an industrial lifting aid  
For movement in the horizontal plane  
HERMAN LUNDBERG

© HERMAN LUNDBERG, 2018.

Supervisor: Richard Holström and Jakob Frid, Hofpartner AB, Tomas Hermansson,  
Camatec

Examiner: Yiannis Karayiannidis, Electrical Engineering

Master's Thesis EX077/2018  
Department of Electrical Engineering  
Division of Systems and control  
Chalmers University of Technology  
SE-412 96 Gothenburg  
Telephone +46 31 772 1000

Cover: The lifting aid Liongrip that was the basis for this thesis.

Typeset in L<sup>A</sup>T<sub>E</sub>X  
Gothenburg, Sweden 2018

Design of a control system to an industrial lifting aid  
For movement in the horizontal plane  
HERMAN LUNDBERG  
Department of Electrical Engineering  
Chalmers University of Technology

## Abstract

Liongrip is an industrial lifting aid that only has vertically actuated movement, therefore an operator has to use physical force for movement in the horizontal plane. Hofpartner AB requested a control structure design that would allow actuated movement in the horizontal plane as well with the aid of the operator. Since an operator was intended to interact with the end effector when controlling the arm safety is of utmost importance and thus the linear velocity and acceleration need to be limited. In this thesis several basic control structures were designed and tested by taking into account the constraints and the safety for the operator. Furthermore possible hardware to build a prototype was investigated. Robust methods using Damped least-square inverse for both velocity and acceleration control were tested and discarded since the velocity errors were too big particularly when the operator moved the arm through a singular configuration. In order to achieve smaller errors while operating the arm of Liongrip within the limits an inverse differential kinematics method was applied called Singular Value Filtering together with a velocity approximation of acceleration control that allowed a smoother response while operating close to singular configurations.

Actuators from SEW-EURODRIVE were proposed with lower gear ratio in order to allow the operator to aid the lifting aid in the movements and to only keep the machine strong enough. The number of RPM of the actuators should be as low as possible to result in a hardware limit of the possible maximum velocity. Low gear ratio and lower possible maximum velocity can minimise the damage if there is an accident. Bumpers should be added to each side of the arm in order to cut the power and activate the breaks in case of emergency. Further research should be conducted with a physical prototype to find what properties should characterise the actuators in a system aided by the operator's force, as well as to evaluate the safety of the system.

Keywords: Lifting Aid, Liongrip, Control, Simscape Multibody, Simulink, Single Value Filtering, PID.



## Acknowledgements

My thanks to all the employees at Camatec, and especially Tomas Hermansson, for welcoming me and giving me an office to work in. My thanks to all the employees at Hofpartner AB, and especially Rickard Holmström and Jakob Frid, who have provided guidance and support through the project. My thanks to Yiannis Karayiannidis who have answered all my questions and provided valuable information. A special thanks to Alf Renhag at SEW-EURODRIVE who answered all my questions about actuators, drivers and gears.

Herman Lundberg, Karlstad, June 2018



# Contents

<b>List of Figures</b>	<b>xi</b>
<b>List of Tables</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	2
1.2 Aim . . . . .	3
1.3 Research Questions . . . . .	3
1.4 Problem specification . . . . .	4
1.5 Limitations . . . . .	4
1.6 Description of Liongrip . . . . .	5
1.7 Thesis levels . . . . .	5
<b>2 Theory</b>	<b>7</b>
2.1 Robotics . . . . .	7
2.1.1 Forward Kinematics . . . . .	7
2.1.2 Differential Kinematics . . . . .	8
2.1.3 Inverse Differential Kinematics . . . . .	9
2.1.4 Robust Inverse Differential Kinematics . . . . .	9
2.1.5 Second Order Inverse Differential Kinematics . . . . .	11
2.2 Control Design . . . . .	13
2.2.1 Feedback and feedforward . . . . .	14
2.2.2 PID-control . . . . .	14
2.2.3 Filtering . . . . .	14
2.2.4 Acceleration vs Velocity control . . . . .	15
2.2.5 Control for safety . . . . .	15
2.3 Dynamic Simulation . . . . .	15
2.3.1 Dynamic properties . . . . .	16
2.3.2 Disturbances . . . . .	17
2.4 Hardware Properties . . . . .	17
2.4.1 Actuators . . . . .	17
2.4.2 Sensors . . . . .	19
2.4.3 Safety incorporated into hardware . . . . .	19
2.4.4 Bumper . . . . .	20
2.5 Programming . . . . .	20
<b>3 Modelling of Liongrip</b>	<b>21</b>

3.1	Kinematic description of Liongrip . . . . .	21
3.1.1	Differential Kinematics . . . . .	21
3.1.2	Singularities . . . . .	23
3.1.3	Safety measures . . . . .	24
3.2	Modelling, Simulation and Experiments . . . . .	24
3.2.1	Modelling . . . . .	24
3.3	Hardware . . . . .	26
3.3.1	Hardware from SEW . . . . .	27
3.3.2	Hardware requirements . . . . .	27
<b>4</b>	<b>Comparison of singularity-robust methods</b>	<b>29</b>
4.1	General information . . . . .	29
4.1.1	The input signal . . . . .	32
4.2	Open-loop . . . . .	32
4.3	Closed-loop . . . . .	34
4.4	Acceleration Control . . . . .	38
<b>5</b>	<b>Results and discussion</b>	<b>43</b>
5.1	Simulation Results . . . . .	43
5.2	Hardware . . . . .	50
5.3	Discussion . . . . .	52
5.3.1	The operator . . . . .	52
5.3.2	Safety . . . . .	52
5.3.3	Sustainability . . . . .	53
5.3.4	Control design . . . . .	53
<b>6</b>	<b>Conclusion and future work</b>	<b>55</b>
6.1	Conclusion . . . . .	55
6.2	Future Work . . . . .	56
	<b>Bibliography</b>	<b>57</b>
<b>A</b>	<b>Appendix 1</b>	<b>I</b>
A.1	Jacobian . . . . .	I
A.1.1	Jacobian . . . . .	II
A.2	Derivation of the singularity . . . . .	III

# List of Figures

1.1	Liongrip - The lifting aid . . . . .	3
1.2	A schematic of Liongrip where the important parts are annotated . . . . .	6
3.1	The defined coordinate system for the end effector . . . . .	22
3.2	A schematic view of links and joints for the arm of Liongrip . . . . .	22
3.3	Active force diagram to calculate friction in the joints of Liongrips arm . . . . .	25
4.1	The 3D model resulting Simscape Multibody complete with actuators . . . . .	30
4.2	The input before and after the filter. The input signals are displayed in solid lines and the filtered signals are shown in dashed lines . . . . .	31
4.3	A block diagram view of the open loop system using DLSI . . . . .	33
4.4	Simulation of the open loop DLSI system . . . . .	33
4.5	A block diagram view of the open loop system using SVF . . . . .	33
4.6	Simulation of the open loop SVF system . . . . .	34
4.7	A schematic view of the control structure while using DLSI and feedback control . . . . .	34
4.8	Simulation of the feedback loop DLSI system . . . . .	35
4.9	The system using SVF and a feedback loop . . . . .	35
4.10	Simulation of the feedback loop SVF system . . . . .	36
4.11	A block diagram of the system using feedback, feedforward and DLSI . . . . .	36
4.12	Simulation of the feedback, feedforward DLSI system . . . . .	37
4.13	A block diagram showing the system using SVF, feedforward, feedback and acceleration control . . . . .	37
4.14	Simulation of the feedback, feedforward SVF system . . . . .	38
4.15	A schematic block view of the open loop system using DLSI and acceleration control . . . . .	38
4.16	Simulation of the open-loop system with DLSI and acceleration control . . . . .	39
4.17	A schematic block diagram of the open loop system using SVF and acceleration control . . . . .	39
4.18	Simulation of the open loop SVF system with acceleration control . . . . .	40
4.19	A schematic view of the system using feedback, feedforward, acceleration control and DLSI . . . . .	40
4.20	Simulation of the feedback, feedforward DLSI system with acceleration control . . . . .	41
4.21	A schematic block diagram of the feedback, feedforward SVF system and acceleration control . . . . .	41

4.22	Simulation of the feedback, feedforward SVF system with acceleration control . . . . .	42
5.1	A block diagram view of the open loop system using SVF and acceleration control . . . . .	43
5.2	The linear acceleration for the end effector when using open loop SVF and acceleration control . . . . .	44
5.3	The linear velocity of the end effector when using open loop SVF and acceleration control . . . . .	44
5.4	The torque required in each of the three actuated joints to perform the movement when using open loop SVF and acceleration control . .	45
5.5	A plot showing the angular velocity in the three joints when using open loop SVF and acceleration control . . . . .	45
5.6	A plot showing the angular acceleration in the three joints when using open loop SVF and acceleration control . . . . .	46
5.7	A block diagram view of the feedback, feedforward system using SVF and acceleration control . . . . .	46
5.8	The Linear acceleration of the end effector in the feedback, feedforward SVF-system . . . . .	47
5.9	The Linear acceleration of the end effector with the feedback, feedforward SVF-system . . . . .	47
5.10	A plot showing the torque needed in the three actuated joints for the system using SVF, feedback and feedforward . . . . .	48
5.11	A plot showing the angular velocity in the three joints using SVF, feedback and feedforward . . . . .	48
5.12	A plot showing the angular acceleration in the three joints using SVF, feedback and feedforward . . . . .	49
5.13	The errors of the SVF systems using acceleration control . . . . .	49
5.14	Overview of the hardware for a proposed control system (updated in drive) . . . . .	50

# List of Tables

3.1	The desired properties on the actuators . . . . .	27
4.1	The parameters used per system . . . . .	32
5.1	Possible hardware list . . . . .	51
A.1	The variables names and values used to create the system . . . . .	I



# 1

## Introduction

A lot of the products and materials in the industry are heavy and cumbersome to move and often mechatronic devices have to be used in order for the load to be moved at all. The load can for example be part of the body to a car, big sheets of metal or heavy duty core shafts. The devices to help move the objects come in different forms and sizes such as a hand truck, or robotic arm-like manipulators either attached to a rig in the roof or attached to the floor [8], [21]. Some tools used to move objects, or the lack of tools, can produce a physical strain on the body of the operator, which can lead to work related injuries. This thesis will look into the evolution of a lifting aid where it shall be operated with actuators to relieve some, or all, of the operators physical strain.

The lifting aid at the basis of the thesis can be modelled as a robotic arm. Extensive research has already been done in the movement and control of robotic arms, where the movement can be done by using kinematics relations between joint positions and end effector position, joint velocities and end effector velocities, et cetera. These relations are often based of the inverse of a matrix and for a matrix to be invertible it has to be square and of full rank. The fact that the matrix of the system can be invertible for some configurations, and not for others makes the matter more complicated. Even more so when proximity to the troublesome regions can induce dangerous movements in the arm. Research to handle this has resulted in at least two different methods: 1) avoid the troublesome configurations, or 2) ensure that the matrix of the system is always invertible [26].

To avoid the troublesome configurations path-planning is used to calculate safe paths and in some instances combined with adding more joints to the robot arm than what is strictly necessary to enable a choice of configurations that avoids these configurations [26]. To ensure that the matrix of the system is always invertible methods like the Damped least-squares inverse [6] and Singular Value Filtering [4] can be used. The drawback is that by alternating the system matrix errors are introduced.

This thesis investigated how to ensure that the lifting aid was controllable in its entire workspace, hence methods to ensure that the system matrix was always convertible. Due to the operator controlling the arm live no optimal path-planning could be made, and due to the lifting aid being preexisting no extra joints could be added to avoid the troublesome configurations.

### 1.1 Background

Solutions to make the movement of heavy objects possible have been used since prehistoric times and could have been as simple as rolling heavy objects on boles, or dragging it on top of a pelt. Modern solutions are using winches, forklifts and tools specifically designed for the given task or object. Some of these can manipulate several kinds of objects and weights and perform different tasks, however a general and easy solution is hard to find. Heavy objects requires sturdier equipment often at the lack of manoeuvrability and vice versa. A company that aims at developing a heavy duty lifting aid that retains great manoeuvrability is Hofpartner AB.

Hofpartner AB develops, sells and manufactures: the lifting aid Liongrip. Liongrip is a smart lifting tool with a pneumatic tower and an arm to hold the load. The arm can be moved in the horizontal plane using physical force and vertically using a pressure sensitive joystick to control the pneumatic actuator. It was first developed to handle heavy core shafts, however the costumers have started to use it to lift other objects as well [11].

Today Liongrip can handle weights up to 240 kg, however it starts to get physically demanding for the operator already at 150 kg to move and angle the weight. When applying larger weights the forces on the mechanical system may result in unstable behaviour, that is the momentum is no longer negated and a drift can occur. To easily manage heavier weights, a possibility could be a control structure with actuated joints for Liongrip's arm.

This thesis aims to design a control scheme that would allow Liongrip to handle weights far above 240 kg for the movement in the horizontal plane. By using a controller an operator can handle heavy loads with little to no physical strain. The control scheme has to negate the drift without the loss of the operators manoeuvrability. The goal is to explore the possibility of a full scale prototype that can help with the handling of 150 kg. The actuators is to aid an operator when moving the load, however not necessarily move it entirely by the torques of the actuators, in turn the operator can help with his or her physical force when needed. As such the system is required to accept help, and to have fail-safes against the degrading of the actuators if the load is to heavy. The resulting control design is to be added side by side to the already implemented control for the vertical movement, and the two systems is not to be interconnected in this thesis.

The Liongrip is presented in figure 1.1. The arm that holds the weight can be moved up and down with the joystick that is in the operators left hand. The right hand is on the handle used in the present solution where force is needed to manually manipulate the object in the horizontal plane to and from the drop off location. The built in inertia of the system results in an arm easier to control smoothly. It is desired that the system reacts calmly, for example with limited acceleration, in order to limit possible harm the arm can cause if an operator by mistake triggers a high velocity move.



**Figure 1.1:** Liongrip - The lifting aid

## 1.2 Aim

The aim of this thesis is to research and design a control system to steer the arm through control inputs and handle unintentionally drift created by a heavy load. The result is to propose a control structure together with a list of possible hardware.

## 1.3 Research Questions

The goal of this thesis was to design and evaluate a control system for the lifting aid called Liongrip. The system were to be evaluated with regards to safety, as well as functionality and manoeuvrability for an operator. All the questions were constrained to changes that affected the movement in the horizontal plane.

- What safeguards are required to ensure operator safety?
- How will the system be designed to incorporate the above mentioned safeguards?
- What control structure makes for precise and comfortable handling by an operator?

### 1.4 Problem specification

A series of specifications were given at the start of the thesis in order to limit and focus the work.

- Resolution of a movement - High precision movements are vital for the handling of heavy duty core shafts and as such the smallest movement possible was to be 1 mm
- Top linear Velocity - Heavy loads with high speeds can quickly cause accidents and therefore the operators control signals were limited to 0.25 ( $m/s$  and  $rad/s$ ) for each component giving a limit of 0.354  $m/s$  for the norm of the linear velocity of the end effector
- Top linear acceleration - A higher acceleration leads to a higher demand of torque in the joints and consequently a higher demand on the actuators, as such this was limited to 0.2 ( $m/s^2$  and  $rad/s^2$ ) for each component of the control signal, giving a limit of 0.283  $m/s^2$  for the norm of the linear acceleration of the end effector
- Size of arm - The length of the arm links will be 1750 mm, giving the total arm length of around 4300 mm
- Max load - Inside this thesis the max load of the arm was no more than 150 kg
- Number of joints - Only three of the five joints were used for this thesis. One was locked and as an effect slightly reduced the work space, and two was geared together
- The entire reachable space of the arm should be usable

### 1.5 Limitations

The added strain on the mechanical structure made possible with the controller was not evaluated in this thesis. Vertical movement is already implemented and was not further developed in this work, nor was the interconnection of the two systems to be researched. A physical prototype was not built. No motion planning was delved into, since the result of this work aimed at a control scheme for an operator driven arm, it was left to the operator to do the planning. External sensors so as to avoid or warn for obstacles while moving was not researched. Introducing areas inside the

reachable space of the arm where the movement of the arm should be blocked, was not investigated.

## 1.6 Description of Liongrip

To put it simply Liongrip is made up of a tower and an arm. The arm is attached to a pneumatic actuator that can move it up and down. The arm has five joints which can all be viewed in figure 1.2, however to reduce cost and complexity only three joints were actuated. The tower joint was locked and the driven joint was connected to the middle joint so that an actuator on the middle joint drove them both. For the scope of the thesis that left 3 actuated joints and Liongrips arm was reduced to a three link planar arm. Throughout this report the set of joints will include 1, 2 and 4, with joint 3 being specifically mentioned when it affects joint 2. The joint number will be used interchangeably with the joint name, for example Joint 4 or Outermost Joint will both be used.

## 1.7 Thesis levels

The following part of the thesis is organised in the following manner:

**Chapter 2 - Theory:** This chapter presents the theory needed to understand the work and specifically handles movement of robot arms, control theory, modelling and simulation and properties of the hardware.

**Chapter 3 - Modelling of Liongrip:** This chapter presents the modelling of Liongrips arm.

**Chapter 4 - Comparison of singularity-robust methods:** This chapter compares different robust methods to control the arm.

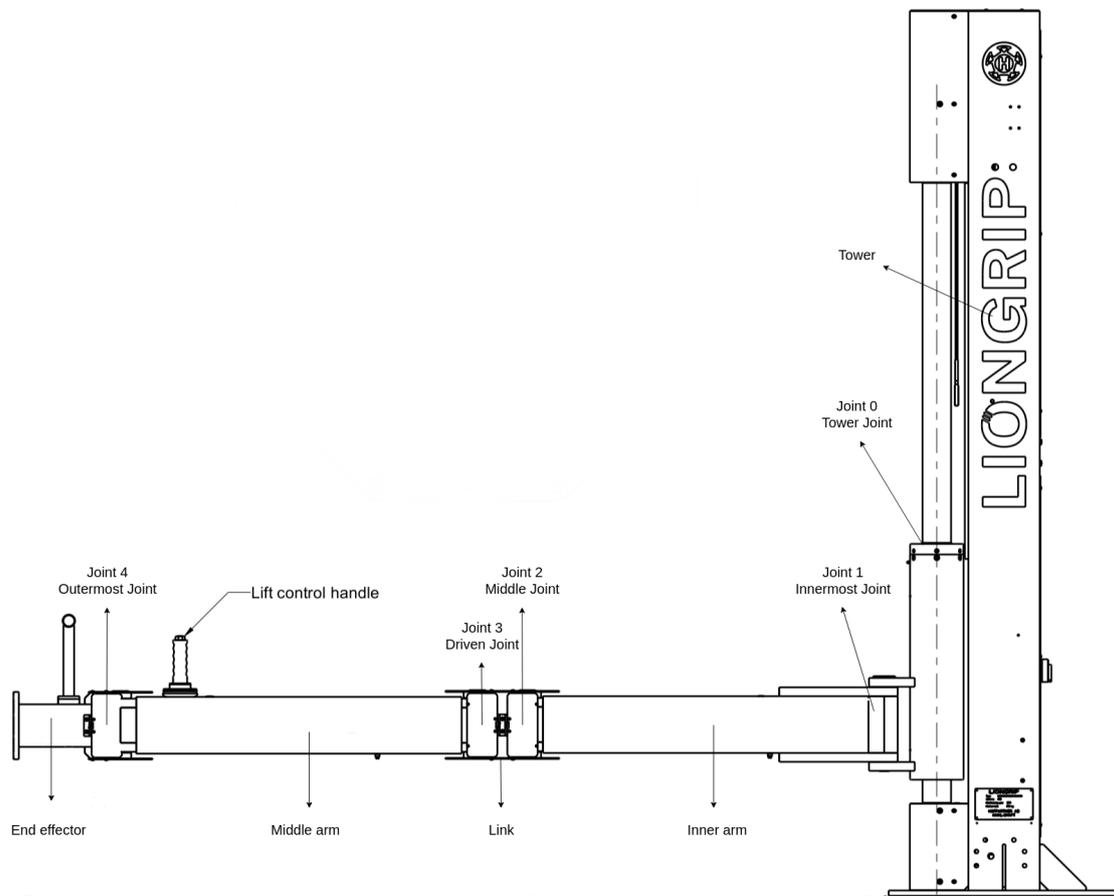
**Chapter 5 - Results and discussion:** This chapter presents the results found in Chapter 4 and discusses the results.

**Chapter 6 - Conclusion and future:** This chapter presents the conclusions drawn from this work, what could be done to improve the results and some suggestions for future work.

**Appendix:** This chapter presents extra material that did not enhance the reading of the report.

# 1. Introduction

---



**Figure 1.2:** A schematic of Liongrip where the important parts are annotated

# 2

## Theory

In the following sections theory required for the understanding of the work done in this thesis is presented. Section 2.1 presents the theory linked to robotic arms, section 2.2 is regarding the theory for the control design, section 2.3 is related to modelling and simulation, section 2.4 is about the hardware, such as actuators, sensors, and microcontroller, and their required properties, and section 2.5 presents information about how the proposed control structure would be programmed.

### 2.1 Robotics

The arm of Liongrip with its five joints can be modelled as a five link planar arm, and with joint 0 locked and joint 2 and 3 geared together it can be modelled as a three link planar arm. Therefore all the theory in robotics that is applicable to these kind of arms were also applicable to the arm of Liongrip. For this thesis emphasises will be put on Differential Kinematics and algorithms for robust movements which can be viewed below.

#### 2.1.1 Forward Kinematics

Degrees of freedom (DOF) describes how many kinds of movements a system can make and the number of variables required for its configurations. It's either a translation, a rotation or any number of combinations of those that makes up the DOF of a system. A train on a rail has one degree of freedom since it can only make a translational movement. The propeller of a windmill has one degree of freedom since it can only perform a rotation. A combination of a windmill and a train would give the propeller 2 degrees of freedom [17].

When the number of DOFs is greater than the necessary amount of variables to describe a specific task the system is said to be kinematically redundant [26]. A three link planar arm can do three rotations in the horizontal plane and thus access all the places in that plane within bounds of the length of the links, thus it has three degrees of freedom. Given that the arms configuration are the set of three variables and the number of DOFs is equal to the number of variables describing the configuration the arm is non-redundant.

Regardless of what commands are used to control the robot arm, for example, position or velocity, it is important to know in what coordinate frame the robot is

defined, otherwise the commanded movement may not equal the actual movement. More of changing frames in section 2.1.2.

Many industrial robots are controlled using position of the joints or of the end effector, however this was not possible for Liongrips arm because the operator did not know the target positions. Instead control through velocities was used and that is called Differential Kinematics [26].

### 2.1.2 Differential Kinematics

Differential Kinematics describes the relationship between the joints angular velocities,  $\dot{q}$ , and the linear and angular velocities of the end effector,  $\dot{p}$ .

$$\dot{p} = J(q)\dot{q} \tag{2.1}$$

where  $q$  is the angles of the joints and  $J$  is the Jacobian of the system [26].

The main part of the differential kinematics is the Jacobian,  $J(q)$ , that translates the input signals to the corresponding angular velocities. Throughout the report  $J(q)$  and  $J$  will be used interchangeably to symbolise the Jacobian. The Jacobian can be formed through the assembly as displayed in the following equation:

$$J = \begin{bmatrix} J_{P1} \cdots J_{Pn} \\ \vdots \\ J_{O1} \cdots J_{On} \end{bmatrix} \tag{2.2}$$

where  $J_P$  is the contribution from linear velocity and  $J_O$  is the contribution from the angular velocity and they are given by the following equations:

$$\begin{bmatrix} J_{Pi} \\ J_{Oi} \end{bmatrix} = \begin{bmatrix} z_{i-i} \times (p_e - p_{i-1}) \\ z_{i-1} \end{bmatrix} \tag{2.3}$$

where  $p_i$  denotes the position vectors of the joints and end effector, from the innermost joint,  $p_0$  to the end effector at  $p_n$ , and  $z_i$  is the unit vector of the joint axis from the innermost joint,  $z_0$  to the outermost joint of  $z_{n-1}$ , where  $n$  is the number of joints [26].

For differential kinematics rotation matrices are used to redefine the frame the object is defined in. Let  $R_e^0$  be the rotation matrix from origin to the end effector and let  $J_0$  be the Jacobian of the system defined in the origins frame. Then the Jacobian,  $J_e$ , defined in the end effectors frame is derived as follows [26]:

$$J_e = R_e^0 J \tag{2.4}$$

### 2.1.3 Inverse Differential Kinematics

In section 2.1.2 the relations between angular velocities of the joints,  $\dot{q}$ , and the velocities of the end effector,  $\dot{p}$ , were presented, however to control the movement using  $\dot{q}$  will not be intuitive for an operator and therefore the relation in equation (2.1) needs to be reversed, see equation (2.5).

$$\dot{q} = J(q)^{-1}\dot{p} \quad (2.5)$$

where  $J(q)^{-1}$  is the inverse of matrix  $J(q)$ . In order to take the inverse of a matrix it has to be symmetric and be of full rank. Because of the full rank condition the Jacobian will not be invertible for all positions of the arm since some angles can make the Jacobian lose rank and render it non-invertible. This is called a singularity [26].

Singularities often happens when some links of an arm are parallel, and they must be either avoided or handled in a special way to prevent hazardous movements. A singularity is a configuration where the mobility of the arm is reduced and small linear movements can incur impossibly high angular velocities and accelerations to the joints, which can be a risk both for the hardware of the arm as well as for the operator. There is also the computational problem where the number of solutions to the jacobian may increase in the proximity of a singularity and there may even be infinitely many solutions at a singularity. There are two types of singularities [26]:

- Boundary singularities occur when the arm is fully outstretched or retracted [26]
- Internal singularities occur inside the reachable space of the arm and they are usually because of the alignment of two or more axes [26]

Singularities are found by investigating when the Jacobian loses rank, which can be calculated by taking the determinant of the Jacobian and investigating at what values it will be equal to 0.

Most of the research consists of avoiding singularities all together, but there are methods that can handle them, such as the Damped Least-squares-inverse (DLSI) and the Singular Value Filtering (SVF). The drawback is decreased accuracy, which is often costly for industrial robots where a given trajectory must be matched to the fraction of a millimeter, however in this thesis this introduced error could be acceptable since there would be an operator able to adjust for the inaccuracies [26].

### 2.1.4 Robust Inverse Differential Kinematics

When a matrix is either not square or not of full rank there is the possibility to use the pseudo-inverse instead of the true inverse, which creates a square matrix that is invertible. For a square full rank Jacobian,  $J$ , the pseudo-inverse,  $J^+$ , defined by:

$$J^+ = J^T(JJ^T)^{-1} \quad (2.6)$$

where  $J^T$  is the transpose of  $J$ , is the same as the inverse Jacobian,  $J^{-1}$  [6]. By premultiplying both sides of (2.1) with the pseudo-inverse given by (2.6) we get the joint velocity minimum norm:

$$\dot{q} = J^+ \dot{p} \quad (2.7)$$

This property that gives joint velocity minimum norm will be used further down to minimise the acceleration as well. The pseudo-inverse can be further developed with a damping factor in order to allow inversion of the Jacobian in close proximity to a singularity. This development is called Damped Least-squares-inverse (DLSI) [6].

DLSI is a method that can handle the high velocities created in proximity to a singular configuration. It is based of the pseudo-inverse with an added damping factor,  $\lambda$ , and when  $\lambda = 0$  the damped least-squares-inverse is equal to the pseudo-inverse. It's defined by the following equation:

$$J^* = J^T (J J^T + \lambda^2 I)^{-1} \quad (2.8)$$

where  $I$  is the identity matrix [6].

Systems with DLSI is often built in such a way that the damping factor is only used when in close proximity to a singularity so that it can be scaled up for more damping when moving closer to a singular configuration. For non-redundant systems using DLSI this is an advantage since when the damping factor is set to 0, the DLSI equals true inverse of the system. This ensures higher precision and accuracy in the safe zones and prevents high velocity moves because of an ill-conditioned Jacobian close to a dangerous zone [6].

The proximity to a singularity and when to use the damping factor can be measured by examining the manipulability measure,  $w$ , of the arm and comparing it to a threshold value. The manipulability measure is defined by the following equation:

$$w = \sqrt{\det(J J^T)} \quad (2.9)$$

When  $w$  is getting smaller the arm is approaching a singularity. A condition can be set, where a threshold value,  $w_t$ , can be used as a design parameter to decide when to dampen the system and thus avoid impossibly high velocities, while retaining maximum accuracy when  $w$  is larger than  $w_t$  [9].

One way of scaling the damping factor,  $\lambda$ , is to use the manipulability measure,  $w$ , with the following equation:

$$\lambda = \lambda_0 \left(1 - \frac{w_t^2}{w}\right) \quad (2.10)$$

where  $\lambda_0$  is a chosen damping coefficient. This will lead to a damping factor that achieves maximum value,  $\lambda_0$ , at a singularity, further away the  $\lambda$  decreases and when the arm is no longer close to a singularity the damping factor equals to 0 [6].

Another method that deals with the singularities is the Singular Value Filtering (SVF) which uses Singular Value Decomposition (SVD) to add a value to the eigenvalues of the Jacobian making it always full ranked and thus invertible. To explain SVF SVD has to be explained first. SVD is the factorisation of a matrix, real or complex, which fulfils the following criteria:

$$J = U\Sigma V^T \quad (2.11)$$

where  $U$  and  $V$  are unitary matrices and  $\Sigma$  is a diagonal matrix with non-negative real numbers of the Jacobian [20]. The  $\Sigma$  matrix diagonal values,  $\sigma$ , are called singular values and they are defined as the square-root of the eigenvalues of the Jacobian. If SVD is applied to the Jacobian,  $J$ , further changes can be made in accordance to Singular Value Filtering.

By adding a function  $h_{\nu,\sigma_0}(\sigma)$  to every singular value of the matrix  $\Sigma$  the Jacobian,  $J$ , will for no configurations be singular and it will always be of full rank, this is called Singular Value Filtering. A Jacobian, based of SVF, can then be defined in the following way:

$$J_{SVF} = \sum_{i=1}^n h_{\nu,\sigma_0}(\sigma_i) u_i v_i^T \quad (2.12)$$

and the following can be used as the Jacobian pseudo-inverse:

$$J_{SVF}^+ = \sum_{i=1}^n \frac{1}{h_{\nu,\sigma_0}(\sigma_i)} u_i v_i^T \quad (2.13)$$

where  $n$  is the number of rows of  $U$  and  $u_i$  and  $v_i$  are the rows of the respective matrix  $U$  and  $V$ . The function  $h_{\nu,\sigma_0}(\sigma)$  is defined as:

$$h_{\nu,\sigma_0}(\sigma) = \frac{\sigma^3 + \nu\sigma^2 + 2\sigma + 2\sigma_0}{\sigma^2 + \nu\sigma + 2} = \sigma + \frac{2\sigma_0}{\sigma^2 + \nu\sigma + 2} \quad (2.14)$$

where  $\sigma_i$  is the singular value corresponding to the row  $i$  and  $\sigma_0$  is the minimum value to impose on the system. By adding  $\sigma_0$  to the system an error is introduced since the matrix inversion is not exact anymore, therefore the smaller  $\sigma_0$  the smaller error introduced.  $\nu$  is a shape factor that should be chosen with the following constraints for the function  $h_{\nu,\sigma_0}(\sigma)$  to remain monotonic:  $\nu > \sigma_0$  and  $2 > \nu\sigma_0$ . Monotonicity ensure that the pseudo-inverse is always constrained by the minimum value  $\sigma_0$ . [4].

### 2.1.5 Second Order Inverse Differential Kinematics

If the acceleration of the arm shall be controlled second order inverse differential kinematics must be introduced. Considering that there are three actuated joints for Liongrips arm there can often be several combinations of angular velocities and accelerations that can be used to produce a certain linear velocity and acceleration

of the end effector, and there will be one combination with minimum angular acceleration that correlates to minimum linear acceleration. This can be solved by a relation based on the optimisation problem in equation (2.1.5) further down. First some mathematical relations and approximations is introduced. In continuous time the relation between the end effector's acceleration,  $\ddot{p}$  and the joints acceleration,  $\ddot{q}$ , are as follows:

$$\ddot{p} = J(q)\ddot{q} + \dot{J}(q)\dot{q} \quad (2.15)$$

$$\ddot{q} = J(q)^{-1}(\ddot{p} - \dot{J}(q)\dot{q}) \quad (2.16)$$

where  $\dot{J}$  is the derivative of the Jacobian  $J$ . In discrete time the acceleration  $\ddot{p}_k$  of the end effector can be approximated as:

$$\ddot{p}_k \simeq \frac{\dot{p}_k - \dot{p}_{k-1}}{T} \quad (2.17)$$

where  $k$  denotes the current time step and  $T$  is the sampling time. The result is that the acceleration control could be applied to a control structure using velocities.

In a similar way as in (2.17) the time-derivative of  $\dot{J}$  can be approximated using:

$$\dot{J}_k \simeq \frac{J_k - J_{k-1}}{T} \quad (2.18)$$

where  $q_k$  is used for  $J_k$  and  $q_{k-1}$  is used for  $J_{k-1}$ . Wherever a joint velocity  $\dot{q}$  is used in continuous time either  $q_{k-1}$  or the current and the last joint position measurement should be used in discrete time, hence the following approximation is used:

$$\dot{q} \simeq \dot{q}_{k-1} \simeq \frac{q_k - q_{k-1}}{T} \quad (2.19)$$

The equation (2.16) can now be approximated in discrete time as:

$$\begin{aligned} \ddot{q}_k &= J_k(q)^{-1}(\ddot{p}_k - \dot{J}_k(q)\dot{q}_k) \\ &\simeq J_k^{-1}\left(\frac{\dot{p}_k - \dot{p}_{k-1}}{T} - \left(\frac{J_k - J_{k-1}}{T}\right)\dot{q}_{k-1}\right) \\ &\simeq \frac{1}{T}(J_k^{-1}\dot{p}_k - J_k^{-1}\dot{p}_{k-1} - \dot{q}_{k-1} - J_k^{-1}J_{k-1}\dot{q}_{k-1}) \\ &\simeq \frac{1}{T}(J_k^{-1}\dot{p}_k - J_k^{-1}\dot{p}_{k-1} - \dot{q}_{k-1} - J_k^{-1}\dot{p}_{k-1}) \\ &\simeq \frac{1}{T}(J_k^{-1}\dot{p}_k - \dot{q}_{k-1}) \end{aligned} \quad (2.20)$$

Applying the minimising properties of the pseudo-inverse, some definitions of numerical differentiation and algebra second order inverse differential kinematics where

the acceleration are to be minimised can be approximately written as a first order inverse differential kinematics equation. It is done by rewriting the following optimisation problem [9]:

$$\ddot{q} = \arg \min_{\ddot{q} \in \mathbb{R}^n} \frac{1}{2} \|\ddot{q}\|^2 \quad \text{subject to} \quad J\ddot{q} = \ddot{p} - \dot{J}\dot{q}$$

where the equation to be minimised can be written in the following way [9]:

$$\ddot{q} = J^+(\ddot{p} - \dot{J}\dot{q}) \quad (2.21)$$

since the pseudo-inverse has minimising properties as mentioned above. In discrete time with time step  $t = t_k$  the equation can be written as follows [9]:

$$\ddot{q}_k = J_k^+(\ddot{p}_k - \dot{J}_k\dot{q}_k) \quad (2.22)$$

Locally minimising the joint acceleration is equivalent to minimising the change in joint velocity from one time step to the next which in discrete time means taking the solution nearest to the previous value, which can be written as:

$$\ddot{q}_k \simeq \frac{\dot{q}_k - \dot{q}_{k-1}}{T} \quad (2.23)$$

Thus equation (2.1.5) can be rewritten in the following way based of equation (2.20) [9]:

$$\dot{q}_k = \dot{q}_{k-1} + J_k^+(\dot{p}_k - J_k\dot{q}_{k-1}) \quad (2.24)$$

The DLSI or the SVF jacobian inverse can be used interchangeably with the pseudo-inverse in these equations [6]. Because equation (2.24) is an approximation of control through acceleration written in terms of velocity control, the resulting control structure will hence forth be referred to as acceleration control.

## 2.2 Control Design

The control scheme is what enables the operator to give velocity commands that translates to movement of the arm. There are simple solutions, such as implementing PID-controllers, and there are advanced solutions that require an extensive model of the system. This section will introduce theory used when the control structures were designed.

### 2.2.1 Feedback and feedforward

A majority of the control systems are built on some kind of closed loop system, for example where a signal is used as feedback to compare with a reference signal and then a system, or subsystem is controlled using the error [7]. Feedforward skips one or more parts of the system to input a signal further ahead. These two can be used in combination, or by themselves [15].

### 2.2.2 PID-control

The Proportional, Integral, Derivative-controller, (PID), is the most commonly used controller for feedback in control [29]. There are controllers that are more optimal and based of optimal control, however they often require in depth knowledge about the system being controlled as in the need for a linearisation where a mathematical model must exist [14]. A PID-controller on the other hand requires no mathematical modelling of the system.

The PID-controller can be applied in variations with 1 to 3 of the parts. They are then called by the parts that are included, for example I-controller. The gain of the respective part can be chosen according to specific methods or by experimentation. Inappropriate values of the gains of a PID-controller can result in unstable behaviour [15].

### 2.2.3 Filtering

Filtering can be used for many reasons and in this thesis it was only used to smoothen the input signal from the operator. To filter the signal a transfer function can be used, which is a mathematical function giving the corresponding output value for each input value [15]. With the aid of a transfer function a signal affected by noise can become smooth for example.

A transfer function is generally written as:

$$G(s) = K \frac{(s + z_1)(s + z_2) \dots (s + z_n)}{(s + p_1)(s + p_2) \dots (s + p_n)} \quad (2.25)$$

where  $z_i$  are the roots of the numerator, which are called the zeros of the transfer function, and  $p_i$  are the roots of the denominator, where  $i = 1, 2, \dots, n$ ,  $s$  is the Laplace operator and  $K$  is the gain. To filter a signal it can be quite simple, for example:

$$G(s) = \frac{K}{(s + p_1)(s + p_2)} = \frac{K}{s^2 + 2p_1p_2s + p_1p_2} \quad (2.26)$$

which can be compared with the characteristic second order polynomial:

$$s^2 + 2\xi\omega s + \omega^2 \quad (2.27)$$

where  $\xi$  is the relative damping and  $\omega$  is the undamped natural frequency. Practically this means that  $\xi$  determines the shape of the response and that  $\omega$  determines the response speed. A transfer function can thus be created by choosing the desired behaviour through  $\xi$ ,  $\omega$  and the gain  $K$  [29].

### 2.2.4 Acceleration vs Velocity control

A control structure based of velocity control cannot quite control the acceleration other than through passive adjustments. Passive adjustments could for example be that the change in the velocity's magnitude is monitored or that hard limits to acceleration are introduced in sub-blocks of the system. To control both acceleration and velocity a control structure built based of second order differentiation is required. In the scope of the thesis there are two limits, one in velocity and one in acceleration, and in chapter 4 velocity and acceleration control will both be investigated to give the answer to the following question: will velocity control fulfil the requirements of the system, or will acceleration control be required?

### 2.2.5 Control for safety

With an operator standing beside the arm there can be no dangerous regions with impossibly high angular velocities or accelerations. The control structure must ensure that the velocities and accelerations are within reasonable boundaries at all times. Furthermore, there cannot be areas that the operator actively has to avoid. In order to make the heavy machinery safer the entire reachable space of the arm will have to be safe to manoeuvre. This is where DLSI or SVF can enable the robot to be operated robustly in the proximity and even when passing through singularities.

The limitation of acceleration and velocity can be done in several ways, such as introducing a feedback loop that ensures that the right levels are kept; by hard coded limits to big changes in movement; by limiting the gain of the input signals; among other methods. For example, when inside the singularity region the damped least-square inverse can procure quite a large error which can result in dangerous wind-up in an I-controller, thus anti-wind-up might be required and a saturation of the controlling signal.

The control system should be designed in such a way that reasonable usage of the arm will not incur a risk for the operator or objects and humans in the vicinity.

## 2.3 Dynamic Simulation

Model-based design has been proven to decrease the cost of development as well as the development time and its purpose is to use the aid of models and software to design and test complex systems virtually before building the actual prototype [23]. Applying this approach the torques of the joints, the inertia of the entire system, the friction, among other things can be estimated beforehand.

### 2.3.1 Dynamic properties

The dynamic properties of the systems can be estimated by using dynamic simulators such as Adams [2] or Simscape Multibody [27] and both of these system can be linked with Simulink [28] in order to design the control system and simulate how the system behaves with differential kinematics. There are advantages and disadvantages with both of the aforementioned dynamic simulators where Adams has a strong visual experience similar to CAD-based software and Simscape instead has a way of modelling using blocks.

Inertia is the resistance of inducing a motion in a body at rest. Even without friction or other forces involved, as long as there is mass there will be inertia that resists the change in motion. Since Liongrip has an arm where the weight is taken by the pneumatic actuator in the tower the inertia is a sizing factor for the torques required of the actuators. Inertia,  $I$ , not only depends on the mass, but also on the shape of the object, however for a particle the equation for inertia is as follows:

$$I = mr^2 \quad (2.28)$$

where  $m$  is the mass of the object and  $r$  is the distance to the axis of rotation [24]. Calculating the inertia one first calculates the inertia when rotating around the objects centre of mass that can be denoted by  $I_{cm}$ , and then calculates the inertia,  $I$ , around the axis of interest through the parallel axis theorem, also known as Steiner's Theorem, as follows:

$$I = I_{cm} + md^2 \quad (2.29)$$

where  $m$  is the mass of the object and  $d$  is the perpendicular distance to the preferred axis [1].

A higher inertia and/or a higher angular acceleration requires a higher torque to produce the commanded movement. The relationship between torque,  $\tau$ , inertia, and angular acceleration,  $\dot{\omega}$  is given below:

$$\tau = I\dot{\omega} \quad (2.30)$$

The inertia of the system can be affected by changing the design of the arm, changing the material properties for example, however the arm still requires to withstand the weight of the load. The torque can however be reduced with the control structure by reducing the acceleration.

Another physical property that will affect the movement of the arm is friction in the joints. A simplified friction model calculating friction is as follows:

$$F_f = \mu F_n \quad (2.31)$$

where  $F_f$  is the frictional forces,  $\mu$  is the friction coefficient and  $F_n$  is the normal force [22]. Friction consists of two parts, it's the static friction that must be overcome

to induce a movement, and it's the kinematic friction that is a resisting force that acts on the moving body after the static friction has been overcome. They share the same relations displayed in equation (2.31), however with different definitions of  $F_f$  and  $\mu$ . For static friction  $F_f$  is the critical friction force that needs to be overcome before movement can happen, and for kinematic friction  $F_f$  is the force that resists movement.  $\mu$  is the friction coefficient to be used before movement for static friction, and after movement for kinematic friction and they are assumed to be almost equal to each other. In reality friction is more complicated than this, however since friction will not be the sizing factor this formula is assumed to be appropriate.

### 2.3.2 Disturbances

All systems have disturbances that affects the performance. The arm of Liongrip is known to produce an unwillingly drift when heavy loads are applied which could be due to some tiny clearances in the joints that creates a small angle that shifts the centre of mass down below the x-y-plane that Liongrip was designed to operate in.

Another disturbance in a system where an operator will control the movement by a handle at the end effector is the operator. When the operator steers the arm forces could be applied that do not correlate with the commands given. For example the operator could drag the arm when the movement is already at maximum velocity, which could make the arm surpass the limits. The control structure would need to be designed with the above mentioned disturbances in mind.

## 2.4 Hardware Properties

With the goal to provide examples to the hardware required for building a full scale prototype some properties had to be identified. Its important not to choose actuators for the joints that are too big since they will add unnecessary high inertia to the system, and it's also important not to choose actuators that are too weak since they could not have the capacity to move the load, even with the help of the operator. A possible hardware list could include actuators for the joints, processor for the control system, a power supply and sensors for the joints.

### 2.4.1 Actuators

There are three main types of actuators that can be used for a project like this and they are: Electric, pneumatic or hydraulic. The main ones used for robotics are the electric actuators due to their higher precision. Pneumatic actuators are not widely used in robotics because they are hard to control accurately with their unavoidably fluid compression error. Hydraulic servos has a high cost and are difficult to miniaturise [26] which is why both pneumatic and hydraulic servos will not be used in this work.

Servomotors are per say not a specific kind of motors. Instead it's an ordinary motor with a servo system that can control the motor with a feedback loop, through position, velocity, or similar. The servomotors gets supplied current and an encoder sends back the state. The driver that supplies the current takes the state and sends it back to the controller that checks if the state is what it should be and depending on the answer gives commands to the driver. All in all a good example of a closed loop feedback to control a specific state. The benefit from already having this closed loop system is that the state, more precisely the angular velocity for this work, is the only thing that needs to be supplied to the actuators and no special regulator has to be constructed, which reduces the complexity of the work [19].

There are a few things to keep in mind when purchasing the right motor for the application. It has to be able to spin at the right speed, to be able to output enough torque, to be able to take a small enough step and the internal inertia of the motor has to be compared to the inertia of the system.

Since servomotors usually output low torque and high speeds they are often used together with gearboxes to increase the torque. When browsing for motors there are often two values to regard, the rated torque and the peak torque. The first represents the average torque in a load cycle that can be outputted without the motors temperature increasing indefinitely. The peak torque is the max torque that the motors can output. In this application where the actuators will move the bulk of the load, but not all of it, the operator will have to help the actuators. This means that there is risk that the peak torque will be surpassed, however as long as the servomotor has a temperature sensor that will give a warning before reaching critical temperature this can be managed.

Given the shape of a three link planar arm the right speed of a motor can simplified be calculated by the target radial velocity which is given by the following equation:

$$V_{rv} = \omega r \quad (2.32)$$

where  $V_{rv}$  is the radial velocity,  $\omega$  is the angular velocity and  $r$  is the radius [10]. By picking one joint to investigate and locking the others the target angular velocity for each motor can be found. In reality all three actuators will work together to produce the linear velocity of the end effector so simulations would give more exact values, however the angular velocity calculated will be a rough estimation.

The possible resolution can be calculated using the equation for the arc length that states that the radius of a circle times the angle made by the arc is equal to the length of the arc. The radius would be the robotic arm, and the arc length would be how much the end effector moved by a step of the actuator. When really small steps are looked at this arc length can be viewed as a short straight line. This is interesting because the worst case scenario to take small steps are when the arm is fully outstretched and the actuator furthest away from the end effector is the only one to move. If different actuators, and/or different gearings are used for all the joints then these calculations might require calculations for each. The arc length equation is as follows:

$$\delta = L\alpha \tag{2.33}$$

where  $\delta$  is how much the tip of the end effector is moved,  $L$  is the length from the joint to the end effector's tip and  $\alpha$  is the angle of the joint measured in radians. When taking the gearing for the specific joint,  $i$ , into account the equation can be written like this [3]:

$$\delta = \frac{L\alpha}{i} \tag{2.34}$$

Because the gearing will increase the rotation required by the actuator to produce the specific angle of the joint the angle is reduced by the gearing [3]. These calculations are conservative since all the joints will aid in making all the movements.

In order for the motor to perform efficiently one should also look at the inertia ratio which is the inertia of the load, divided by the internal inertia of the motor. If the inertia ratio is big the motor will have a harder time keeping the load in control. Gearboxes reduces the inertia of the load by the square of the gearing [18]. The ratio which should not be surpassed is in the span of 1-15:1 where a bigger ratio may give a system that is not in control of the load [13]. However in systems where the operator helps the arm in its movements, the gear ratio might not be a sizing property and physical experimentation might be required to get the ratio right.

## 2.4.2 Sensors

In order to design a motion control system for a robotic arm the actual angles of the joints must be acquired through sensors. There are several different sensors that could accomplish this, such as an incremental encoder, an absolute encoder or a resolver. The incremental encoder and the resolver knows where it is depending on where it has been, while the absolute encoder uses optics to now where it is now. The effect of this is that both the incremental encoder and the resolver requires a start up position coordinated with the code of the differential kinematics, and the absolute encoder will read the right position as soon as it's powered on [26]. If velocity is used to control, either a sensor or a driver used with the servomotor that can measure velocity would be preferred.

## 2.4.3 Safety incorporated into hardware

Heavy machinery working in close proximity to humans requires a high safety factor so as not to hurt anybody. The safety can be incorporated in both the hardware and the software. One of the basic properties to include is a break in order to quickly stop the arm if a movement is at the risk of personal harm. Following a break an emergency button that will activate the breaks and turn off the power to the system can be important. There are several ways that a human can be hurt by the arm for example: Crushing or hitting. Crushing could be between the arm and another

object or in between the arm, and depends on the torque the actuators can output. Hitting could be when the arm has either a high velocity or high acceleration and collides with a human. To limit the amount of torque the actuators can output will also limit weight a lifting aid can handle and the torque used to move the load will potentially be dangerous even at the low end of the spectrum. The actuators and gearing could be chosen in such a way that the top velocity is limited by the hardware.

The emergency button could be extended to be a long button at the sides of all the arm links. If the arm hits an object or a human the button will be pushed by the movement, the arm will power-off and the breaks will be applied. These kind of switches are often called bumpers.

### 2.4.4 Bumper

Bumpers are long switches that usually are placed in areas to increase safety. They can be used to cut the power to the machinery when they are activated which in many cases are enough to prevent an accident. They are often made up of a rubber list that can be quite a bit protruded from the base so that the bumper can cut the power at the first touch. If an object where to come in contact with a bumper on a machine the perturbation creates a distance that the rubber list can deform before the object and the machine would touch, thus increasing the safety in-case the machine continuous to move for a time after the power is cut [12].

## 2.5 Programming

Programming language and microcontroller will be decided in unison, because many microcontrollers only have support for C-code for example. Some controllers have such a high performance that they can run some Python versions, usually micro-python [16]. The control structure proposed in this thesis will be computational light enough to run on a microcontroller and it will be programmed much like a block diagram of the control structure, see for example figure 4.13.

An important area will be how a sensor and the microcontroller will talk to each others. For example how the signal from the sensors for the joints or the command signals given by the operator will be read. The system will be programmed in a way that resembles Real Time Operating System (RTOS), since the sensor must be read and up to date before the movement calculations is performed. RTOS are operating systems that are used for multitasking where timing is critical. In the programming for the movement of Liongrips arm a scheduler will be used to ensure that the different blocks are run at a acceptable frequency and semaphores to ensure that the sensors signals are read before the movement is calculated and not during.

# 3

## Modelling of Liongrip

Section 3.1 presents the robotic configurations for Liongrip, section 3.2 presents the modelling and section 3.3 is about the hardware selection and requirements.

### 3.1 Kinematic description of Liongrip

It was important to define the arm in the way that is done in the Robotics world in order to benefit from the vast amount of research already done in that field and as such the arm of Liongrip was modelled and described as a three link planar arm capable of movement only in the horizontal plane where all three joints can do a yaw rotation. The degrees of freedom for the arm of Liongrip was 3 and the end effector could perform two types of linear movements, x-wise and y-wise, and a rotational movement, yaw rotation. Since there were only three degrees of freedom for the end effector of Liongrip and three joint variables that described the system the arm was non-redundant. In figure 3.1 the end effector frame can be viewed where the z-axis is omitted since this thesis only considers the movement in the xy-plane. The origin of this frame was placed at the tip of the end effector, it was thus possible that any change of gripping tool may have resulted in an origin that was not placed at the tip. This placement might require further research and testing on a physical prototype so that it is placed in a way that is intuitive to an operator. The positive direction of y was forward and the positive direction of x was moving to the right while facing forward.

The links and the joints of Liongrips arm are displayed in figure 3.2. Note that the middle links two joints, joint 2 and joint 3, were geared together and the angle they created had its origin in the middle of them.

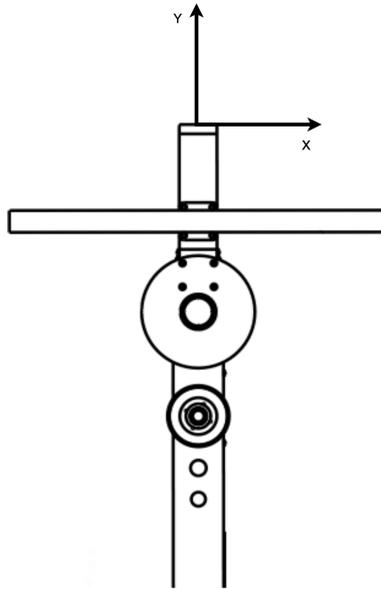
#### 3.1.1 Differential Kinematics

The arm was controlled using inverse differential kinematics where the operator could give three signals: linear velocity in x-direction,  $\dot{x}_e$ , and y-direction,  $\dot{y}_e$ , for the end effector and the rotational speed of the outermost joint,  $\omega_e$ . Hence the input signals was the vector  $\dot{p} = [\dot{x}_e, \dot{y}_e, \omega_e]$ .

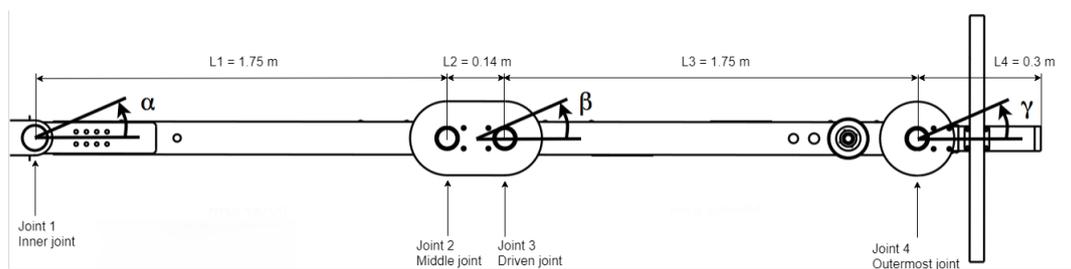
Complete calculations for the Jacobian can be found in appendix A.1. With equation (2.3) and (2.2) that shows how the partition fits together the Jacobian defined in origin,  $J_o$ , was calculated to:

### 3. Modelling of Liongrip

---



**Figure 3.1:** The defined coordinate system for the end effector



**Figure 3.2:** A schematic view of links and joints for the arm of Liongrip

$$J_o = \begin{bmatrix} -a_1c_1 - a_2c_{12} - a_3c_{123} & -a_1c_1 - a_2c_{12} & -a_1c_1 \\ a_1s_1 + a_2s_{12} + a_3s_{123} & a_1s_1 + a_2s_{12} & a_1s_1 \\ 1 & 1 & 1 \end{bmatrix} \quad (3.1)$$

where  $s_1$  is short form for  $\sin(\alpha)$ ,  $s_{12}$  is  $\sin(\alpha + \beta)$  and  $s_{123}$  is  $\sin(\alpha + \beta + \gamma)$ .  $c_i$ , where  $i \in 1, 12, 123$ , is representing the same but for cosine.

The Jacobian given in (3.1) was defined in the base frame, o, which was placed at the innermost joint. This was not practical since the operator would be beside the end effector and therefore the velocities should be defined in that frame. This was done by taking the rotation matrix from the base frame to the end effector,  $R_e^o$  and multiplying it with the Jacobian  $J_o$ .

$$J_e = R_e^o J_o \quad (3.2)$$

where  $R_e^o$  :

$$R_e^o = \begin{bmatrix} c_{123} & s_{123} & 0 \\ -s_{123} & c_{123} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

The Jacobian,  $J_e$ , used can be viewed in appendix A.1.

### 3.1.2 Singularities

Finding the singularities of a three link planar arm was done by finding all the angles where the Jacobian loses rank. This was done by setting the determinant of the Jacobian equal to 0:

$$\det(J_e) = 0 \quad (3.4)$$

and solving the resulting equation (3.4) by using trigonometric formulas as shown in appendix A.2 the resulting relation was found:

$$\sin(\beta) = 0 \quad (3.5)$$

which was true for  $0, n\pi$  where  $n = 1, 2, \dots, \infty$ . Therefore when the middle arm was parallel with the inner arm a singularity was reached. The calculations can be viewed in appendix A.2.

Liongrip is sometimes used in tight spaces where it can be beneficial to be able to choose in what direction the combined joint of joint 2 and 3 will rotate and that can only be changed if the arm is outstretched. Such movement would allow the arm to pass through a singularity. The arm of Liongrip is often folded to make room, which

leaves the arm in a position of a singularity. With the inertia in the motors and the gears it might be physically demanding, or impossible, to take the arm through these areas with no help from the actuators.

If the singularities were to be avoided it would mean that the outermost ring was not made available to the operator and would have to be added as a constraint to the workspace of Liongrip, as well as the position where the arm was completely folded. This was not desirable and research was done to handle this with DLSI or SVF. The control structures and parameters used for them can be viewed in chapter 4.

#### 3.1.3 Safety measures

When designing the control structure safety was considered and by analysing the responses of the velocities and accelerations and adjusting the parameters any oscillations were limited in order to avoid risk. Even though oscillations did not always impact stability, it could induce vibrations or even small movements in the arm that could disrupt the operators concentration or even lead to unintended commands. Oscillations resulting in movements could also trigger possible emergency breaks when the arm was controlled in tight spaces or cause damage to nearby equipment, making it impossible to achieve the precise movements that was needed of the operator.

The robust methods of DLSI and SVF were used to increase safety and to limit any kinds of dangerous velocities or accelerations due to singularities. Several iterations of the simulations were done to tune the damping factors of  $\lambda$  and  $\sigma_0$  until there were no hazardous movements induced by the troublesome regions.

## 3.2 Modelling, Simulation and Experiments

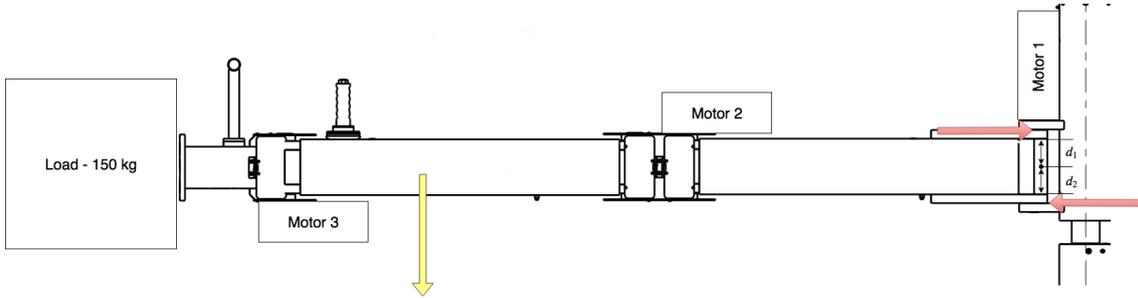
In order to identify the actuator requirements an iterative process was employed that used models and simulations. Note that the simulation software of Simscape Multibody [27] provided the torque in the joints and hence a complete mathematical model for the calculations of torque in the joints was not required. Simscape Multibody was used in cosimulation with Simulink to simulate the dynamics of the arm together with the control structure. Friction and worst case inertia of the entire system were modelled outside Simscape using mathematical formulas. Lastly the smallest step required by a joint to achieve the target resolution was calculated.

### 3.2.1 Modelling

Simscape Multibody has no inherent way to calculate friction so a basic friction model was created using Coulomb-equation for dry friction [22]. The arm of Liongrip was designed to have a reasonable friction in the joints in order to enhance the controllability and provide slow movements for an operator, however with the

motors in place this controllability was left to the control system instead. The friction coefficient of 0.2 resulted in an unnecessary high friction torque in the joints of 96, 65 and 8 Nm, from innermost to outermost joint, which led to the decision to switch to bearings with a lower friction coefficient. The resulting friction torque was ranging from 2, 1.3 and 0.15 Nm, from innermost to outermost joint.

The method to calculate the friction were as follows: the arm was locked in outstretched mode, the mass centre was estimated, see figure 3.3, basic torque relations was used to find the reaction forces in the joint of the arm, and through Coloumb friction the resulting friction forces was calculated. The calculations for the reaction forces were done in the 2D-plane of the outstretched arm and they were used as the normal force for the friction calculations. The yellow arrows in figure 3.3 represents the force,  $F_m$  from the mass acting on the arm in the mass centre and the two red arrows,  $F_{r1}$  and  $F_{r2}$ , indicate the reaction forces from the bending moment provided by the mass. Rectangles were added to represent the placement of the motors where motor 1 did not contribute any weight since it was assumed fastened in the tower.



**Figure 3.3:** Active force diagram to calculate friction in the joints of Liongrips arm

The formula for the equilibrium of the torques are provided in the equation below:

$$F_m d_m = F_{r1} d_1 + F_{r2} d_2 \quad (3.6)$$

where  $d_m$  is the distance from the innermost joint to the mass centre,  $d_1$  and  $d_2$  is the distance from the point of respective reaction force to the middle point between them, which was chosen as the pivoting point. Because of the placement of the pivoting point  $d_1 = d_2$  and since equilibrium must hold in this static model  $F_{r1} = F_{r2}$  the following simplification can be made:

$$F_m d_m = 2F_{r1} d_2 \quad (3.7)$$

By solving for  $F_{r1}$  and using Coulomb friction relation of  $F_f \leq \mu F_n$  [22] the friction force,  $F_f$  was identified through the following formula:

$$F_f = F_{r1} \mu = \mu \frac{F_m d_m}{2d_2} \quad (3.8)$$

To calculate the torque friction  $\tau_f$  the friction force was multiplied by the motor shaft radius,  $r_m$ :

$$\tau_f = F_f r_m \quad (3.9)$$

The calculations for the middle and outermost joint followed the same formula with only the contributing forces left and the reaction forces moved to the corresponding joint. The simplification was made that the middle joint friction torque was doubled due to the fact that it also drove joint 3.

By using Simscape Multibody the inertia around the centre of mass,  $I_{cm}$ , was calculated and by using the Parallel axis theorem [1] the rotation of the object was moved to coincide with the z-axis of the innermost joint. Given the properties of moment of inertia it was then summarised to become the inertia of the arm when revolving around the z axis of the innermost joint. The calculations were done having the arm fully extended since this was the worst case scenario for the innermost joint. The following inertia applied to the specified joints were found:

- Innermost joint - 145  $kgm^2$
- Middle joint - 123  $kgm^2$
- Outermost joint - 62  $kgm^2$

Since the friction torque corresponds to a relatively small contribution, the Inertia together with the angular acceleration were the main consideration for sizing the motors. The inertia was also considered when sizing the actuators with regards to the inertia ratio.

By using equation (2.33) the minimum resolution of the joints revolution could be calculated [3]. The smallest possible step  $\delta = 1$  mm, the entire length of the arm,  $L = 4300$  mm and the angle  $\alpha$ , measured in radians, was unknown. By applying algebra the angle could be calculated by the following equation:

$$\alpha = \frac{\delta}{L} = \frac{1}{4300} \quad (3.10)$$

where  $\alpha$  is measured in radians and  $2\pi$  radians equals a revolution hence this fraction were to be multiplied by  $\frac{1}{2\pi}$  to give the smallest step measured per revolution,  $s$ .

$$s \approx \frac{1}{27004} \quad (3.11)$$

## 3.3 Hardware

To build a prototype the right hardware was to be chosen and for that the sizing requirements and the required properties of the system was to be identified. By bringing the operators physical force into the picture the requirement on the torque was relaxed some: if it was possible to move the weight by hand, then it would also be possible to move the weight with the added help from actuators. In the iterative process there were simulations done, hardware selected, the system rework

to included lower torques, simulations reworked and hardware selected again, all with the help of SWE-EURODRIVE [25] that produces electric actuators and servo systems. These discussions lead to a final proposal based on the data in the following sections and the results is presented in section 5.2.

### 3.3.1 Hardware from SEW

Discussions were made together with SEW to try and find actuators with gearing that were suitable to this work. The sizing parameters were torque and the inertia ratio between the arm with load and the actuators. Following the hard requirements a combination using breaks, temperature sensors and absolute encoders were proposed.

### 3.3.2 Hardware requirements

The main hardware to select was the actuators because they require specific sizing to be strong and fast enough, and they need a driver to be controlled and this driver would set requirements on the microcontroller on how to communicate. From early simulations using a basic control structure an approximation of the required torque was made and used in the discussions with SEW-EURODRIVE. The desired properties for the actuators can be viewed in table 3.1:

	Joint 1	Joint 2	Joint 4	Requirement?
Peak Torque [Nm]	120	121	9	No
Average Torque [Nm]	19	21	1.5	No
Inertia [kgm <sup>2</sup> ]	145	123	62	No
Temperature sensor	Yes	Yes	Yes	Yes
Absolute encoder	Yes	Yes	Yes	Yes
Break	Yes	Yes	Yes	Yes
Servo control	Yes	Yes	Yes	Yes

**Table 3.1:** The desired properties on the actuators

The requirement column is due to the uncertainty introduced by the operator aid to the movement because choosing actuators is no loner as simple as identifying one that, for example, is strong enough. The peak Torque is the maximum torque that the actuator can output. With no downtime average torque approximately correlates to the rated torque which is the average torque that should not be exceeded during a load cycle in order to keep the temperature of the motor from rising towards infinity.

Since the operator was to provide physical force to help the system the topmost three properties were not requirements, instead they were viewed as guidelines. The temperature sensor, the absolute encoder and the break was requirements. The temperature sensor would be used to send a warning if the actuator was overheating, and thus it was vital for a system where the operator would provide physical force to aid the movement. The absolute encoder was required to avoid a start up sequence

just for the arm to know its angles. The break was a requirement to help ensure safety. Servo motors were required so that the input could be angular velocity without the need to build a separate control system for the conversion of the signal. The internal feedback of servomotors was also required to internally handle unwillingly drift or similar external disturbances that could result in displacement, which the servomotor would compensate for.

With safety in mind the actuators were to be chosen in such a way that their RPM together with the gearing were limited to the angular velocities corresponding to the velocity boundaries of the end effector. This was a way to introduce redundancy, where velocities were limit by the control structure and the hardware. By looking at the angular velocity of the joints in the simulation in section 4 was set and used as a guideline when choosing the RPM and gearing of the actuators.

To further increase the safety bumpers were to be added to the sides of the inner arm, the outer arm, and the end effector. When the bumper were deformed the contact from the power supply to the actuators would be cut, and when the current were cut to the motors the breaks would activate. As long as the breaks are able to stop the movement fast the clamping would be kept at a minimum. This would also increase safety for nearby humans and decrease the risk of the arm breaking itself or other nearby equipment.

# 4

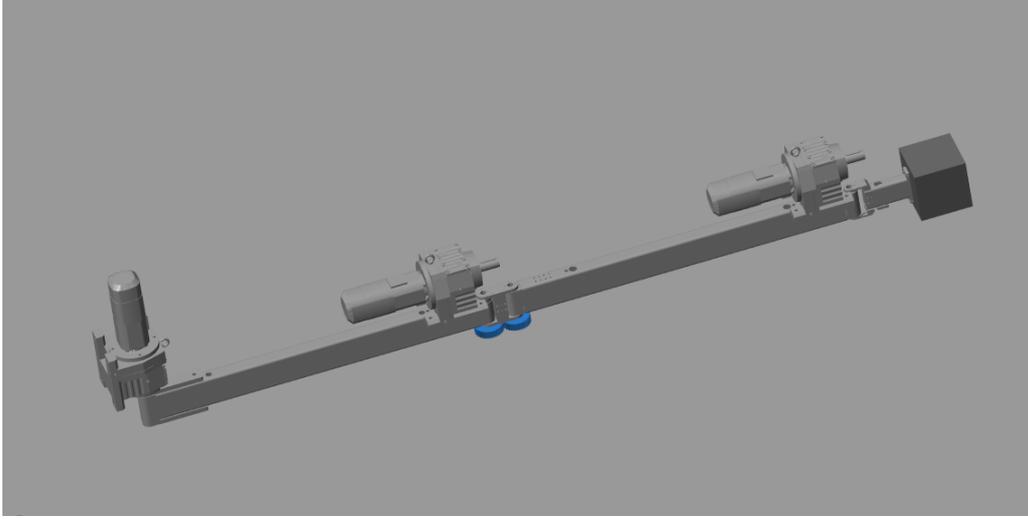
## Comparison of singularity-robust methods

In the following sections the control structure used for the simulations are displayed along with interesting parts of the result to highlight positive and negative features of the respective system. Section 4.1 presents some general information about the simulations, section 4.2 presents the simulations when an open-loop system with velocity control was used, section 4.3 displays the simulations when a closed-loop system with velocity control was used, and section 4.4 demonstrates the simulations done using acceleration control.

### 4.1 General information

The dynamic simulations were performed using the software Simscape Multibody [27] where the CAD-model provided by Hofpartner AB was imported. The control system was simulated using Simulink which was connected to the dynamic simulations of Simscape Multibody. The mass for each part was calculated by the internal calculator of the software based of dimensions and the type of material used. The material was set to steel with a density of  $7.85 \text{ kg}/m^3$  [5].

In Simscape Multibody the dynamic model was built using the same block methods as Simulink uses and the joints of the CAD-model were actuated. Since Liongrip's costumers aim at lifting a large variety of objects with the arm, a massive cube was added at the end effector to simulate that an object of 150 kg was lifted. The control signals for the joints were angles, which was not on par with the theoretical system built which used velocities as input to the actuators. This was solved by using a Simulink numerical integrator block. To make realistic simulations motors with gearing were added to the model. The models were provided by SEW-EURODRIVE and were part of one of their suggestions. No connections of the motor shafts and joints were made so the addition was purely for the change of weight and of inertia. The simulation model can be viewed in figure 4.1.



**Figure 4.1:** The 3D model resulting Simscape Multibody complete with actuators

Following the limitations introduced in section 1.4 a maximum velocity of  $0.354 \text{ m/s}$  and maximum acceleration of  $0.283 \text{ m/s}^2$  were used as part of the evaluation of the control structure measured using a Simscape Multibody sensor block. The systems were evaluated based of the velocity error,  $\dot{e}$  defined as:

$$\dot{e} = \dot{p}_{filtered} - \dot{p}_{actual} \quad (4.1)$$

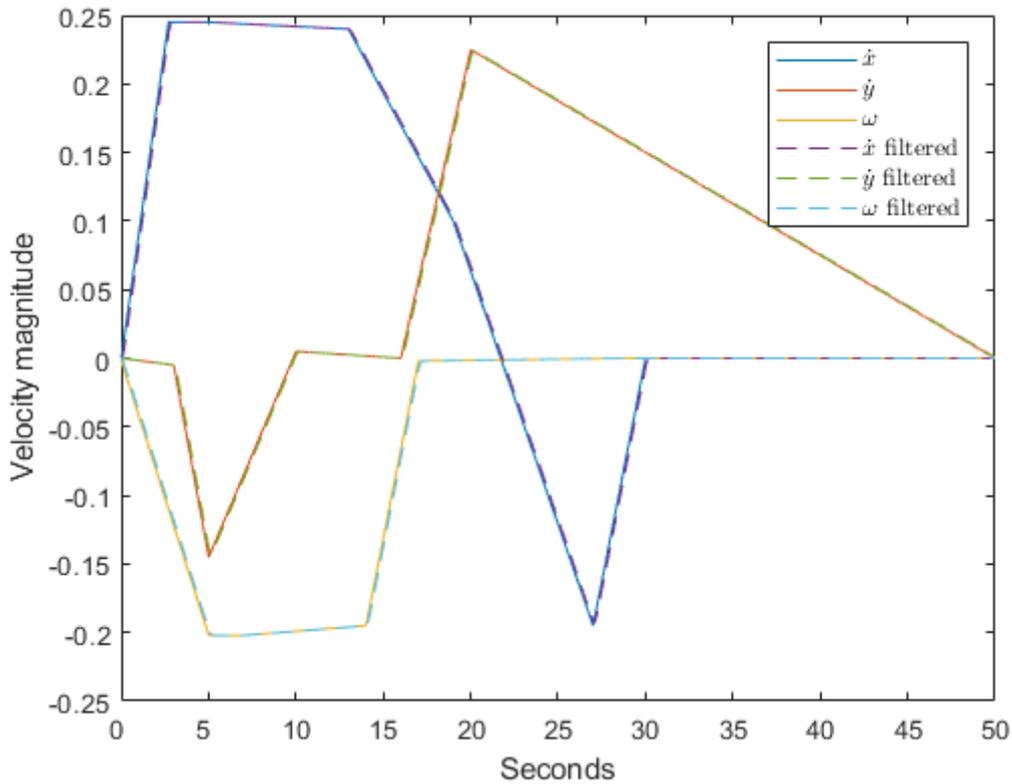
where  $\dot{p}_{filtered}$  is the filtered vector of the operator's input,  $\dot{p} = [\dot{x}_e, \dot{y}_e, \omega_e]$ , where  $\dot{x}_e$  and  $\dot{y}_e$  were linear velocities along x-direction respectively y-direction, and,  $\omega_e$  the angular velocity, and  $\dot{p}_{actual}$  is the actual velocity. All systems presented performed inside the boundaries of the maximum linear velocity and acceleration for the end effector.

The filter used corresponds to the following transfer function shown:

$$\frac{10}{s^2 + 10s + 100} \quad (4.2)$$

The purpose was to make the control signals smoother and as a consequence the movement of the arm smoother. In (4.2) a damping,  $\xi$ , of  $\frac{1}{2}$  and a natural frequency,  $\omega$ , of 10 was used.

Figure 4.2 shows the input signal used in the simulations together with the filtered signal. The filtered signals are portrayed with dashed lines, while the original signals are solid lines and the visual differences between the signals are small. Because of the small difference between the filtered and the unfiltered input signal the commands from an operator will most likely not be noticeably changed, however smoothening of the filter did remove some acceleration spikes.



**Figure 4.2:** The input before and after the filter. The input signals are displayed in solid lines and the filtered signals are shown in dashed lines

The two most important differences between Liongrip and the majority of the industrial robots were: a) the trajectory was not known before hand, that means that the control system had to handle everything online. And b) the operator would be there as a human trajectory planner and feedback loop, that could manually correct the trajectory and help the system to move to the desired configuration. This paradoxically both relaxed and increased the demand on the control structure, because the error could be larger and constantly adjusted by an operator, but no optimal control of a specific trajectory could be made.

All simulations were done with methods that could safely control inside and in close proximity to singularities. Several variations of systems using Damped least-squares inverse, Singular Value Filtering, velocity control and acceleration control were compared.

The different parameters used can be viewed in table 4.1 where O-l stands for Open-loop, Fb for feedback, fwd for feedforward, acc for acceleration control. The systems that are not marked for acceleration control used velocity control.

	P/I	$PI_{limit}$	$K_b$	$\lambda$	$w_t$	$\sigma_0$	$\nu$
O-l, DLSI	-	-	-	3	2.5	-	-
O-l, SVF	-	-	-	-	-	1.1	1.8
Fb, DLSI	-/1.5	0.25	200	4	2.9	-	-
Fb, SVF	1.5/13.5	0.5	200	-	-	1.1	1.8
Fb, fwd, DLSI	-/1.5	0.25	200	3.5	3	-	-
Fb, fwd, SVF	1.5/13.5	0.25	200	-	-	1.2	1.5
O-l, acc, DLSI	-	-	-	5	3	-	-
O-l, acc, SVF	-	-	-	-	-	0.3	6.6
Fb, fwd, acc, DLSI	1.5/0.5	0.15	200	3.5	10	-	-
Fb, fwd, acc, SVF	-/11.5	0.15	200	-	-	0.3	6.6

**Table 4.1:** The parameters used per system

The variables are as follows: P/I is the gain used for the proportional- and integral-part depending on the controller used;  $PI_{limit}$  is how much the control signals were limited;  $K_b$  is the back-calculation gain that controls how the signal behaves when anti-wind up is activated and the signal shall move from a saturated state;  $\lambda$  is the damping factor used in DLSI;  $w_t$  is the threshold value for when to use the damping in DLSI;  $\sigma_0$  is the minimum value added to the singular values when SVF is used and lastly  $\nu$  is a shape factor chosen to keep the function  $h_{\nu,\sigma_0}(\sigma)$  monotonic.

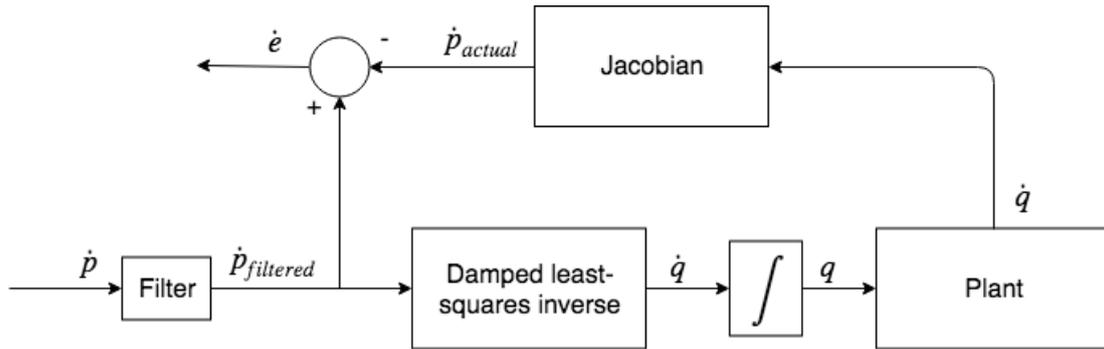
### 4.1.1 The input signal

The simulations used the same input signals to make them comparable. It was one where the arm started fully outstretched, at a singularity, then bent to an almost completely folded position, in the vicinity to a singularity, and opened up again. Depending on how large the error was for the utilised control structure this may not have been the actual movement performed by the robotic arm.

Furthermore the input was predefined to be close to ideal, with the added filter to make it smoother. This was done in order to focus on the control structure first and leave problem areas to future improvements. There were no sudden changes in the input, for example if  $\dot{x}$  would be 0.25 m/s in one instance and -0.25 m/s in the next, and there were no commands that tried to execute movements that were not possible, for example trying to move the arm further away when it was fully outstretched.

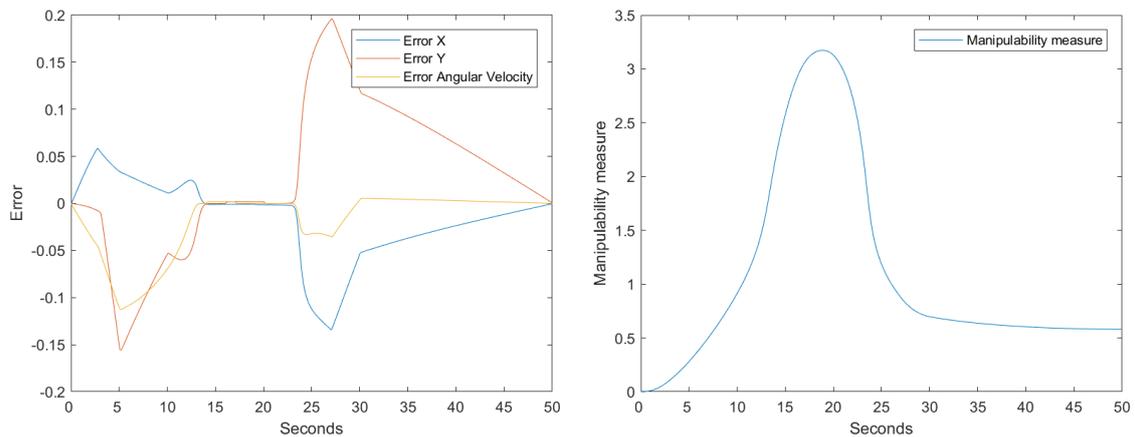
## 4.2 Open-loop

A system based on the Singular Value filtering with an open loop where the signals from the operator went directly through the system were tested first in order to examine how the system responded and behaved. The used control structure can be viewed in figure 4.3 where the Plant was the dynamic model made in Simscape Multibody.



**Figure 4.3:** A block diagram view of the open loop system using DLSI

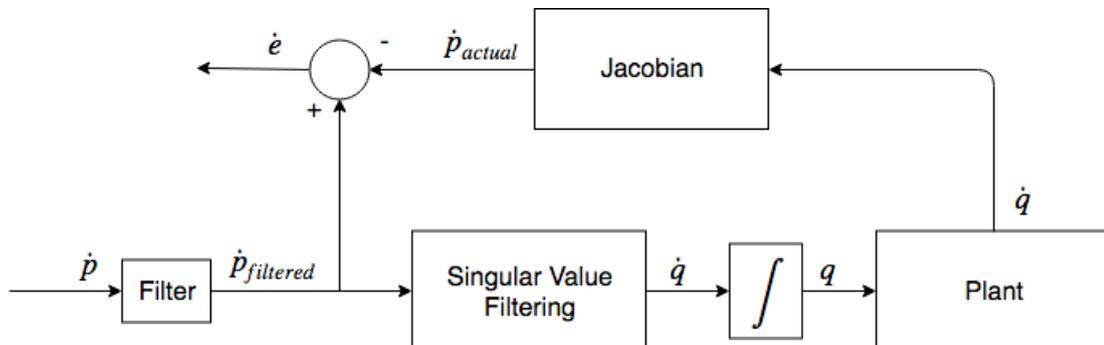
In figure 4.4a quite large errors of the system are displayed, where the global maximum of the error were 80 percentage of the max input signal. In figure 4.4b one can see the manipulability measure, where a lower value on the manipulability measure means the system was closer to a singularity.



(a) Velocity Error

(b) Manipulability Measure

**Figure 4.4:** Simulation of the open loop DLSI system

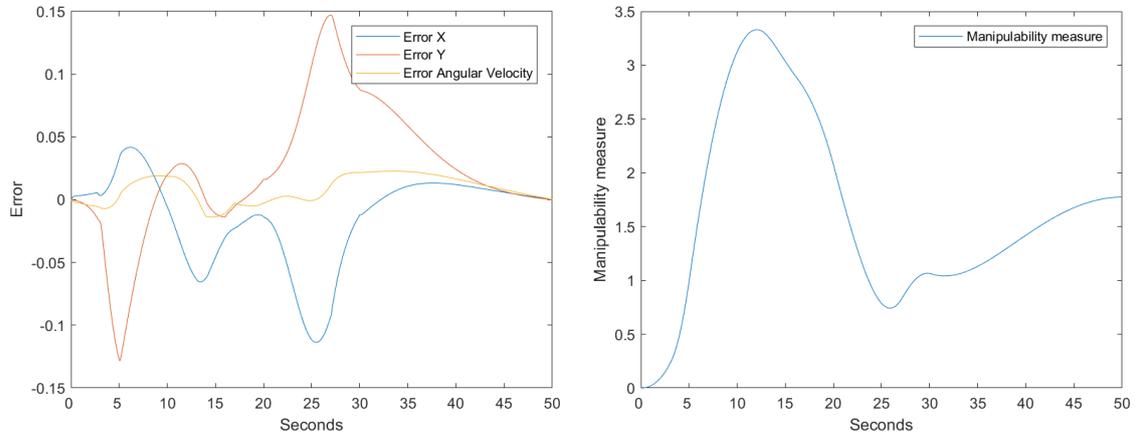


**Figure 4.5:** A block diagram view of the open loop system using SVF

The corresponding control structure based of SVF can be viewed in figure 4.5 and the error for that system is shown in figure 4.6a. The global maximum of the error

#### 4. Comparison of singularity-robust methods

was lower, about 60 % of the maximum input, than in the open loop DLSI system, however there was no prolonged period with an error close to 0. In figure 4.6b the manipulability measure is displayed.



(a) Velocity Error

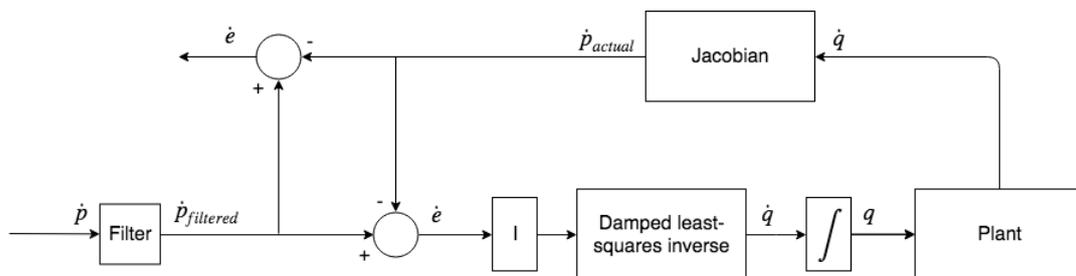
(b) Manipulability Measure

**Figure 4.6:** Simulation of the open loop SVF system

Both open loop systems had large errors that required reduction. SVF showed a smaller interval of errors, and the system seemed to have an overall higher manipulability measure most of the simulation time. Both systems required further investigation and a more advanced control system was required to reduce the errors.

### 4.3 Closed-loop

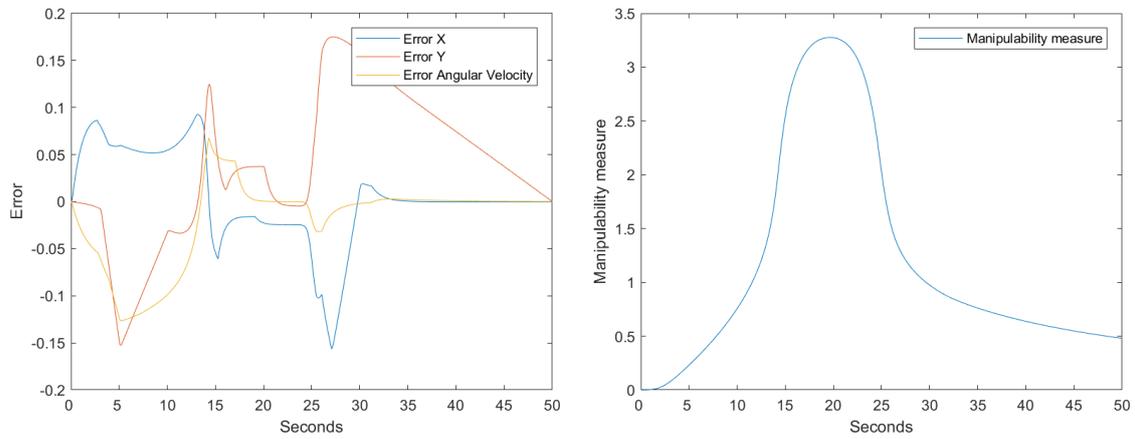
In order to help minimise the error a feedback loop can often be used in combination with a gain-controller, such as PID, PI, I, or similar. The P-part was required to be small for the system to be improved instead of deteriorating. Using feedback proved to be difficult for the systems using DLSI, where the results were worse than with the simple open loop. The control structure used for the feedback system with DLSI is shown in figure 4.7.



**Figure 4.7:** A schematic view of the control structure while using DLSI and feedback control

Errors that are continuously large over a longer period of time can make an I-controller ramp up the signal to hazardous levels. In worst case scenarios this behaviour can make the system unstable, which was unacceptable when ensuring the safety of the operator and as such the controller's output was limited and anti-windup used. The negative effect was that the controller could get slower at regulating big errors, which was not desirable in a system that was supposed to feel responsive and exact to an operator.

The feedback system had troubles with acceleration spikes which caused a less than advantageous regulation to keep within the boundaries. Combined with the limitation of the controllers signal could be the main reasons why the error, presented in figure 4.8a does not show evident improvements compared to the open loop system. In figure 4.8b the manipulability measure shows that the arm was close to a singularity at the majority of the simulation time.

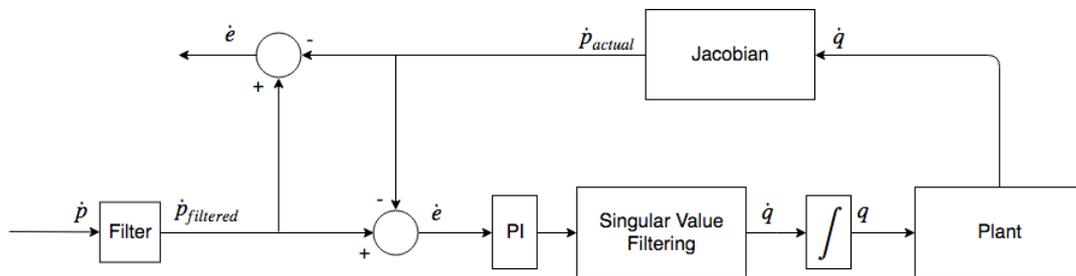


(a) Velocity Error

(b) Manipulability Measure

**Figure 4.8:** Simulation of the feedback loop DLSI system

The feedback system using SVF showed promising results and the control structure used can be viewed in figure 4.9.

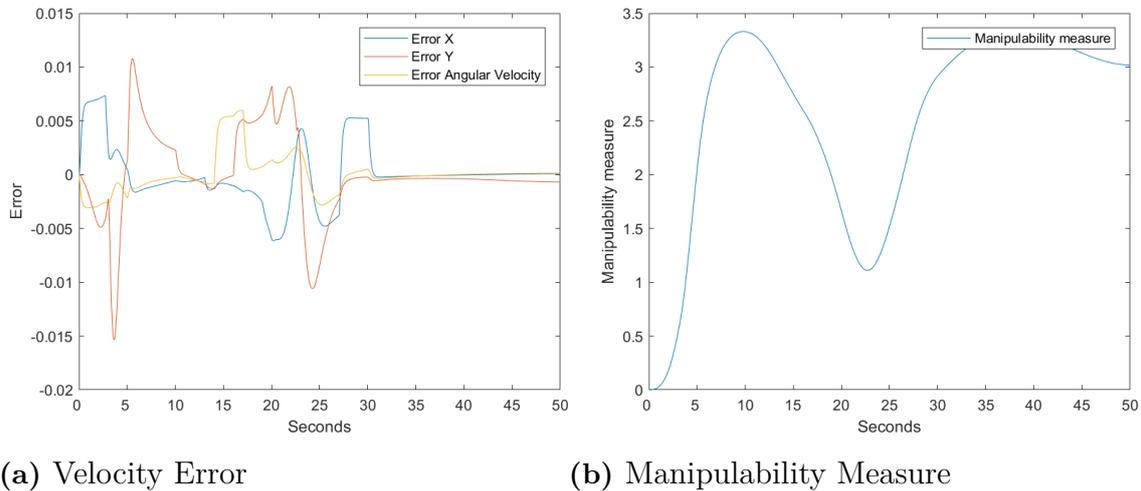


**Figure 4.9:** The system using SVF and a feedback loop

In figure 4.10a the error can be viewed and it showed improvements with a region with close to no error between and 30-50 seconds. Figure 4.10b can give a hint to the large errors from 0 to 30 seconds, because singular regions were nearby, it could

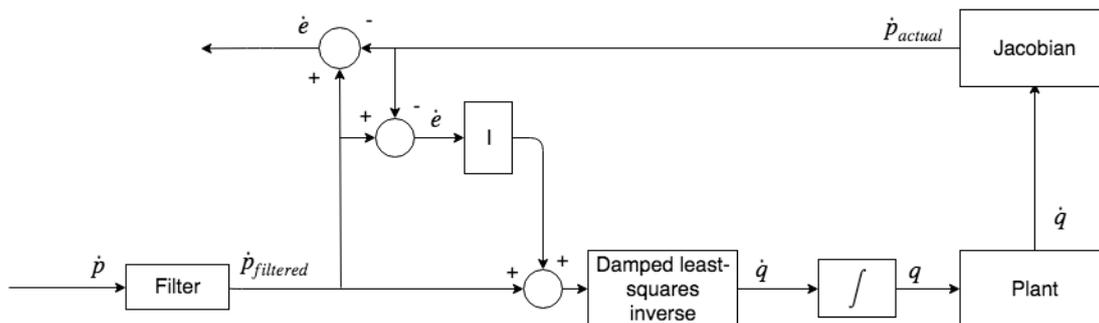
#### 4. Comparison of singularity-robust methods

also explain why the error went down at about the 14th second, since the arm had been outside a singularity for some time at that point. That the error reduction showed first when the manipulability measure was decreasing again, could be due to settling time, which means that it took time for the system to adjust and reduce the error.



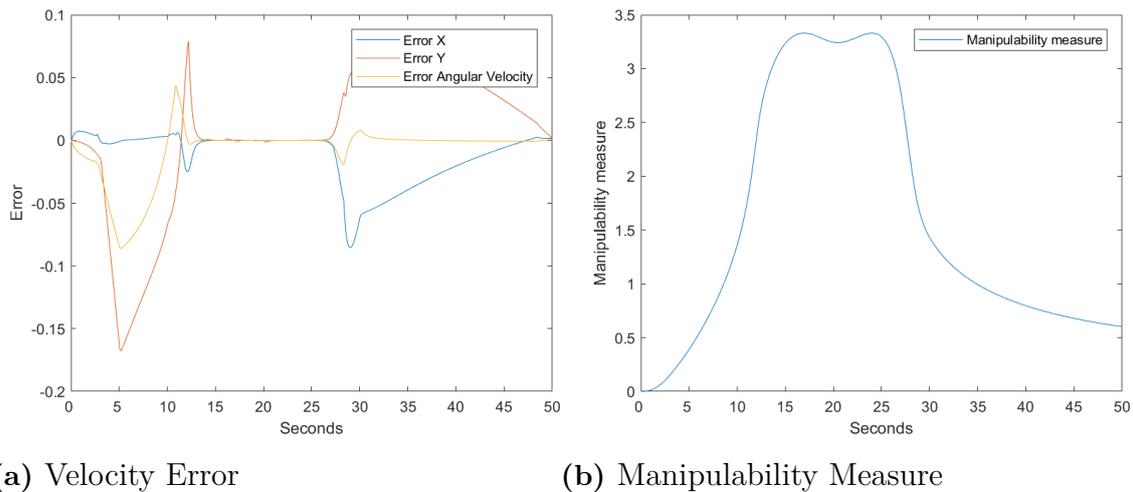
**Figure 4.10:** Simulation of the feedback loop SVF system

Further improvements of the structures were required before they could be thought of as potential candidates for a physical prototype, especially DLSI still required great improvements. Therefore a system using feedback while forwarding the control signal from the operator were tested. These kind of systems are usually better at handling quick changes in the control signals, which was an added benefit. An example of the DLSI-system using feedback and feedforward is shown in figure 4.11.



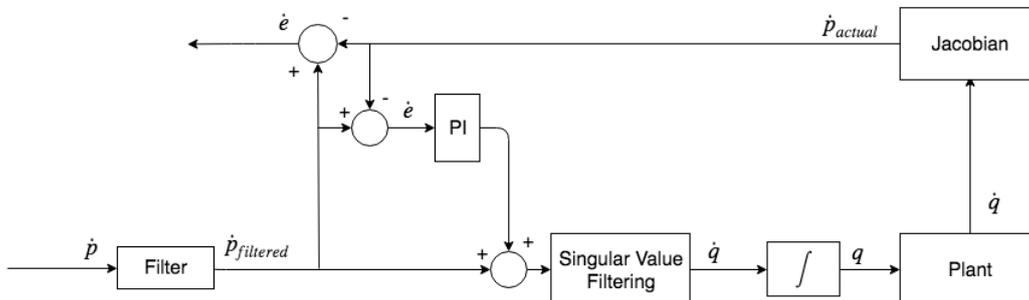
**Figure 4.11:** A block diagram of the system using feedback, feedforward and DLSI

The resulting error showed some improvements and can be viewed in figure 4.12a while the manipulability measure can be viewed in figure 4.12b. In this simulation the DLSI had an overall high manipulability measure which correlates with the error that was close to 0 at 13 to 26 seconds. However the system did not reduce the error in the vicinity of a singularity. The system produced acceleration spikes and required sub-optimal regulations to be within the bounds.



**Figure 4.12:** Simulation of the feedback, feedforward DLSI system

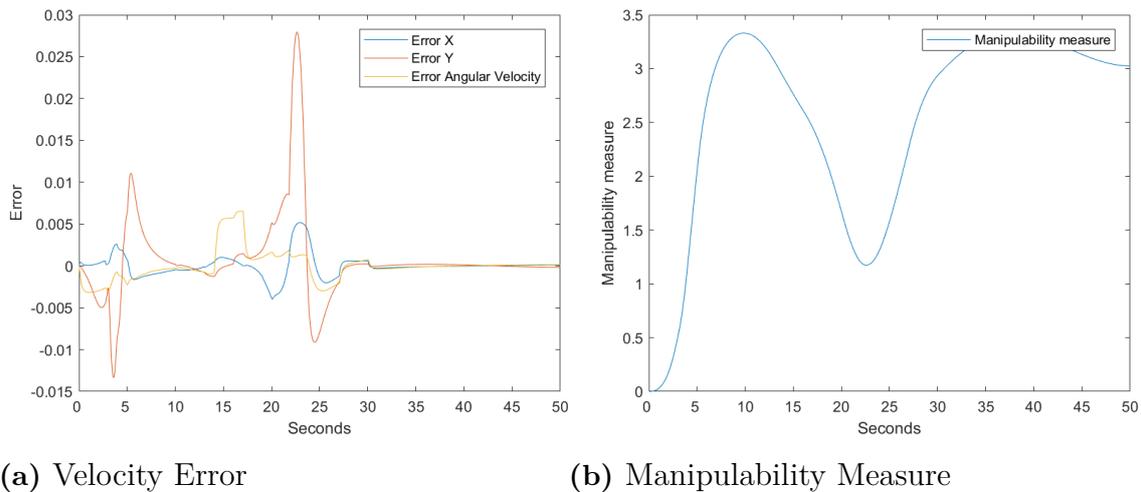
The SVF system using feedback and feedforward can be viewed in figure 4.13 and the error shown in 4.14a is small and might even have been acceptable to an operator. The manipulability measure in figure 4.14b reveals that at 20-25 seconds a singularity was close by, which correlates to the biggest error spike. The response from the system was fast and the error was quickly reduced when singularities were further away.



**Figure 4.13:** A block diagram showing the system using SVF, feedforward, feedback and acceleration control

Using a PI- or I-controller in a system that is heavily constrained can be difficult since they can produce acceleration that is outside the boundaries while it tries to correct itself. The controllers used was therefore carefully tuned, however some control structures were difficult and did not show promising results. The closed loop systems using DLSI only used the integral part, while SVF used the combined integral and proportional part. All systems had trouble with acceleration spikes that required disadvantageous regulations to be within the boundaries, and most of all a larger  $\lambda$  respectively  $\sigma_0$ . If the acceleration spikes could be kept under control potentially  $\lambda$  respectively  $\sigma_0$  could be reduced resulting in systems with less inherent errors. Therefore acceleration control were tested for both DLSI and SVF.

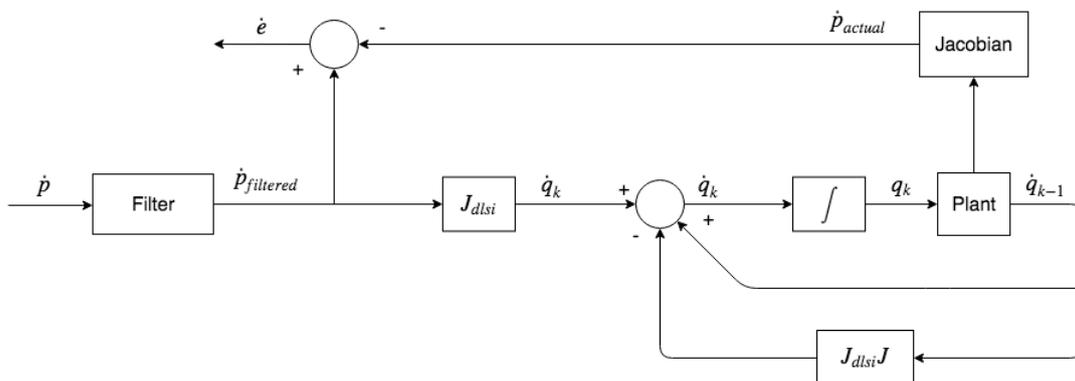
#### 4. Comparison of singularity-robust methods



**Figure 4.14:** Simulation of the feedback, feedforward SVF system

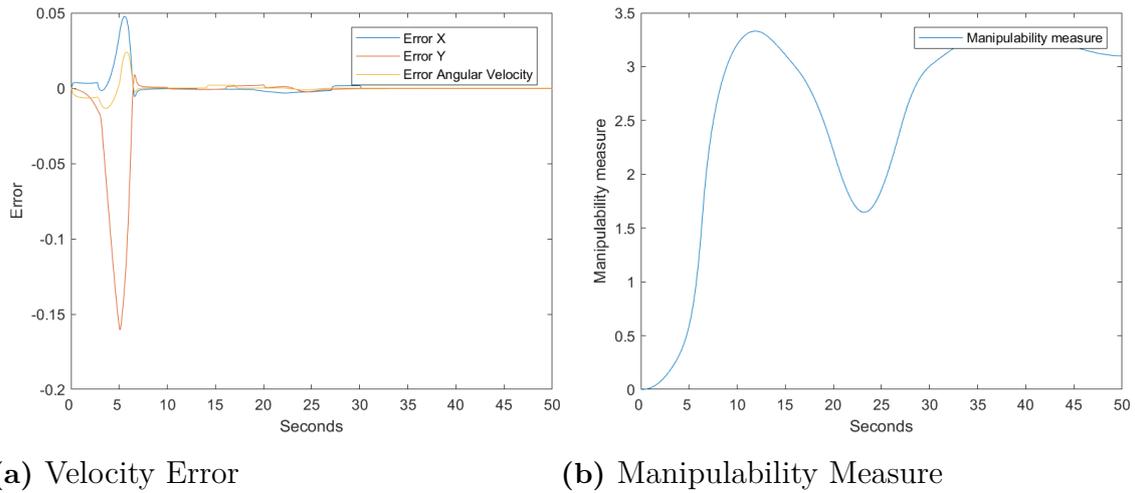
### 4.4 Acceleration Control

To reduce the acceleration spikes when using DLSI and potentially be able to in turn decrease the damping factor,  $\lambda$ , and the threshold,  $w_t$ , acceleration control was used. The control structure used for the open loop DLSI is displayed in figure 4.15, where  $J_{dlsi}$  is the Damped least-squares inverse and where  $J$  denotes the Jacobian.



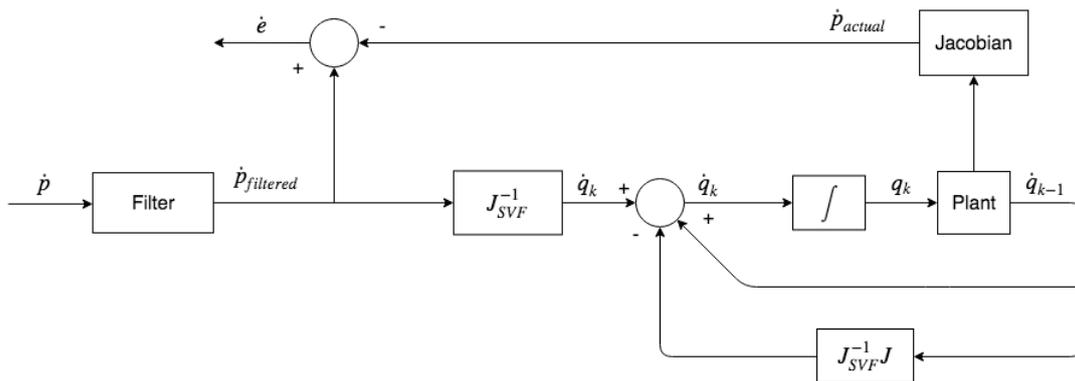
**Figure 4.15:** A schematic block view of the open loop system using DLSI and acceleration control

In figure 4.16a the error is shown for the DLSI open loop system using acceleration control. The maximum error was about 0.15, and after 5 seconds the error went toward 0 and compared to the corresponding system using velocity control this was a huge improvement. In figure 4.16b the manipulability measure can be viewed. The system responded fast to changes since the errors quickly went towards 0 when the manipulability measure increased.



**Figure 4.16:** Simulation of the open-loop system with DLSI and acceleration control

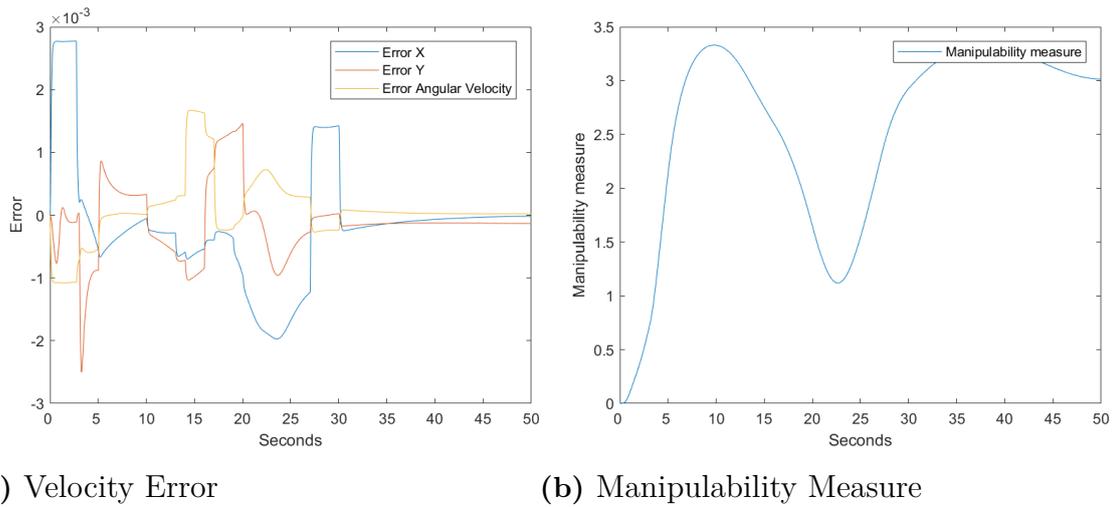
For SVF using acceleration control the control structure can be seen in figure 4.17, where  $J_{SVF}^{-1}$  denotes the Jacobian pseudo inverse based of SVF. The structure enabled the reduction of  $\sigma_0$  from 1.1 to 0.3, which resulted in a great reduction of the error which can be seen in figure 4.18a. The manipulability measure in figure 4.18b shows that the arm started in a singularity and that a singular region was close-by at 20-25 seconds. The errors were in the interval of -0.0025 and 0.003. By using a system based of the feedback and feedforward that previously have shown great improvements for SVF-systems, this error might be even further reduced.



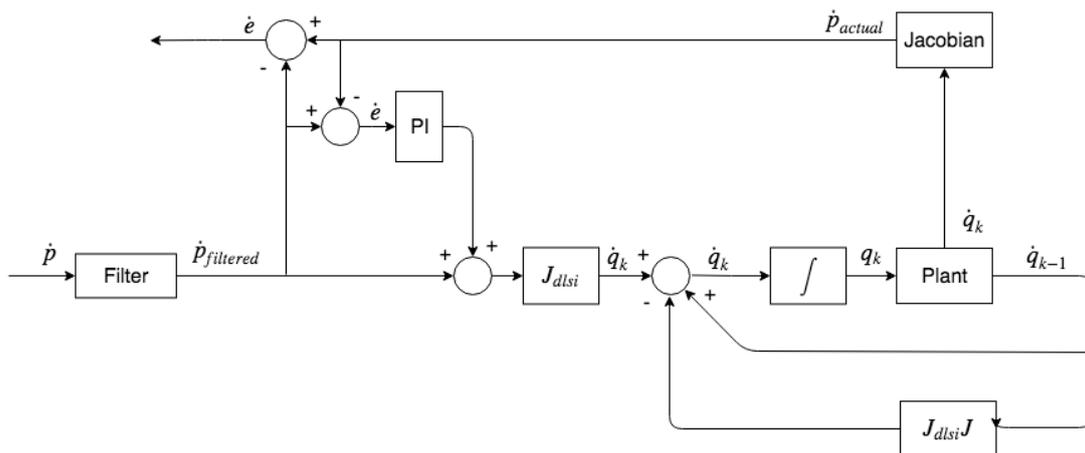
**Figure 4.17:** A schematic block diagram of the open loop system using SVF and acceleration control

However, no improvements emerged when the feedback, feedforward system with DLSI was used. The system schematic can be viewed in figure 4.19.

#### 4. Comparison of singularity-robust methods



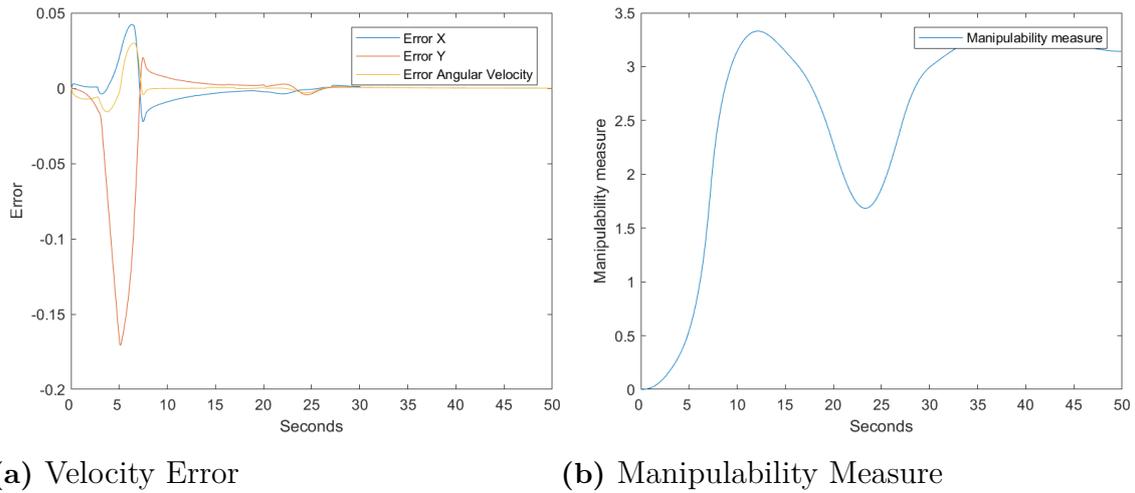
**Figure 4.18:** Simulation of the open loop SVF system with acceleration control



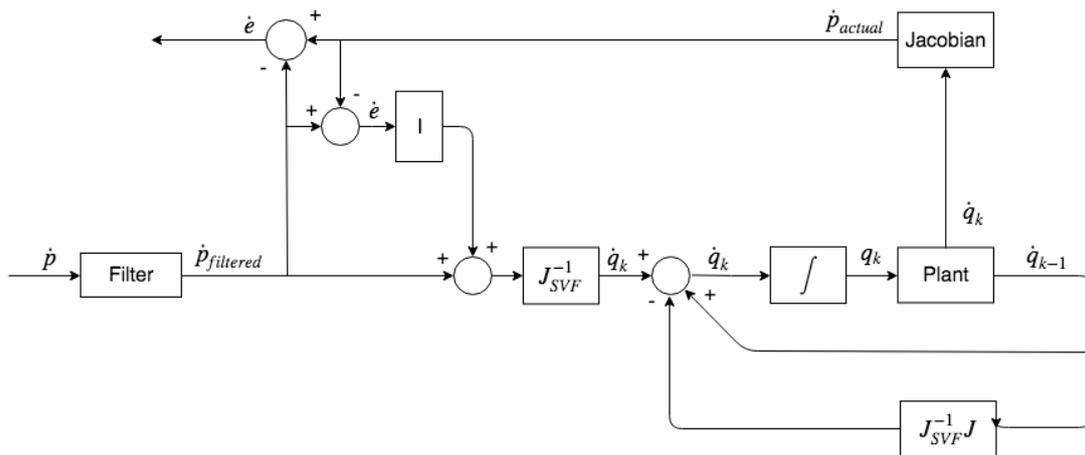
**Figure 4.19:** A schematic view of the system using feedback, feedforward, acceleration control and DLSI

The error of figure 4.20a does not show a performance even remotely close to that of the open loop SVF with acceleration control. The error had a global maximum even larger than in the open loop DLSI system with acceleration control. The manipulability measure can be viewed in figure 4.20b. This system produced acceleration spikes that required both a high  $\lambda$  and disadvantageous regulations to be within its bounds.

In figure 4.21 the control structure used for a feedback, feedforward, SVF system with acceleration control can be seen.



**Figure 4.20:** Simulation of the feedback, feedforward DLSI system with acceleration control

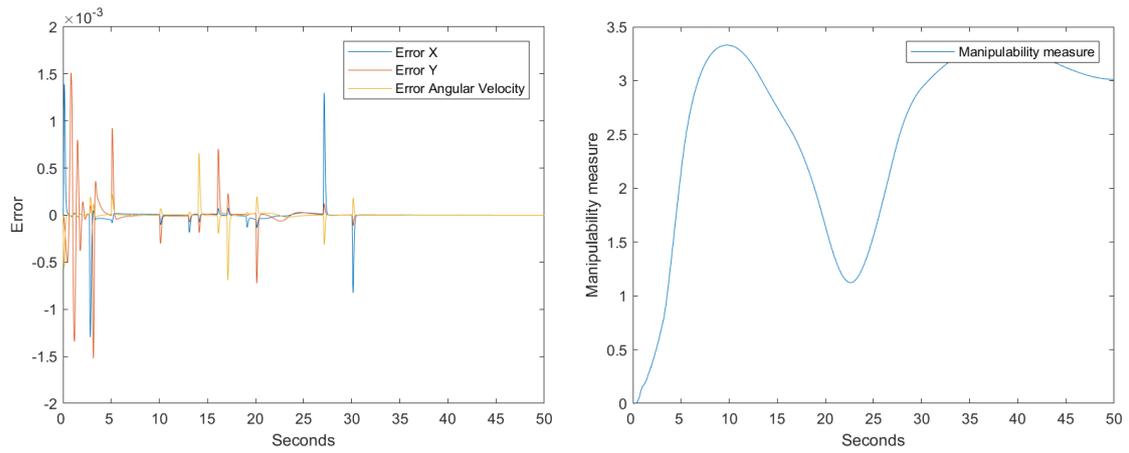


**Figure 4.21:** A schematic block diagram of the feedback, feedforward SVF system and acceleration control

The error can be seen in figure 4.22a with its 10 factor improvement compared to its counter part where velocity control was used. The global maximum of the error went up to 0.0015, and the global minimum to -0.0015 and the error looked to be small spikes followed by quick reductions down to 0. From the manipulability measure in figure 4.22b can be seen that the most spikes around 0 to 5 seconds correlated to a proximity to a singularity, and likewise at 15-30 seconds this resulted in spikes. The biggest error spikes were 0.6 % of the maximum velocity command and they might have been small enough not to be noticed by an operator, thus making for responsive and exact movements of Liongrips arm.

#### 4. Comparison of singularity-robust methods

---



(a) Velocity Error

(b) Manipulability Measure

**Figure 4.22:** Simulation of the feedback, feedforward SVF system with acceleration control

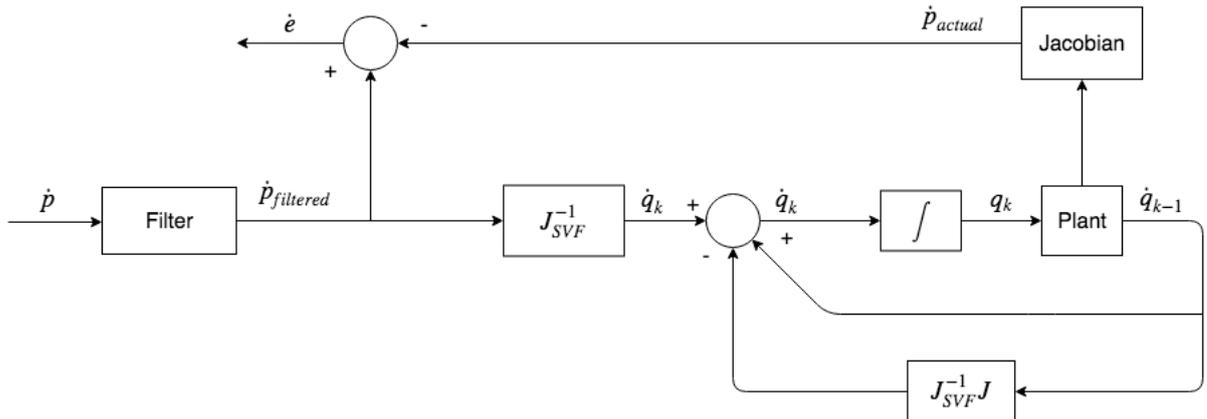
# 5

## Results and discussion

The results from the simulations are presented in section 5.1. Section 5.2 suggests a hardware setting that can be used with the proposed algorithm on Liongrip. A discussion on the result follows in section 5.3.

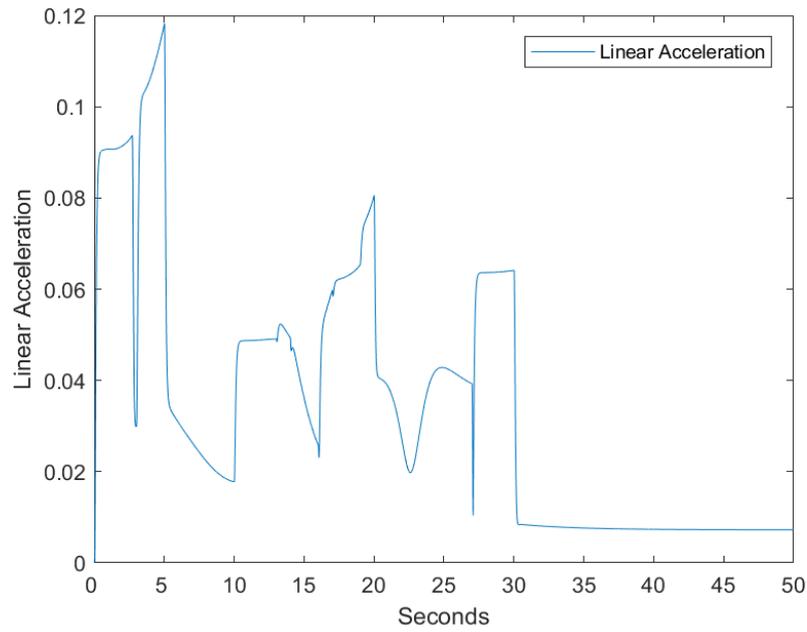
### 5.1 Simulation Results

The two different control structures with smallest error was chosen to be presented in depth. It was the open loop SVF and the feedback, feedforward SVF with acceleration control for both, since that allowed for a smaller value of  $\sigma_0$  and hence a smaller introduced error. The first mentioned control structure can be viewed in figure 5.1 and the second in figure 5.7.



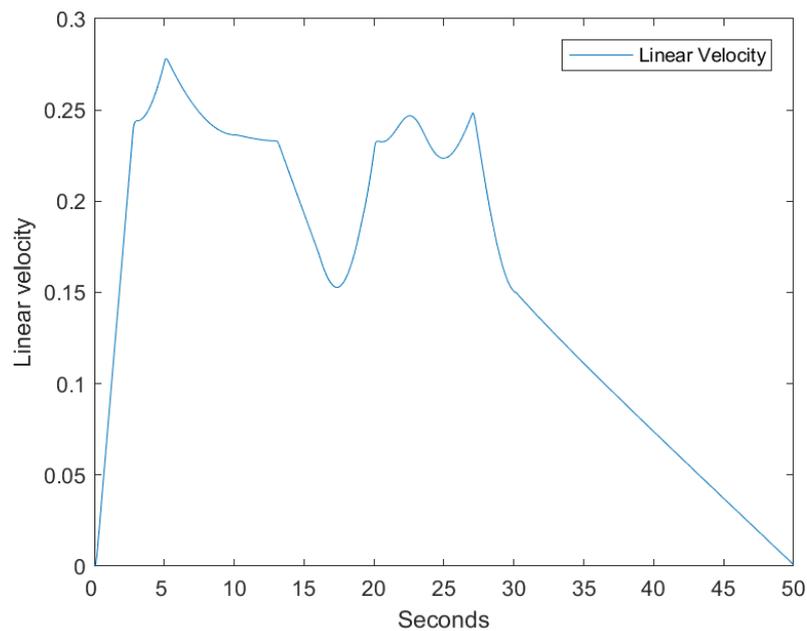
**Figure 5.1:** A block diagram view of the open loop system using SVF and acceleration control

The linear acceleration can be viewed in figure 5.2 and it was well within its bounds of  $0.283 \text{ m/s}^2$ . A global maximum of 0.12 can be seen at the 5th second.

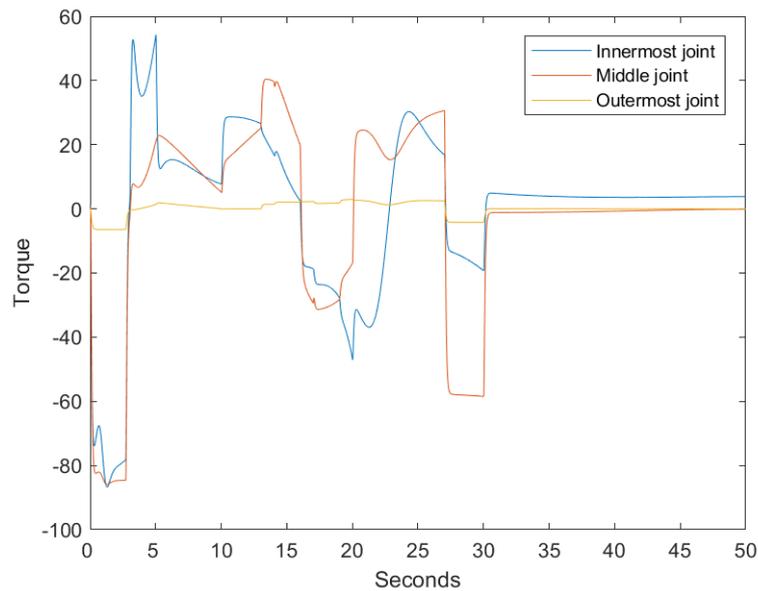


**Figure 5.2:** The linear acceleration for the end effector when using open loop SVF and acceleration control

In figure 5.3 the linear velocity of the end effector can be viewed and it was under its limit of  $0.354 \text{ m/s}^2$ .

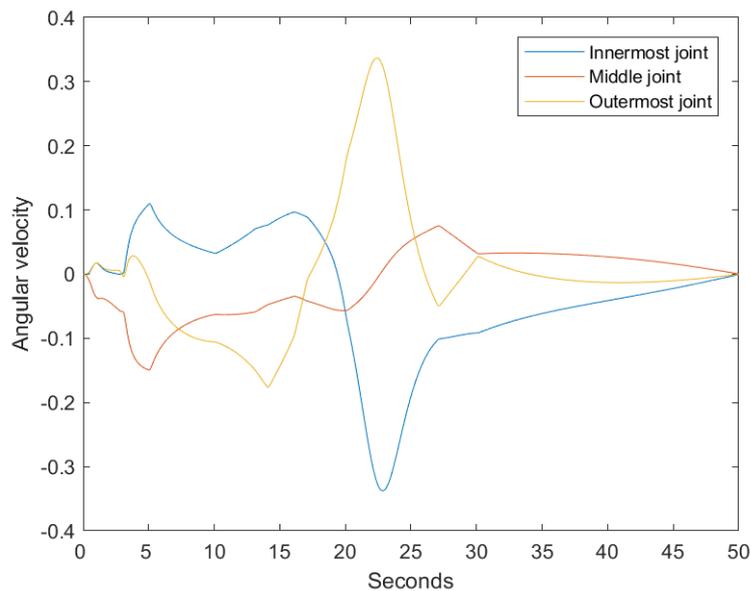


**Figure 5.3:** The linear velocity of the end effector when using open loop SVF and acceleration control

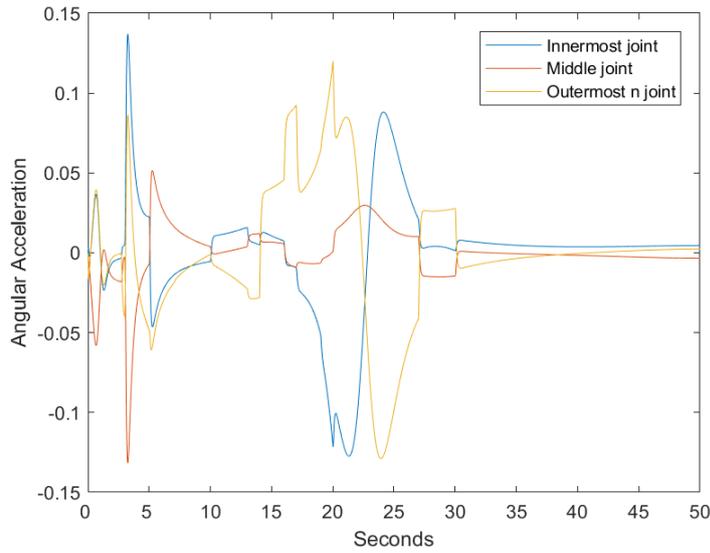


**Figure 5.4:** The torque required in each of the three actuated joints to perform the movement when using open loop SVF and acceleration control

The torque required by the actuated joints to move the arm can be inspected in figure 5.4. Comparisons with the torque listed in table 3.1 shows that the early approximation is similar, but this simulation generally shows a lower torque, which is reasonable given the relationship of torque and angular acceleration, equation (2.30), since the acceleration control calculates minimum acceleration. The torque is revisited further down in section 5.2.



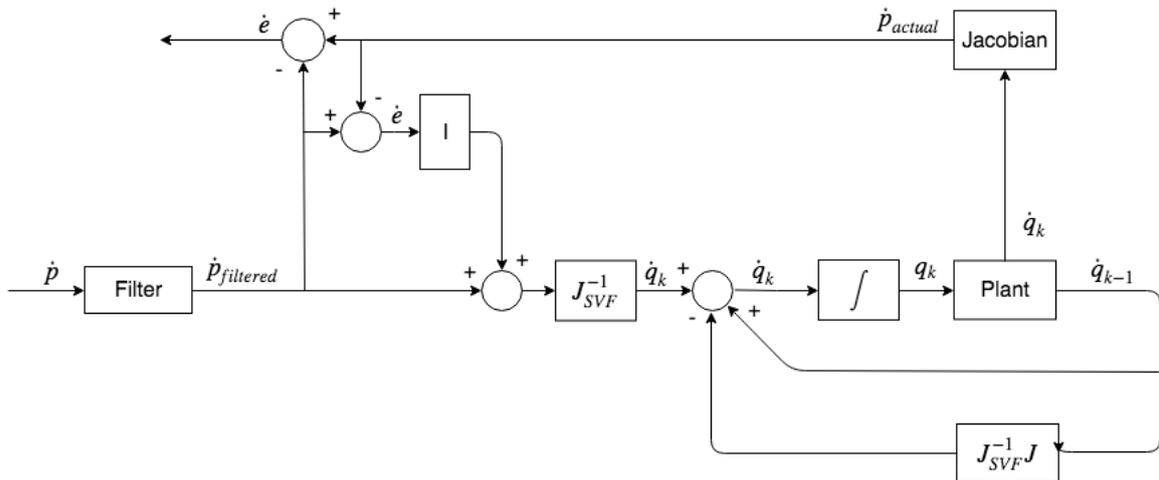
**Figure 5.5:** A plot showing the angular velocity in the three joints when using open loop SVF and acceleration control



**Figure 5.6:** A plot showing the angular acceleration in the three joints when using open loop SVF and acceleration control

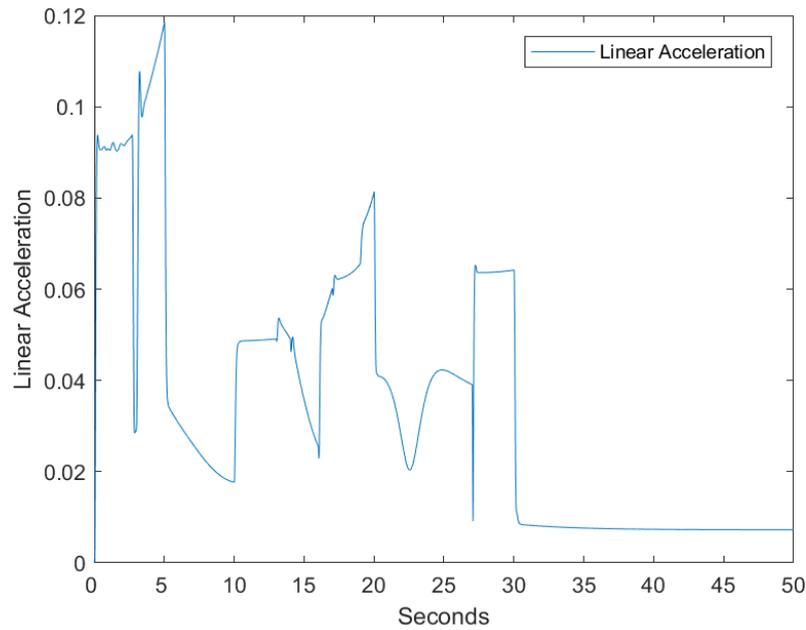
Figure 5.5 and 5.6 shows the corresponding angular velocity and acceleration of the three joints. The velocity was within the interval of  $-0.35$  to  $0.35$   $m/s$ . The acceleration was within  $-0.15$  and  $0.15$ . The angular velocity is displayed in  $rad/s$ , and the angular acceleration is displayed in  $rad/s^2$ .

The control structure used for the feedback, feedforward can be seen in figure 5.7.



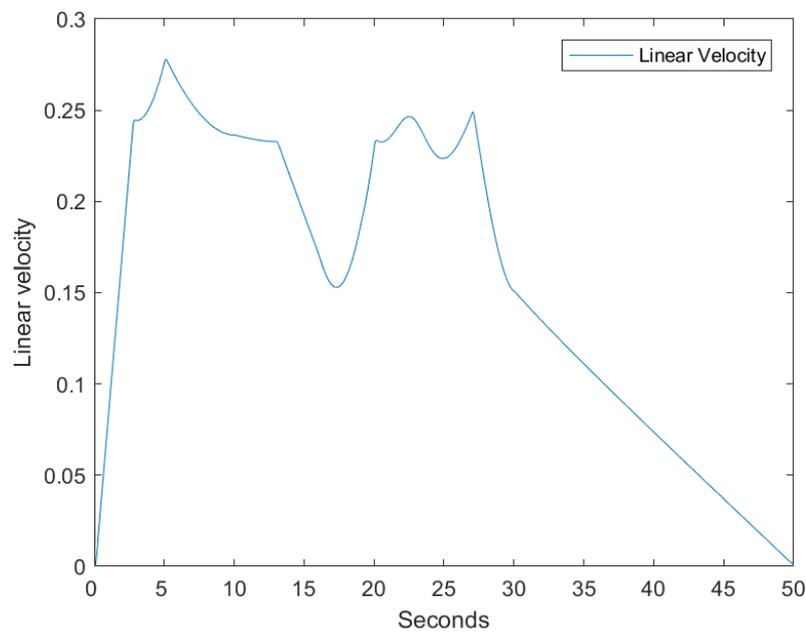
**Figure 5.7:** A block diagram view of the feedback, feedforward system using SVF and acceleration control

The linear acceleration, shown in figure 5.8, was well within the limits of  $0.283$   $m/s^2$  with a global maximum of close to  $0.12$ , which follows closely to the linear acceleration profile of the open loop system, as can be seen in figure 5.2.



**Figure 5.8:** The Linear acceleration of the end effector in the feedback, feedforward SVF-system

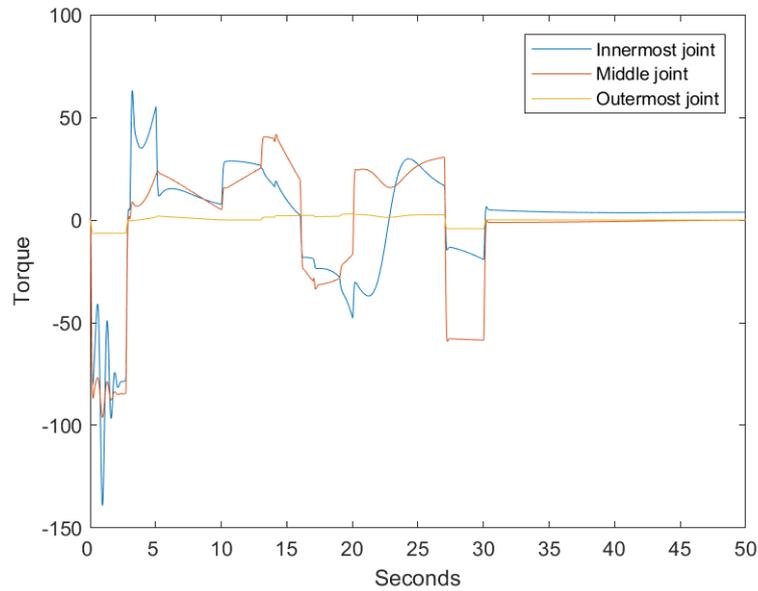
The linear velocity, shown in figure 5.9, is within the bounds of  $0.354 \text{ m/s}$  with a global maximum around  $0.28 \text{ m/s}$  and several local maximum around  $0.25 \text{ m/s}$ .



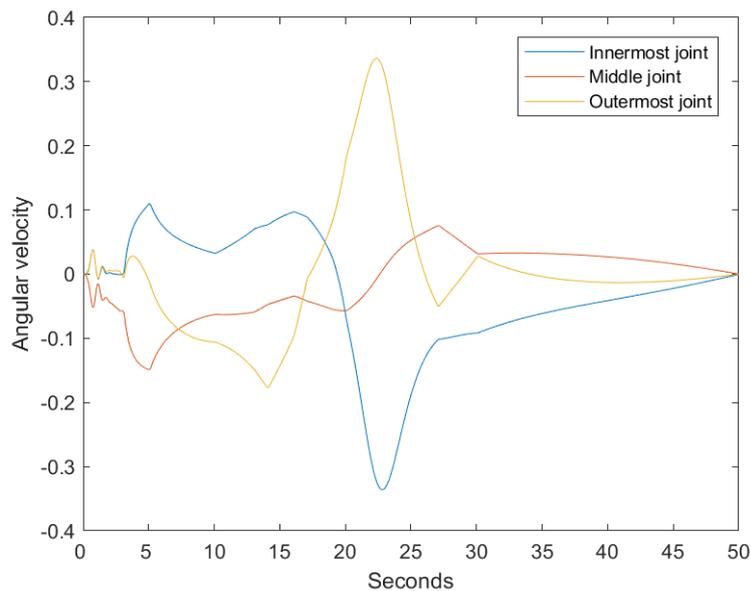
**Figure 5.9:** The Linear acceleration of the end effector with the feedback, feedforward SVF-system

In figure 5.10 the required torque for each joint is displayed. This is revisited further

down in section 5.2. As can be expected the torque is highest for the innermost joint, since it is moving the entire arm and the load, with the middle joint following close after. The outermost joint only requires a portion of their torque. Comparisons with the torque listed in table 3.1 show that the early approximation is quite evenly matched with this simulation.

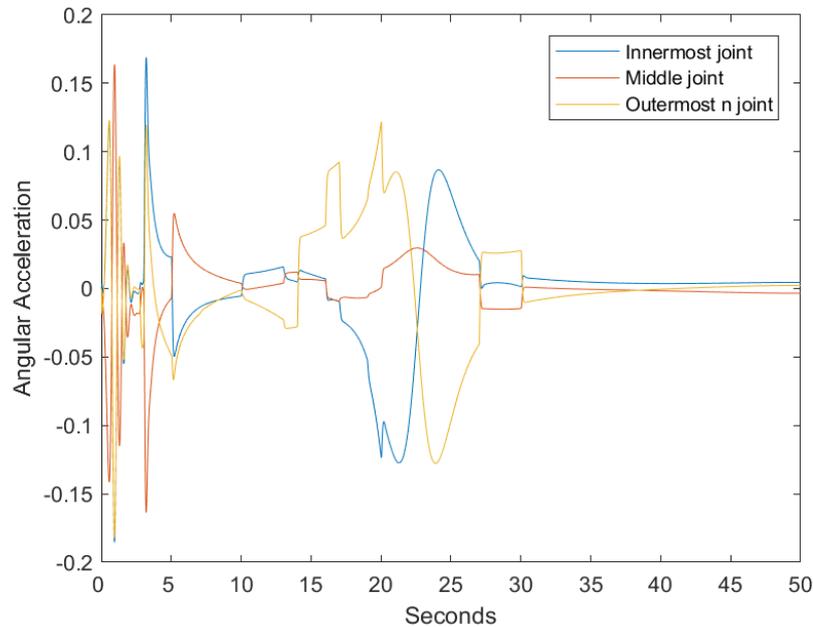


**Figure 5.10:** A plot showing the torque needed in the three actuated joints for the system using SVF, feedback and feedforward

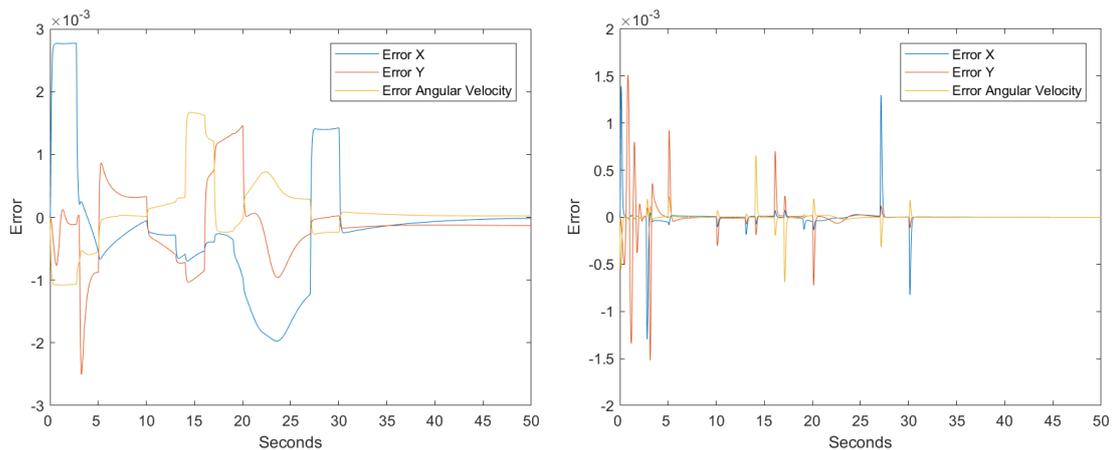


**Figure 5.11:** A plot showing the angular velocity in the three joints using SVF, feedback and feedforward

Figure 5.11 and 5.12 shows the corresponding angular velocity and acceleration of the three joints, where the velocity was within the interval of  $-0.35$  to  $0.35$   $m/s$  and the acceleration was within  $-0.19$  and  $0.17$   $m/s^2$ . They are both similar to the open loop SVF system portrayed above. The angular velocity is displayed in  $rad/s$ , and the angular acceleration is displayed in  $rad/s^2$ .



**Figure 5.12:** A plot showing the angular acceleration in the three joints using SVF, feedback and feedforward



(a) Open loop

(b) Feedback, feedforward

**Figure 5.13:** The errors of the SVF systems using acceleration control

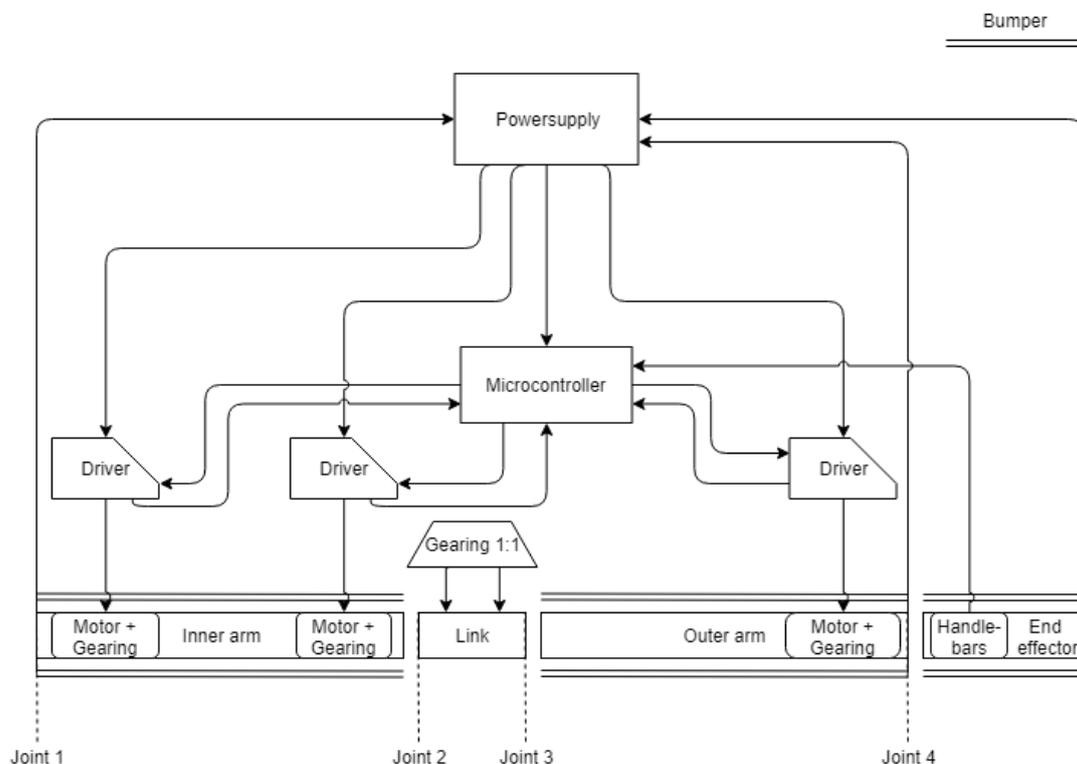
To finish off the errors are shown in figure 5.13a and 5.13b. The scale of the two figures y-axis is not the entirely the same. The former plot shows the open loop

SVF system using acceleration control. So far no clear advantage from one control structure to the other have been shown in the plots regarding velocity, acceleration or torque, however the errors are smaller for the feedback, feedforward system, with only spikes, and the open loop systems errors has larger magnitude and does not go toward 0.

The error in both these systems looks to be quite small only being about 0.6-1.5 % of the maximum input amplitude. Whether these systems were responsive and the error was small enough for an operator to feel completely in control was left for testing with a physical prototype.

## 5.2 Hardware

In figure 5.14 a schematic overview can be viewed that presents a possible configuration for Liongrips arm. In the bottom of the figures the four main parts of the arm are displayed with markings of the four relevant joints. The connection between joint 2 and 3 can be done with some sort of gearing or similar. The breaks and temperature sensors are included in the motors and are not depicted in the figure. The connection from the bumper to the power supply is, as can be viewed, independent of the microcontroller to ensure that the delay from switch to power off is kept at the bare minimum.



**Figure 5.14:** Overview of the hardware for a proposed control system (updated in drive)

In table 5.1 the hardware that were used to evaluate the system is presented. A CMP50S with the shaft angled at 90 degrees, which is one of SEW-EURODRIVES smallest servomotors, with a gearing of 512. The angle was in order to lay the motor flat on the arm instead of having it stand on end in order to actuate the joint. An AK0H, absolute encoder, was added which can keep track of 4096 rotations, which divided by 512 translates to 8 full rotations before it lost track of where it was. This was almost 8 times more than what was needed for Liongrip. A break was built into each motor with a breaking torque of 2.4 Nm which comes up to 1228.8 Nm when the gearing was taken into account. The rated torque for the CMP50S was 1.3 Nm which gave a total possible output of each motor of 665,6 Nm when the gearing was taken into account which was well above the maximum torque simulated in figure 5.10, this could imply that the combination was unnecessary strong. Due to the internal temperature sensor in the servomotors the system can work at peak torque while requiring the operators to perform the movements without the motors breaking. The microcontroller can be programmed to send warnings before the temperature rise to dangerous levels and shut off the motors when the dangerous levels are reached.

	Name	Additional
Actuator	CMP50S	Absolute encoder, Breaks, Gearing: 512
Driver	MDX61B0005-5A3-4-00	-
Microcontroller	-	CAN-support
Bumper	-	-

**Table 5.1:** Possible hardware list

The CMP50S motor had a resolution of 4096 steps per revolution and the gearing of 512 increased the resolution of the joint to a resolution of 2,097,152 giving the smallest possible step of  $\frac{1}{2097152}$  which is smaller than  $\frac{1}{27004}$ , and thus the smallest step of 1 mm was ensured.

The drivers proposed by SEW communicates through the protocol Controller Area Network (CAN), so consequently the microcontroller requires support for CAN.

Figure 5.4 and 5.10 shows that the torque requirement of the system was lower than what the proposed actuators could handle. That means that with these actuators a system that could handle the load without the help of an operator was created, and therefore it was possible to use a lower gearing to again have a system that required the operators help. However the combination of motor and gearing used the inertia ratio as the sizing factor and further research is required to determine how important the inertia ratio actually is for a system where the operator provides his/her force to aid in the movement.

The proposed RPM of the motors were 4500, with the gearing of 512, which correlated to a angular velocity of 0.92 *rad/s*. According to figure 5.11, that was higher than the requirements by a factor of about 2.8 so to increase safety the actuator speed could be decreased, which would set a lower hardware limit to the maximum possible velocities.

## 5.3 Discussion

In the discussion four main subjects are touched upon. First in section 5.3.1 the role of the operator is discussed, in section 5.3.2 the safety issue is delved into, in section 5.3.3 the sustainability is discussed and lastly the control design is debated in section 5.3.4

### 5.3.1 The operator

No simulations were done on the possible disturbance and/or help an operator can produce, however the system was developed with that in mind. The proposed actuators were too strong to require help from an operator with the simulated load of 150 kg and the constrained linear velocities and accelerations, however the system had everything needed to be a support system to an operator instead of operating autonomously. The actuators had temperature sensors that would signal if they were working above the rated torque for too long and as such they could, with the aid of an operator, do work with a load that was actually too heavy.

Since no trajectory was known beforehand the system would always strive to minimise the error in the present time step. An operator could thus have controlled the arm to maximum velocity and used its own force to try and drag the arm to velocities that surpassed the limit, this however would be resisted by the internal feedback loop of the servomotors and any feedback of the control structure.

### 5.3.2 Safety

When heavy machinery is to be used specifically in close proximity with humans safety is vital. The obvious reason is to minimise the risk for physical harm, however there are also secondary reasons such as minimising the worry in operators and the engineers that built the machine.

The actuators and gearing proposed by SEW-EURODRIVE had a torque that was too high to require the help of the operator. A higher torque this close to an operator correlates to a higher risk if something were to happen. There is of course a boundary where higher torque would not equal a higher risk however that would need further research.

The inherent closed-loop system of servomotors would aid in ensuring that the maximum linear velocity of the end effector was not surpassed, which would increase safety. However the linear acceleration of the end effector was only constrained with the use of preferable input and acceleration minimisation, and thus a filter handling sub-optimal input and a way of actively constraining acceleration should be implemented. The limits in the thesis should be tested with a prototype to verify that they were chosen to a reasonable magnitude.

Adding bumpers at all the sides of the arm would increase safety with regards to the risk of hitting or clamping due to movement, however it does nothing to reduce the

risk of hitting or clamping with the load due to movement. Further research into a smart system sensing forces applied in the horizontal plane might remedy this.

### 5.3.3 Sustainability

If the lifting aid's control structure built is robust and safe the social sustainability gain is possibly huge. An operator using an actuated Liongrip many times throughout the day will feel a tremendous relief since much or all of the physical strain is removed. Depending on the actuators rated torque and the weight of the load the operator will get more or less help from the control structure where a higher rated torque will give a greater relief up to when the actuators control the load by themselves. A lower physical strain for the operator can decrease the number of days for sick-leave and it will have the possibility to create a greater equality since there can no longer be a reason for choosing a person that is stronger because the work has heavy lifting. A system that is fully controlled by the actuators could be viewed as a better system, since no physical strain would be put on the operator.

One reason for buying a lifting aid could be to increase the efficiency in the work place which might result in a reduction of the workforce. Is it ethical to replace humans with machines? During the human history we have always worked towards improvements in efficiency, to automate what needs to be done, to get a better yield. From the irrigation systems of the fertile crescent to autonomous cars of today there has always been the curiosity in mankind to wonder what can be done and what can be improved. With that said it's important to use any extra time gained from efficiency on other tasks.

The environmental impact as a consequence of this project is predicted to be low. An evolution of the lifting aid would use more material which will contribute to resource depletion unless a closed loop system is implemented regarding the recycling and production of the lifting aid. The amount of energy used while operating will increase and in that regard the environmental impact depends on where the energy comes from. The environmental sustainability is not foreseen to benefit from this project unless the lifting aid is chosen instead of equipment that has a larger environmental impact.

### 5.3.4 Control design

A control structure using optimisation in that incorporates the limitations to velocity and acceleration could possibly reduce the error and ensure that the system is always within its bounds, for example Model Predictive Control might be an idea. This of course would demand more from the processing unit and might have to be run on a computer.

A constraint that lowers the velocity when the arm is almost folded could improve safety even further and minimise the risk for equipment damage if a mistake is done with the command signals.

A basic system has several benefits compared to a more advanced system, for example there are less parts that can go wrong and it can be run on a microcontroller, which lowers the production cost and complexity of a commercial product. With a basic system there is lower risk of the response time being too long due to calculations.

Verification could have been done with different sets of input signals and when moving different sized loads. For example, many customers use Liongrip to move long shafts where they could be lifted in centre of mass as well as off-centre, and therefore the result from such a simulation would be interesting.

# 6

## Conclusion and future work

### 6.1 Conclusion

The results from the simulations showed that it is possible to build an actuated version of Liongrips arm. However the constraints and that the control signal was processed live made the design non-trivial. Using bumpers that can cut the power when pressed; sizing the actuators to limit the possible max velocities and accelerations; and using a control structure that was robust the risks could be greatly reduced, however physical testing in a controlled environment is needed before a product hits the market. It is important that the risks of heavy machinery are never overlooked and that all development is with this in mind.

Using Singular Value Filtering with acceleration control the impossible high velocities and accelerations close to singularities were removed and the error was kept at a minimum. The system using feedback and feedforward will be concluded as the best performing candidate for Liongrips control system for movement in the horizontal plane. The feedback loop together with the servomotors built in feedback the system will aid in the handling of unwillingly drift due to heavy loads as long as the torque created by the drift is less than the torque provided by the servomotors and gearing.

The proposed actuators and gearing system from SEW-EURODRIVE were far stronger than required and that should be reduced in a physical product, provided that the system functions properly with a less than ideal inertia ratio. The speed of the actuators together with the gearing was approximately a factor 2.8 larger than required, and the speed of the actuators should therefore be reduced to lower the hardware's maximum possible velocity and increase safety. The adjustments to the proposal should be made in such a way that the hardware is just strong and fast enough to further increase safety. The absolute encoder, the breaks, the temperature sensors, and the drivers all fulfil the requirements of the system.

At the beginning of this report three questions were asked:

- What safeguards are required to ensure operator safety?
- How will the system be designed to incorporate the above mentioned safeguards?
- What control structure makes for precise and comfortable handling by an operator?

The safeguards were that: a) neither the linear velocity or acceleration limits were surpassed, b) the actuators had emergency breaks, c) bumpers, or similar, were placed at the side of the arm that engages the breaks when pressed, and d) that no impossibly high joint velocities or accelerations occurred in or around singularities.

Following the mathematical relations presented in equation (2.5) the linear velocity and acceleration would not be surpassed when using an open-loop system. When a closed loop system was used this was ensured by limiting the output signal of the controller to ensure that no overcompensation happened. When incorporating emergency breaks into the system it is important not to give the control system complete control over the breaks. This means that there should be another basic system concerning the bumpers and the emergency breaks that cannot be overridden by the control structure concerning the movement of the arm. Singular Value Filtering will make sure that no impossibly high joint velocities or accelerations will happen by the properties of the method.

The last question is open for discussion since there has been no possibility of physical tests, however the feedback, feedforward SVF system using acceleration control showed the most promise with reasonable simulations and smallest error and might therefore be the answer to this question.

## 6.2 Future Work

A physical prototype is required to research what effect the system's inertia has on the actuators internal inertia when the actuators are there to aid the operator with the movement. It could be that a very high inertia ratio and high torque would give a flimsy feel while a high deficit in required torque but with a lower inertia ratio would feel sturdier.

For the movement control the most intuitive placement of the origin requires physical prototyping and real life trials. It is one thing to guess where it should be placed, but another to feel it in action. A poorly placed origin can result in poor controllability and in worst case scenarios result in hazardous movements.

The safety of the system should be further researched and tested on a physical prototype.

Simulations could be done to investigate both the disturbance an operator can put on the system and how well the control structure can handle it and how big a disturbance the system can handle when it is due to unwillingly drift. If a prototype is built the unintentional drift is likely to change. Simulations concerning sub-optimal trajectories, where an operator tries to move the arm further out than it can go for example, should be done to investigate if those kind of trajectories requires an adjustment to the control structure.

Further research should be done to identify specifically which microcontroller and bumper should be used.

# Bibliography

- [1] Abdulghany, A. Generalization of parallel axis theorem for rotational inertia. *American Journal of Physics*, 85, 2017.
- [2] Adams - the multibody dynamics simulation solution (student edition 2017.2), 2017.
- [3] Arc length definition. <https://www.mathopenref.com/arclength.html>. Accessed: 2018-05-15.
- [4] Torras. C Colomé, A. Redundant inverse kinematics: Experimental comparative review and two enhancements. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.
- [5] Density Of Steel. <https://hypertextbook.com/facts/2004/KarenSutherland.shtml>. Accessed: 2018-03-12.
- [6] Walker, I. Deo, A. Overview of damped least-squares methods for inverse kinematics of robot manipulators. *Journal of Intelligent and Robotic Systems*, 14:43–68, 1995.
- [7] DiStefano, J., et. al. *Feedback and control systems*. McGraw-Hill, 2 edition, 1990.
- [8] Hydraulic or Pneumatic Manipulator Arms and Vertical Lifters. <https://www.ergonomicpartners.com/manipulator-arms.aspx>. Accessed: 2018-02-13.
- [9] De Luca, A. Flaco, F. Discrete-time velocity control of redundant robots with acceleration/torque optimization properties. *IEEE International Conference on Robotics & Automation (ICRA)*, pages 5139–5144, 2014.
- [10] Hibbeler, R. *Engineering Mechanics: Dynamics*. Pearson Prentice Hall, 12 edition, 2010.
- [11] Lifting Device liongrip. <http://www.hofpartner.com/en/lifting-aids/lifting-aid-liongrip/>. Accessed: 2018-01-15.
- [12] Klämlist. <https://www.oemautomatic.se/produkter/sensor-,-a-,-maskins%C3%A4kerhet/kl%C3%A4mlister>. Accessed: 2018-05-14.
- [13] Stepper Motor vs Servo Motor which should it be? <https://www.kollmorgen.com/en-us/service-and-support/knowledge-center/>

- white-papers/stepper-motor-or-servo-motor-which-should-it-be/. Accessed: 2018-01-18.
- [14] Kwakernaak, H., Sivan, R. *Linear Optimal Control Systems*. Wiley & Sons, Inc, 1 edition, 1972.
- [15] Lennartson, B. *Reglerteknikens grunder*. Studentlitteratur, 4 edition, 2000.
- [16] Micropython. <https://micropython.org/>. Accessed: 2018-05-14.
- [17] Chapter 2 Precision Engineering Principles. [https://ocw.mit.edu/courses/mechanical-engineering/2-76-multi-scale-system-design-fall-2004/readings/reading\\_13.pdf](https://ocw.mit.edu/courses/mechanical-engineering/2-76-multi-scale-system-design-fall-2004/readings/reading_13.pdf). Accessed: 2018-01-18.
- [18] Faq: What is servo motor inertia and why does it matter? <https://www.motioncontroltips.com/faq-what-is-servo-motor-inertia-and-why-does-it-matter/>. Accessed: 2018-01-18.
- [19] Servo Motor Features Overview. <http://www.orientalmotor.com/servo-motors/technology/servo-motor-features.html>. Accessed: 2018-03-7.
- [20] et. al Orly, A. Singular value decomposition for genome-wide expression data processing and modeling. *Proceedings of the National Academy of Sciences of the United States of America*, pages 10101–10106, 2000.
- [21] Partner Equo. <http://www.dalmec.com/partner-equo/>. Accessed: 2018-02-13.
- [22] Popov, V. *Contact Mechanics and Friction*. Springer, 1 edition, 2010.
- [23] Lunzmann, S. Reedy, J. Model based design accelerates the development of mechanical locomotive controls. Technical report, SAE international, 2010.
- [24] Serway, R. *Physics for Scientists and Engineers*. Saunders College Publishing, 2 edition, 1986.
- [25] Welcome to SEW-EURODRIVE! <https://www.sew-eurodrive.se/hemsida.html>. Accessed: 2018-03-8.
- [26] Siciliano, B., et. al. *Robotics, Modelling, Control and Planning*. Springer, 2nd edition, 2010.
- [27] Model and simulate multidomain physical systems. <https://se.mathworks.com/products/simscape.html>. Accessed: 2018-03-8.
- [28] Simulation and Model-Based Design. <https://se.mathworks.com/products/simulink.html>. Accessed: 2018-03-8.
- [29] Murray, R. Åström, K. *Feedback Systems - An introduction for Scientists and Engineers*. Princeton University Press, v2.11b edition, 2012.

# A

## Appendix 1

Material not required for the understanding of the work, that could be interesting to read, has been collected here. Section A.1 shows the derivation of the jacobian as well as the entire Jacobian used in the simulations and section A.2 displays the complete calculations to identify the singularities of Liongrips arm.

### A.1 Jacobian

Since the joints axes are all parallel to the z axis the following holds:

$$z_0 = z_1 = z_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (\text{A.1})$$

The position vectors were calculated with the help of the angle, the length of the arm and ordinary trigonometric functions of sine and cosine. The variables are defined as is listed in the table A.1.

Variable	Value
a1	L1 + L2/2
a2	L3 + L2/2
a3	L4

**Table A.1:** The variables names and values used to create the system

Note that  $s_1$  is short form for  $\sin(\alpha)$ ,  $s_{12}$  is  $\sin(\alpha + \beta)$  and  $s_{123}$  is  $\sin(\alpha + \beta + \gamma)$ .  $c_i$  is representing the same but for cosine, where  $i = [1, 12, 123]$ . Giving the following position when the origin was placed at the innermost joint:

$$P_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (\text{A.2})$$

$$P_1 = \begin{bmatrix} a_1 c_1 \\ a_1 s_1 \\ 0 \end{bmatrix} \quad (\text{A.3})$$

$$P_2 = \begin{bmatrix} a_1 c_1 + a_2 c_{12} \\ a_1 s_1 + a_2 s_{12} \\ 0 \end{bmatrix} \quad (\text{A.4})$$

$$P_3 = \begin{bmatrix} a_1 c_1 + a_2 c_{12} + a_3 c_{123} \\ a_1 s_1 + a_2 s_{12} + a_3 s_{123} \\ 0 \end{bmatrix} \quad (\text{A.5})$$

With equation (2.3) and (2.2) that shows how the partition fits together the Jacobian defined in origin,  $J_o$ , was calculated to:

$$J_o = \begin{bmatrix} -a_1 c_1 - a_2 c_{12} - a_3 c_{123} & -a_1 c_1 - a_2 c_{12} & -a_1 c_1 \\ a_1 s_1 + a_2 s_{12} + a_3 s_{123} & a_1 s_1 + a_2 s_{12} & a_1 s_1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad (\text{A.6})$$

The Jacobian now contains three rows of 0 which refers to the information of the linear velocity along axis z, and the rotation around the axes x and y, and in the configuration of Liongrips arm they were neither necessary nor needed and was thus removed leaving the following Jacobian:

$$J_o = \begin{bmatrix} -a_1 c_1 - a_2 c_{12} - a_3 c_{123} & -a_1 c_1 - a_2 c_{12} & -a_1 c_1 \\ a_1 s_1 + a_2 s_{12} + a_3 s_{123} & a_1 s_1 + a_2 s_{12} & a_1 s_1 \\ 1 & 1 & 1 \end{bmatrix} \quad (\text{A.7})$$

### A.1.1 Jacobian

$$\begin{aligned} J_{11} &= -c_{123} a_2 s_{12} + a_3 s_{123} - s_{123} (a_1 s_1 - a_2 s_{12} + a_3 s_{123}) \\ J_{12} &= c_{123} (a_1 s_1 + a_2 s_{12} + a_3 s_{123}) - s_{123} (a_2 s_{12} + a_3 s_{123}) \\ J_{13} &= -a_3 s_{123} \\ J_{21} &= c_{123} a_2 c_{12} + a_3 c_{123} - s_{123} (a_1 c_1 - a_2 c_{12} + a_3 c_{123}) \\ J_{22} &= s_{123} (a_1 c_1 + a_2 c_{12} + a_3 c_{123}) - s_{123} (a_2 c_{12} + a_3 c_{123}) \\ J_{23} &= -a_3 c_{123} \\ J_{31} &= c_{123} + s_{123} \\ J_{32} &= s_{123} - c_{123} \\ J_{33} &= 1 \end{aligned} \quad (\text{A.8})$$

## A.2 Derivation of the singularity

$$\det(J) = 0 \tag{A.9}$$

$$\begin{aligned} \det(J) &= a_1 a_2 \cos^2(\alpha + \beta + \gamma) \sin(\alpha + \beta) \cos(\alpha) \\ &\quad - a_1 a_2 \cos(\alpha + \beta + \gamma) \cos(\alpha + \beta) \sin(\alpha) \\ &\quad - a_1 a_2 \sin^2(\alpha + \beta + \gamma) \cos(\beta) \sin(\alpha) \\ &\quad + a_1 a_2 \sin^2(\alpha + \beta + \gamma) \sin(\alpha + \beta) \cos(\alpha) = 0 \end{aligned} \tag{A.10}$$

The expression  $a_1 a_2$  can be removed through division.  $\cos^2(\alpha + \beta + \gamma)$  and  $\sin^2(\alpha + \beta + \gamma)$  can be combine into two trigonometric ones leaving the following equation:

$$\sin(\alpha + \beta) \cos(\alpha) - \cos(\alpha + \beta) \sin(\alpha) = 0 \tag{A.11}$$

From the relation of two angles  $\sin(\alpha + \beta)$  and  $\cos(\alpha + \beta)$  can be rewritten which leaves the following equation:

$$\cos(\alpha)(\sin(\alpha)\cos(\beta) + \cos(\alpha)\sin(\beta)) - \sin(\alpha)(\cos(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta)) = 0 \tag{A.12}$$

Further algebraic simplifications gives the following equation:

$$(\cos^2(\alpha) + \sin^2(\alpha))\sin(\beta) = 0 \tag{A.13}$$

Which leaves that  $\sin\beta = 0$  which is true for  $\beta = 0, n\pi$  where  $n = 1, 2, \dots, \infty$