NTNU
Norwegian University of
Science and Technology

Marine Crane Dynamics Lab

Modelling and experimental validation

Thomas Haraldsen Evang

Marine Technology
Submission date: July 2017
Supervisor: Eilif Pedersen, IMT

Norwegian University of Science and Technology
Department of Marine Technology

Preface

This master's thesis is submitted to the Norwegian University of Science and Technology (NTNU) in fulfilment of the requirements for the degree Master of Science in Marine Technology. The work aims to develop control laws for positioning the arms of a marine crane and for soft landing of cargo. The work has been performed under the supervision of Associate Professor Eilif Pedersen.

Special thanks goes to engineer Gunnar Bremset, senior engineer Frode Gran and the rest of the engineers at the department's workshop. Gunnar provided good advice, technical expertise and assisted with the mechanical parts of the lab. He was always available when components broke in the lab and provided quick repairs. Frode provided expertise and assistance with the electronic components in the lab and assisted with the design of motor controller circuits, a cabinet for storing electric components and the general setup of the lab. The lab experiments performed in this thesis would not have been possible without them.

I would also like to thank my mother for providing me with motivation and support during the work.

Thomas H. Evang
Trondheim, Sunday 9th July, 2017

Abstract

In this thesis, simulation models for a knuckleboom crane and its winch system have been developed. The simulation models have been used to design model based control laws for positioning the orientation of the crane joints and for softly landing its cargo. The control law for positioning the crane has been implemented and tested in an experimental crane lab at the department of marine technology at NTNU. As part of this process, the lab has been improved with a new set of sensors, filters and hands-on control systems.

The simulation model of the crane has been based upon the physical crane in the lab and was derived using Lagrangian mechanics and modelling procedures for robotic manipulators. The model was derived by treating the crane as a system of linked rigid bodies and includes the weight of the actuator, but not their dynamics. The load of the crane was modelled as a constant weight acting at the tip of the crane. Simulation studies of the model has shown that the crane behaves as an undamped, double gravity pendulum when not influenced by external forces, or with limitations imposed on the joint angles.

The control law for positioning the crane joints was designed using the crane simulation model and the non-linear backstepping technique. The procedure yielded a globally asymptotic stable control law for the system. When the control law was tested with the crane simulation model it was able to follow a ramp reference signal lifting and lowering the crane arms and perform heave compensation with the upper arm, with a load of 16kg attached to the tip.

Testing the control law in the marine crane lab revealed that the crane was able to follow ramp reference signals and perform heave compensation with the upper arm, without an attached load. However, the lower arm was not able to lift a load of 6kg when using a wire with a diameter of 1.7mm. The actuator motor drew full current, but was not able to move the arm. This was most likely due to increased resistance caused by using a thin wire or by the friction profile of the actuator.

The simulation model of the winch has been used to design a control law for softly landing crane cargo. The control law was designed by defining the control objective as tracking, where the wire velocity should follow a desired speed profile. Simulation results have shown that the control law achieved the objective with good accuracy.

The lab has been improved with new gyroscopes that fits the velocity range of the crane and accelerometers that have been attached to the mass centres of the lower and upper crane arm. The accelerometers have been used to measure the absolute orientation of the crane by calculating their orientation relative to the gravitational axis. A low-pass filter has been designed to remove high frequency components from the gyroscope measurements and a Kalman filter to remove noise from the accelerometer and estimate gyro bias. In addition, the crane's winch has been finished and a simple hands on-control system for controlling the winch speed has been implemented. Meaning that the state of the marine crane lab has been improved and that the process of implementing model based controllers simplified.

Sammendrag

I denne oppgaven er det blitt utviklet simuleringsmodeller for en knuckleboomkran og vinsj. Simuleringsmodellene har blitt benyttet til å konstruere modellbaserte kontrollsystemer for å posisjonere kranen og for forsiktig senking av last. Kontrollsystemet for posisjonering har blitt implementer og testet i en eksperimentell kranlab ved institutt for marin teknikk ved NTNU. Som en del av denne prosessen har laboratoriet blitt forbedret med et nytt sett med sensorer og filtre.

Simuleringsmodellen av kranen er basert på den fysiske kranen i laben og har blitt utviklet ved å benytte Lagrange-mekanikk og modelleringsmetoder for manipulatorer. Kranen har blitt modeller som en et system av sammenhengende, stive legemer og inkluderer vekten av aktuatorene, men ikke dynamikken. Kranens last er antatt å være en konstant vekt som henger i kranspissen.

Styringssystemet for å posisjonere kranarmene har blitt utviklet ved hjelp av backstepping-metoden og er basert på simuleringsmodellen. Dette resulterte i en global asymptotisk kontroll-lov. Tester med simuleringsmodellen viste at det var mulig å få kranen til å følge et rampesignal, for å heve og senke kranarmene, og et sinussignal for å utføre hiv-kompensasjon.

Testing av kontroll-loven i laben viste at kranen var i stand til å følge de samme referansesignalene, dog uten last. Eksperimenter med en 6 kilo tung vekt viste at den nedre armen ikke klarte å overvinne statisk friksjon, selv når den trakk maksimalt med strøm.

Simuleringsmodellen av vinsjen har blitt benyttet til å konstruere et styringssystem for forsiktig senking av last. Problemet har blitt formulert ved å kreve at vaierhastigheten skal følge en forhåndsdefinert hastighetsprofil. Profilen har blitt valgt slik at maksimal hastighet holdes for størstedelen av operasjonen og at hastighetene sakte avtar når lasten nærmer seg bakken.

Simuleringsresultater har vist at kontroll-loven klarte å få vinsjen til å følge den definerte hastighetsprofilen med god nøyaktighet.

Kranlaboratoriet har blitt forbedret med nye gyroskoper som er bedre egnet for kranens hastighetsområde og akselerometre har blitt installert for å måle kranens absolutte posisjon. Posisjonen måles ved å finne akselerometerets vinkel i forhold til tyngdeaksen. Et lavpass- og Kalman-filter har blitt designet for å fjerne høyfrekvent støy fra sensorene og estimere feilen i gyroskopene. I tillegg er vinsjen blitt ferdigstilt og implementert. Det betyr at laben er blitt betydelig forbedret og at prosessen med å implementere modellbaserte kontrollsystemer gjort enklere.

Table of Contents

List of Tables

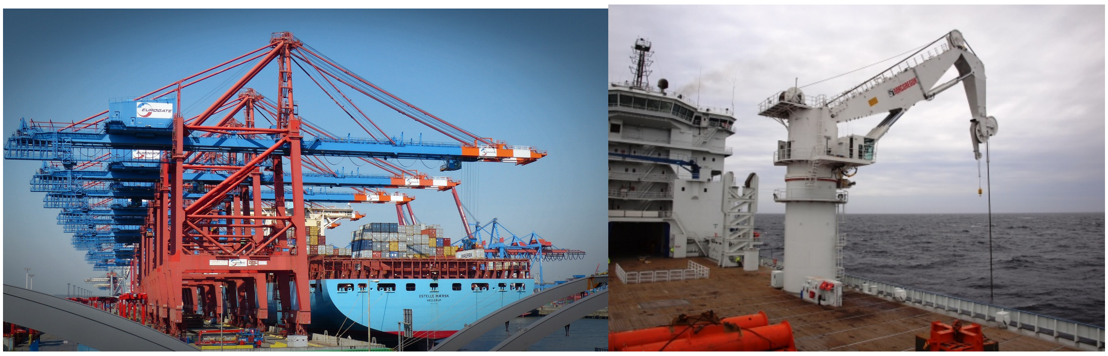
List of Figures

Chapter 1

Introduction

Marine cranes play an important part in several marine operations. They are expected to perform a wide range of different tasks and there are many different designs optimized to handle specific challenges and conditions. For instance, gantry cranes (Figure 1.1a) are often used to fetch and transport cargo in container terminals and between vessels. These cranes may run on rails allowing rapid relocation and are usually capable of lifting heavy load. During these operations speed are essential in order to minimize loading time, thereby decreasing the docking time of vessels and increasing the port's efficiency. Since the cranes are able to draw power from the electric grid and are subject to few environmental disturbances, aspects such as power consumption and environmental disturbance rejection may be less important.

Offshore cranes, on the other hand, face different challenges. Placed on-board vessels, they are affected by vessel motions induced by environmental forces such as wind, waves and current. The amplitude of the motions depend upon weather conditions and may change drastically during an operation. In addition, if the load of the crane is to be lowered beneath the ocean surface, it will be affected by hydrodynamical forces such damping, buoyancy and waves, which induces motions relative to the tip of the crane. Also, the power drawn by the crane is limited by the power plant of the vessel and fuel reserves. Hence, offshore marine cranes have stricter requirements regarding energy consumption and environmental disturbance rejection, while limitations to time and speed may be slacker.



(a) Gantry Crane

(b) Knuckleboom Crane

Figure 1.1: Two types of marine cranes.

A popular crane type used offshore is the knuckleboom crane, Figure 1.1b. It is similar to a boom crane, but with an added knuckle that provides an additional degree of freedom for the upper part of the boom. This allows the crane to fold when not in use, thereby reducing occupied deckspace. The crane's disadvantage is its high power consumption and maintenance due to a large amount of moving parts. The arms are usually positioned using hydraulic actuators and the load is suspended by a wire whose length is controlled by a winch. Hoisting and lowering of the crane load is accomplished by positioning the arms and controlling wire length.

With the recent focus upon limiting emissions and transitioning into greener technology, reducing power consumptions by using new control laws may be of interest. A vessel's power plant may experience large fluctuations in demand during crane operations. For instance, the power demand may peak when hoisting is initialized or if the crane must react quickly to counteract an external disturbance. This may lead incomplete combustion and increased emissions. However, energy can be recovered during lowering operations and stored in batteries, one of the principles behind hybrid power systems. The stored energy can be used for peak shaving and assisting the power plant during heavy demand periods, thereby reducing incomplete combustion and emissions.

Another interesting field is interconnected vessel-crane dynamics. This involves studying motions that are induced in the crane due to vessel motions, and vice-versa if the crane and its load is large and the vessel small. Active heave compensation is a technique that relies upon knowledge of interconnected dynamics.

The abovementioned discussion has presented several challenges related to marine crane control and that there is potential for developing new control laws for increasing crane performance. During controller design, the first step is usually to derive a mathematical model of the system that is to be controlled, and then design a control law using this model. Testing the control law with the simulation model can provide an initial set of tuning parameters and an estimate of the system response. However, there is no guarantee that the real system will behave similarly when subject to the control law. This may be due to modelling inaccuracies, sensor noise or the PLC (Programmable Logic Controller) that executing the control law. In addition, if the control law is designed to operate under dangerous situations such as harsh weather or hardware failures it may be unsafe, expensive or impossible to run full scale experiments in operating conditions. Another alternative is laboratory experiments with a scaled down version of the physical system.

1.1 Marine Crane Lab

A marine crane lab has newly been designed and constructed at the department of marine technology. This section will present the state of the lab at the beginning of this thesis and the relevancy to its work. The lab consists of a scaled down knuckleboom crane with three degrees of freedom and its design was done as part of the master thesis of (Gyberg, 2017).

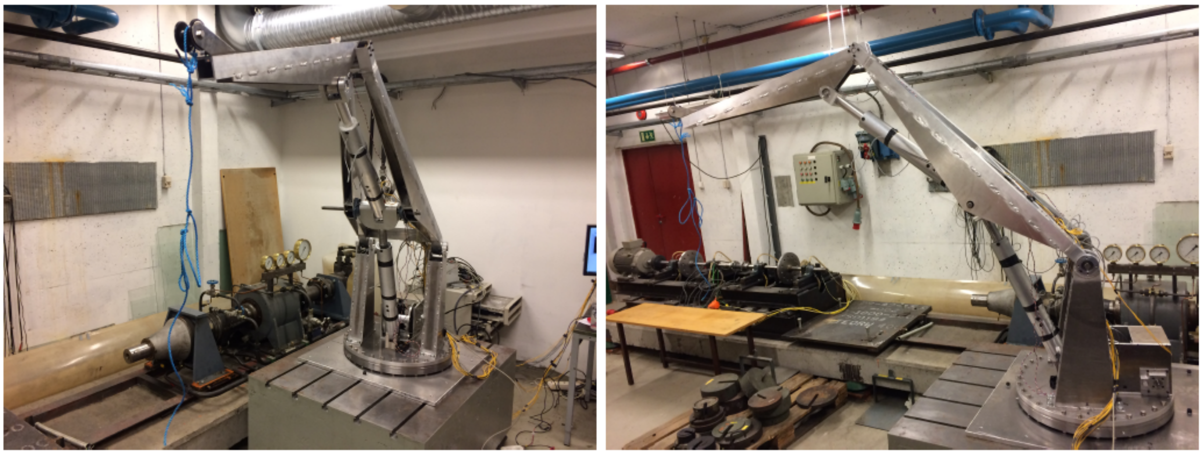


Figure 1.2: The lab crane as it appeared at the beginning of this thesis. Taken from (Gyberg, 2017).

At the beginning of this thesis the crane lab appeared as shown in Figure 1.2. The crane's actuators and winch are controlled by electric motors and a hands-on control system for positioning the arms has been implemented using low-level motor controllers. The actuators are operated by a set of joysticks that provides setpoints to the motor controllers.

The winch has yet to be fitted on the crane, only the reel that stores wire is depicted in Figure 1.2. The belt connecting the winch reel and motor has not been fitted onto the winch and the brake that holds the winch in place is being manufactured. In addition, the low-level motor controller that controls the winch motor must be installed and configured.

A PLC from Bachmann electronics is used as the main controller in the lab. It is responsible for low-level signal handling and for connecting the joysticks and motor controllers. The simple hands-on control system is executed by the PLC.

A set of gyroscopes have been placed on the crane measuring the angular velocity of the arms and base. However, filters for removing noise and estimating bias has not been implemented. The measurement range of the gyroscope are much larger than the expected operating velocity of the crane. This means that the measurement resolution will be low. If the gyroscope bias is to be estimated, a set of accelerometers must be installed, this will also make it possible to measure the absolute orientation of the crane.

In summary, the crane lab is not yet fully completed and some effort must be made to finish the design of (Gyberg, 2017). However, when completed with an operating winch and basic set of sensors, it will allow for a wide range of experiments. For instance, combining the crane with a power simulator would allow research into hybrid power plants. Power simulators for marine vessels have been proposed by (Bø et al., 2015) where the positioning system and power plant of a marine vessel is combined into a single simulator. Combining the simulator with the crane rig could create a powerful tool for power management research.

In addition to power management, the crane lab can also be used to investigate interconnected vessel crane dynamics. This topic has been discussed by (Rokseth et al., 2016) whose work could be further investigated through experiments. The rig's modular design makes it possible to mount it on a barge like mechanism simulating the motions of a vessel. The rig can also be placed on an actual barge making it possible to carry out experiments in the ocean basin at the department of marine technology.

However, before any of these experiments can be carried out, the lab must be finished and a model based control law should be implemented and tested. This will reveal strengths and weaknesses with the design. In order to design a model based control law, a simulation model of the crane and winch must be developed.

1.2 Related Work

Much attention has been given to the modelling of marine cranes. In theory, the modelling procedure is similar to that of robotic manipulators, since marine cranes are in essence large scale manipulators. The most common way of modelling such systems is by using Lagrangian mechanics. This field is well explored and the theoretical foundation well explained in (Ginsberg, 2010).

Lagrangian mechanics is used by (Rokseth, 2014) to model manipulators for remotely operated vehicles, by (Fang et al., 2016) for dynamics analysis and non-linear control, by (Qian and Fang, 2016) to model an offshore ship-mounted crane subject to wave disturbances and also by (Alibeji and Sharma, 2017) and (Smith et al., 2016). (Rokseth et al., 2016) uses Lagrangian mechanics to model an interconnected vessel-crane system.

Crane models developed by Lagrangian mechanics will be highly non-linear due to the presence of non-linear inertia matrices and Coriolis and centripetal effects. Hence, control laws can be designed by either linearising the system and use linear systems theory or use non-linear stability theory. The focus of this thesis will be on designing non-linear control laws. One of the most popular methods for non-linear control is Lyapunov stability theory. A subject that is thoroughly explored in (Khalil, 2014). Of special interest is the backstepping procedure which can be used to design stabilizing controllers for non-linear systems.

Winch dynamics are treated by (Skjong. and Pedersen, 2014) for a marine crane with a hydraulic actuated winch. The work describes how the inertia and diameter of the winch reel can be modelled as functions of the amount of wire left on the reel. It also proposes a wire model that can be used to predict the pendulum motions of the crane load. Even though the winch is hydraulic actuated, it is simple to replace the hydraulics with an electric motor such that the model can be used to represent the winch in the crane lab.

1.3 Scope of Work

The previous discussions has shown that there are many different approaches that can be taken to further develop the crane lab, too many to be treated in a single master thesis. There are also limits to what can be achieved in the lab today, and since the crane is a work in progress, effort should be undertaken to complete its fundamental components.

First, work on the winch should be finished as soon as possible. This will complete the design envisioned by (Gyberg, 2017) and open up the possibility of experimenting with control laws handling lowering and lifting of cargo. By finishing the winch, the lab will be equipped with a functioning crane capable of performing basic crane operations.

Parallel to finishing the winch, a model describing the winch dynamics will be developed. This model will allow the prediction of winch performance and make it possible to develop model based control laws. The main modelling uncertainties will be the parameters used in the model, especially those determining the transmission between load, electrical motor and winch. A similar system, using hydraulic actuation, have been treated by (Skjong. and Pedersen, 2014) and the same procedure will be used in this thesis.

The winch dynamics will be used to design a control law for softly landing the crane load. However, the definition of soft landing must be decided. It can be interesting to investigate a tracking approach, where the system tries to follow a defined path and speed profile.

(Gyberg, 2017) suggests that more sophisticated controllers for positioning the crane should be developed and implemented. The low-level motor controllers are fast and the hands-on control is responsive, but reaching a desired crane position must be done manually and there is not possible to follow a defined reference signal. A model based position controller can make path following possible and enable the crane to follow advanced reference signals.

Before a model based position control law can be designed, a model describing the crane dynamics must be constructed. The model will serve the same purpose as the winch model in predicting the performance of the crane and providing the foundation for the control law design. A model has already been developed by (Gyberg, 2017), but the bond-graph implementation is not suited for the work in this thesis. Instead, a simplified model inspired by the same procedures will be constructed and put on state-space form. The modelling procedure will be closely related to the techniques used to model robotic manipulators.

In summary, the work presented in this thesis seeks to investigate control systems for marine cranes. The focus will be on deriving the equations of motion for a three degrees of freedom knuckleboom crane and the dynamics of its winch system. The crane dynamics will be used to design a control law for positioning the joints at a desired orientation or follow a desired reference signal. The winch dynamics will be used to investigate control laws for softly landing the cargo of the crane. Both the control laws will be tested in a newly constructed crane lab at the department of marine technology.

The crane lab is in the prototype stage and at the beginning of this thesis, only a simple control system without feedback has been implemented. In addition, the winch system has not been implemented. The secondary motivation of this thesis is therefore to contribute to the development of the crane lab by finishing the implementation of the winch, add additional sensors, design sensor filters and improve the user friendliness of the lab.

The main goals and scope of work can be summarized as

- Finish implementation of the winch.
- Construct a simulation model of the crane dynamics.
- Construct a simulation model of the winch.
- Design a control law for soft-landing.
- Design a control law for positioning the crane joints.

- Install accelerometers in the lab and design sensor filters for removing noise and estimating gyro bias.

- Propose further extensions to the lab

1.4 Software

This section will briefly present the software that will be used in this thesis.

Matlab

Matlab (matrix laboratory) is a numerical computing environment and programming language developed and maintained by MathWorks. The work in this thesis will rely on some of Matlab's many toolboxes, its ability to quickly and efficiently process matrices and its plotting capabilities.

Simulink

Simulink a toolbox for Matlab adding a block diagram environment for multidomain simulation. The equations of motions governing the crane dynamics will be implemented and simulated using Simulink.

Symbolic Math Toolbox

The Symbolic Math Toolbox adds functionality for solving and manipulating symbolic math equations to Matlab. In this thesis the toolbox will be used to define the geometry of the crane and solve Lagrange's equation. Thereby reducing work involved and reducing the chance of mathematical errors.

1.5 Structure of the Thesis

The introduction will finish by introducing the structure of the rest of the thesis. A small summary of each section and the attached appendices will follow.

Chapter 2 provides the theoretical foundation needed to understand and accomplish the goals of this thesis. The chapter includes introduction to kinematics of rigid bodies, Lagrangian mechanics, modelling of robotic manipulators and non-linear stability theory.

Chapter 3 presents the derivation of the equations of motion of the knuckleboom crane introduced in Section 1.1. The modelling proccedure will be based on the Lagrangian mechanics presented in Chapter 2. In addition, a dynamic model of the crane winch will be derived. Both models will be implemented as simulation models in Simulink

Chapter 4 presents the design of control laws for positioning the orientation of the knuckle-boom crane and softly landing cargo. The control law design will be based on the simulation models developed in Chapter 3 and will be designed using the non-linear control theory presented in Chapter 2.

Chapter 5 presents the marine crane lab, its equipment and the extension that have been made in order to test the control laws. This chapter will also include the design of a low-pass filter for removing gyroscope noise and a Kalman filter for estimating gyro bias.

Chapter 6 presents the results of the tests run in the marine crane lab with the control laws designed in Chapter 4. The chapter will also present the performance of the low-pass and Kalman filters.

Chapter 7 presents the final discussion and conclusion of this thesis and some recommendation for further work.

Appendix A.1 presents the Matlab script for generating the crane model and position control law.

Appendix A.2 presents the Matlab code for initializing the crane simulation.

Appendix A.3 presents the Matlab script for generating the winch model and soft landing control law.

Appendix A.4 presents the Matlab script for initializing the winch simulations.

Appendix A.5 presents the Matlab script for generating the speed profile used in the soft landing control law.

Chapter 2

Background Material

This chapter will give an overview of the most essential theory needed to accomplish the goals of this thesis. The objective is not to provide textbook detailed introductions to all topics, but to act as reference and small introduction to areas that may be unfamiliar to the reader.

The chapter will start by introducing kinematics of rigid bodies and Langrangian mechanics, which will be essential when modelling the knuckleboom crane from Section 1.1. Next, an introduction to the modelling procedures of robotic manipulators will be given, as these will be useful when putting the crane model on state-space form. The chapter will finish by discussing non-linear systems, Lyapunov stability and backstepping.

2.1 Rigid Body Kinematics

Rigid body kinematics is concerned with describing positions, velocities and accelerations of moving points, bodies or systems of bodies. In this thesis it will be used to describe the kinematics of the knuckleboom crane from Section 1.1.

A common way of describing the position of a particle in three dimensional space is extrinsic coordinates. These are coordinates that are independent of the particle's path and can be described by using the three unit vectors. Meaning that the position of a particle relative to the origin of the coordinate frame can be given by

$$\boldsymbol{r} = x(t)i + y(t)j + z(t)k \tag{2.1}$$

and the particle velocity and acceleration is simply given by the time derivatives of (2.1)

$$\boldsymbol{v} = \dot{\boldsymbol{r}} = \dot{x}(t)i + \dot{y}(t)j + \dot{z}(t)k + \boldsymbol{\omega} \times \boldsymbol{r} \tag{2.2}$$

$$\boldsymbol{a} = \dot{\boldsymbol{v}} = \ddot{x}(t)i + \ddot{y}(t)j + \ddot{z}(t)k + \boldsymbol{\omega} \times \boldsymbol{v} \tag{2.3}$$

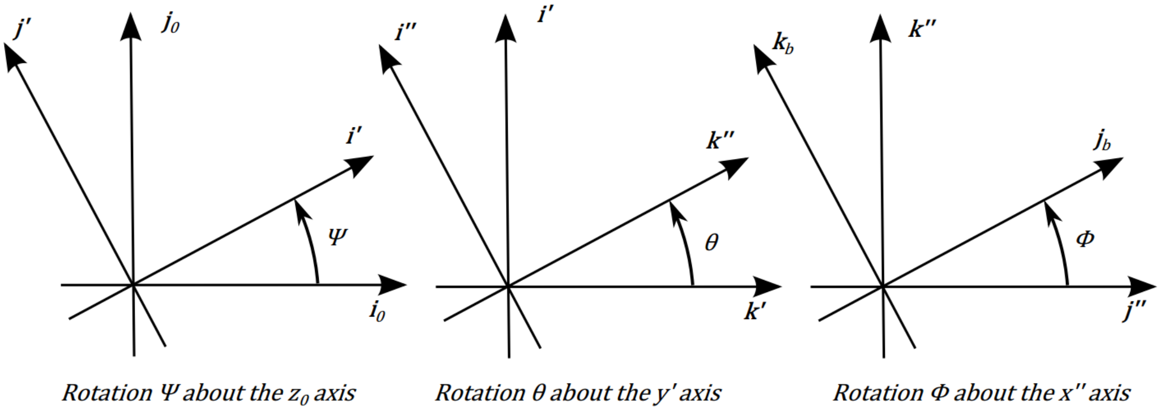
where $\boldsymbol{\omega}$ is the particle's, or coordinate frame's, angular velocity.

(2.1) through (2.3) provides a mathematical description of particle motions relative to the origin of a reference frame, and even accounts for angular motions of the unit axes. However, the discussion so far has only accounted for a single coordinate system and set of unit vectors. In order to properly describe the motions of multiple rigid bodies, this framework must be extended to include multiple reference frames. The next section will therefore introduce the concept of coordinate transformations.

2.1.1 Coordinate Transformations

Modelling systems of linked rigid bodies often necessitates the use of multiple reference frames, because some positions are more conveniently expressed in a local reference frame. However, it is often desired to know these positions relative to a global or inertial reference frame. In fact, when deriving a system's equations of motion by using Lagrange's equation, velocities and positions defining system energies must be expressed in an inertial frame. Coordinate transformations provide the set of tools that can be used to transform vectors between reference frames by using the relative orientation of the frames.

Coordinate transformations can be defined in several ways, this thesis will use the euler angle definition. The three basic rotations are presented in Figure 2.1 showing the rotation of a reference frame about each of the three unit axis.



Rotation Ψ about the z_0 axis

Rotation θ about the y' axis

Rotation Φ about the x'' axis

Figure 2.1: The three principal rotations abut each of the three unit axes. Figure taken from (Rokseth, 2014)

Each of the three principal rotations can be described by a rotation matrix that transforms a vector from one reference frame to the other. The three basic rotation matrices can be written as

$$
\begin{aligned}
\boldsymbol{R}_x &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) \\ 0 & -\sin(\theta) & \cos(\theta) \end{bmatrix} \boldsymbol{R}_y = \begin{bmatrix} \cos(\phi) & 0 & -\sin(\phi) \\ 0 & 1 & 0 \\ \sin(\phi) & 0 & \cos(\phi) \end{bmatrix} \\
\boldsymbol{R}_z &= \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}
\end{aligned}
\tag{2.4}
$$

where θ, ϕ and ψ are the orientations about the x-, y- and z-axes. Referring to Figure 2.1, $\boldsymbol{R}_x(\theta)$ transforms a vector from $j''k''$ to $j_b k_b$. In addition, the rotations matrices give by (2.4) have the following property

$$\boldsymbol{R}^{-1} = \boldsymbol{R}^T \tag{2.5}$$

meaning that the reverse transformation is given by the transpose of the rotation matrix.

Rotation about an arbitrary axis can be expressed by a combination of the three principal rotations (Ginsberg, 2010). For instance, if a frame is first rotated about the x-axis and then the y-axis, the total rotation can be described by

$$\boldsymbol{e}' = \boldsymbol{R}_y \boldsymbol{R}_x \boldsymbol{e}_0 \tag{2.6}$$

$$\boldsymbol{e}_0 = \boldsymbol{R}_x^T \boldsymbol{R}_y^T \boldsymbol{e}' \tag{2.7}$$

where $\boldsymbol{e}_0$ and $\boldsymbol{e}'$ are the fixed and rotating frames.

2.1.2 Vector Notation

Working with multiple reference frames requires a strict notation expressing which reference frame a vector is described in. This simplifies the process of book keeping vectors and avoids confusion. In this thesis the notation employed by (Rokseth, 2014) will be used

$$\boldsymbol{r}^i_{j/k} \tag{2.8}$$

where i is the reference frame the vector is expressed in and j is the point relative to the origin of reference frame k.

The framework for describing the motion of particles in multiple reference frames has now been laid and Lagrangian mechanics can be introduced.

2.2 Lagrangian Mechanics

This section will introduce Lagrangian mechanics and it will be shown how the equations of motion of a system of rigid bodies can be derived by considering the system's kinetic and potential energy. The first topic will be generalized coordinates, followed by the definition of potential and kinetic energy. Finally Lagrange's equation will be introduced and the section will finish by discussing generalized forces.

2.2.1 Generalized Coordinates

Generalized coordinates are a set of variables that uniquely describe the position of a system, relative to some reference (Ginsberg, 2010). The minimum number of generalized coordinates that are required are called the system's degrees of freedom. However, there are often several sets of variables that can be chosen as the generalized coordinates and their selection will influence the final form of the system dynamics, since the number of equations of motion given by Lagrange's equation is the same as the number of generalized coordinates.

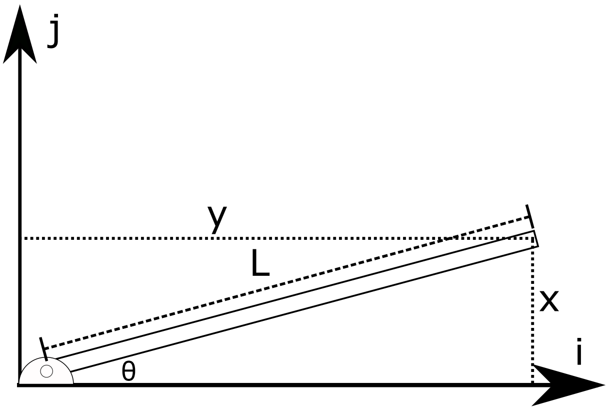


Figure 2.2: A generic beam with one fixed end. The beam is allowed to rotate about the joint and the orientation relative to the i axis is θ.

For instance, each position on the fixed beam shown in Figure 2.2 can be described using its orientation relative to the i axis. The position along i and j are given by

$$x = \cos(\theta)h \tag{2.9a}$$

$$y = \sin(\theta)h \tag{2.9b}$$

where h is a fraction of the beam length L. Therefore, the example beam has one generalized coordinate, θ, and since this is the smallest number of coordinates required, it is also the beam's degree of freedom.

2.2.2 Potential and Kinetic Energy

The kinetic energy of a system is defined by the velocity of its mass centres. The general expression for a system consisting of n rigid bodies can be written as

$$T = \sum_{i=1}^{n} \boldsymbol{v}_i^T m_i \boldsymbol{v}_i \tag{2.10}$$

where T is the total kinetic energy, $\boldsymbol{v}_i$ is the velocity of mass centre i and m_i the mass. When applying (2.13) in Lagrangian mechanics it is important that the velocity vectors are given in terms of generalized coordinates, and that they are expressed in the same inertial reference frame.

A system's potential energy is given by the relative position of its mass centres in the gravity field and other storage elements. In general, the potential energy for a system of n bodies and m storage elements is given by

$$V = \sum_{i=1}^{n} m_i g h_i + \sum_{i=1}^{m} k_i(\boldsymbol{x}) \tag{2.11}$$

where h_i is the position of mass centre i along the axis where gravity is acting, g is the gravitational acceleration and $k_i(\boldsymbol{x})$ is other contributions to the potential energy such as springs.

The kinetic and potential energy, given by (2.10) and (2.11), are essential in Lagrangian mechanics and now that they have been defined, Lagrange's equation can be presented.

2.2.3 Lagrange's Equation

Lagrange's equation is a specialized version of Hamilton's principle for systems with a discrete number of generalized coordinates (Ginsberg, 2010). The equation is formulated by the kinetic and potential energy of the system that it describes and is sometimes given in terms of the Lagrangian

$$L = T - V \tag{2.12}$$

where T is the kinetic energy given by (2.10) and V the potential energy given by (2.11). Hence, for a system of a finite number of rigid bodies, described by a set of n generalized coordinates, Lagrange's equation is given by

$$\frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{q}_k}\right) - \frac{\partial L}{\partial q_k} = Q_k \qquad k = 1, 2, \cdots, n \tag{2.13}$$

where n is the number of generalized coordinates, q_k and $\dot{q}_k$ is the kth generalized coordinate and velocity, L the Lagrangian and Q_k generalized forces influencing the kth generalized coordinate. Lagrange's equation is in fact a representation of the conservation of energy and momentum. The term $\frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{q}_k}\right)$ represents the change in momentum, while $\frac{\partial L}{\partial q_k}$ is the change in potential energy.

If $Q_k = 0$, then there are no external forces acting on the system and the change in energy is zero. Meaning that the system is conservative and no energy is added or dissipated. However, if $Q_k \neq 0$, then there are non-conservative forces influencing the total energy. Q_k usually represents forces such as resistance, damping or actuators. Modelling generalized forces will be presented in the next section.

2.2.4 Generalized Forces

The term Q_k is called generalized forces and represents the non-conservative forces and moments acting on the system. It is used to represent external forces that can not be described or covered by the system's potential and kinetic energy. For instance, friction and resistance are not conservative forces since they remove energy from the system and causes a decrease in the total energy, without the possibility of recovery. Another example is the actuators controlling the motion of a robotic manipulator. According to (Ginsberg, 2010) a force or moment $\boldsymbol{F}$ acting at a position $\boldsymbol{r}_F$ is described as a generalized force by

$$Q_k = \boldsymbol{F}_k^T \frac{\partial \boldsymbol{r}_{F_k}}{\partial q_k} \tag{2.14}$$

where the position and force vector must be given in an inertial reference frame and in terms of the generalized coordinates.

2.3 Modeling of Robotic Manipulators

The knuckleboom crane introduced in Section 1.1 is not different from a robotic manipulator, in terms of mathematical modelling. Therefore, an introduction to procedures for modelling robotic manipulators will be presented. This will cover state-space formulation of manipulators, Jacobians, derivations of the generalized inertia matrix, the Coriolis and centripetal matrix, the gravity vector, external forces and how the system matrices are connected to Lagrange's equations (2.13).

2.3.1 State-space Formulation of Robotic Manipulators

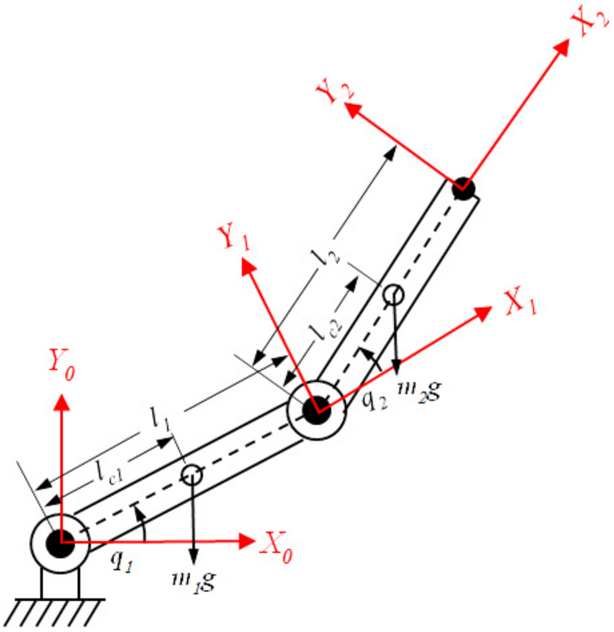


Figure 2.3: A generic robotic manipulator with joint angles q_1 and q_2 and three defined reference frames. From (Scilab, 2017).

A generic robotic manipulator consisting of two linked arms is shown in Figure 2.3. The figure also shows the placement of three coordinate frames: X_0Y_0, X_1Y_1 and X_2Y_2, Where X_0Y_0 is the inertial reference frame and the other two are aligned with the manipulator arms. The placement of the mass centre of each arm is also defined.

Such robotic manipulators can easily be modelled using Lagrange's equation (2.13) and the particle kinematics and coordinate transformations presented in section 2.1. The resulting equations of motion can be written compactly as

$$\boldsymbol{D}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} \tag{2.15}$$

where $\boldsymbol{M}$ is the inertia matrix, $\boldsymbol{C}$ the centripetal and Coriolis matrix, $\boldsymbol{g}$ the gravity vector and $\boldsymbol{\tau}$ external forces.

However, the relationship between Lagrange's equation and the terms in (2.15) is not explicitly clear. For instance, the general inertia matrix can be derived directly from the kinetic energy and Jacobian of the system, and one representation of the Coriolis matrix is given by the partial derivatives of $\boldsymbol{D}$ with respect to the generalized variables. In the coming sections the Jacobian will be defined and the inertia and Coriolis matrices and gravity vector in (2.15) will be connected directly to the kinetic and potential energy in Langrange's equation.

2.3.2 Jacobian

The Jacobian is one of the most important quantities in the analysis and control of robot motion and arises in virtually every aspect of robotic manipulation, Spong et al. (2006). In the context of this thesis, it defines the transformation from generalized to body velocities, which can be written as

$$\begin{bmatrix} \boldsymbol{v}_n^0 \\ \boldsymbol{\omega}_n^0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}_v \\ \boldsymbol{J}_\omega \end{bmatrix} \dot{\boldsymbol{q}} = \boldsymbol{J}\dot{\boldsymbol{q}} \tag{2.16}$$

where $\boldsymbol{v}_n^0$ and $\boldsymbol{\omega}_n^0$ are the linear and angular body velocities expressed in the inertial frame, $\boldsymbol{J}$ the Jacobian and $\dot{\boldsymbol{q}}$ the generalized velocities.

The definition of the Jacobian will depend upon whether the joints of the manipulator are prismatic or revolute. The ith element is generally given by

$$\boldsymbol{J}_{\boldsymbol{v}_i} = \begin{cases} \boldsymbol{e}_{i-1} \times (\boldsymbol{o}_n - \boldsymbol{o}_{i-1}); & \text{for revolute joint } i \\ \boldsymbol{e}_{i-1}; & \text{for prismatic joint } i \end{cases} \tag{2.17}$$

$$\boldsymbol{J}_{\boldsymbol{\omega}_i} = \begin{cases} \boldsymbol{e}_{i-1}; & \text{for revolute joint } i \\ 0; & \text{for prismatic joint } i \end{cases} \tag{2.18}$$

where $\boldsymbol{e}_{i-1}$ is the axis of rotation or parallel to translation, $\boldsymbol{o}_n$ is position of the point and $\boldsymbol{o}_{i-1}$ the origin of a preceding reference frame. For the generic manipulator in Figure 2.3, the Jacobian for mass centre m_2 will be

$$\boldsymbol{J} = \begin{bmatrix} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \times (\boldsymbol{o}_{m_2} - \boldsymbol{o}_0) & \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \times (\boldsymbol{o}_{m_2} - \boldsymbol{o}_1) \\ \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T & \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \end{bmatrix} \tag{2.19}$$

The next section will show how the generalized inertia matrix can be derived from the Jacobian.

2.3.3 Inertia Matrix

The inertia matrix in (2.15) can be directly derived from the kinetic energy calculated as part of the Lagrangian formulation. An alternative expression of (2.10) for a robotic manipulator is given by (Spong et al., 2006) as

$$T = \frac{1}{2}\dot{\boldsymbol{q}}^T \left[\sum_{i=1}^{n} (m_i \boldsymbol{J}_{\boldsymbol{v}_i}^T \boldsymbol{J}_{\boldsymbol{v}_i} + \boldsymbol{J}_{\boldsymbol{\omega}_i}^T \boldsymbol{R}_i(\boldsymbol{q}) \boldsymbol{I}_i \boldsymbol{R}_i(\boldsymbol{q})^T \boldsymbol{J}_{\boldsymbol{\omega}_i}) \right] \dot{\boldsymbol{q}} \tag{2.20}$$

where the inertia matrix is given by

$$\boldsymbol{D}(\boldsymbol{q}) = \left[\sum_{i=1}^{n} (m_i \boldsymbol{J}_{\boldsymbol{v}_i}^T \boldsymbol{J}_{\boldsymbol{v}_i} + \boldsymbol{J}_{\boldsymbol{\omega}_i}^T \boldsymbol{R}_i(\boldsymbol{q}) \boldsymbol{I}_i \boldsymbol{R}_i(\boldsymbol{q}) T \boldsymbol{J}_{\boldsymbol{\omega}_i}) \right] \tag{2.21}$$

where $\dot{\boldsymbol{q}}$ is the generalized velocities, $\boldsymbol{J}_{\boldsymbol{v}_i}$ the linear velocity Jacobian, $\boldsymbol{J}_{\boldsymbol{\omega}_i}$ the angular Jacobian, m_i mass and $\boldsymbol{I}_i$ the moment of inertia of body i. $\boldsymbol{R}_i$ is the rotation matrix from the local reference frame of body i to the inertial reference frame.

2.3.4 Centripetal and Coriolis Matrix

The Coriolis and centripetal matrix represents the fictional Coriolis forces that acts on the system due to motions relative to a rotating reference frame. According to (Spong et al., 2006), it can be diectly derived from the inertia matrix by using the first order Christoffel symbols. The kjth element in the Coriolis and centripetal matrix is given by

$$c_{kj} = \sum_{i=1}^{n} \frac{1}{2} \left(\frac{\partial d_{kj}}{\partial q_j} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k} \right) \dot{q}_i \tag{2.22}$$

where d_{kj} is the kjth element of the generalized inertia matrix (2.21) and q_k is the kth generalized coordinate.

2.3.5 Gravity Vector

The system's gravity vector is derived directly from the potential energy (2.11). Each term in the gravity vector in (2.15) can be found by partially differentiating the potential energy with respect to the generalized coordinates. The kth element of the gravity is therefore given by

$$g_k = \frac{\partial V}{\partial q_k} \tag{2.23}$$

where V is the potential energy given by (2.11) and q_k the kth generalized coordinate.

2.3.6 External Forces

The term $\boldsymbol{\tau}$ on the right hand side of (2.15) collects all external forces acting on the manipulator and is directly connected with the generalized forces Q_k in Lagrange's equation (2.13). The general expression is given by (2.14), but for some system it may given directly. Most importantly, the term includes the actuator forces which are used to control the manipulator.

2.4 Non-Linear Systems

The robotic manipulator model will, when used to model the knuckleboom crane from Section 1.1, be highly non-linear. This means that non-linear control theory must be used to design stabilizing controllers for the system. This section will give an overview of the most essential non-linear control theory that will be used in this thesis and will include representations of non-linear systems, Lyapunov stability theory and backstepping.

2.4.1 Representation of Nonlinear Systems

The work in this thesis will deal with non-linear dynamic equations that can be written as a set of lumped, first order, time invariant differential equations. In general, such systems can be written as

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u} \tag{2.24}$$

where $\dot{\boldsymbol{x}}$ is the state vector, $\boldsymbol{f}(\boldsymbol{x})$ the system matrix, $\boldsymbol{g}(\boldsymbol{x})$ the input matrix and $\boldsymbol{u}$ the input vector.

2.4.2 Lyapunov Stability Theory

Lyapunov stability theory is a powerful set of tools that can be used to design control laws for non-linear systems and prove their stability. It will be widely used in this thesis in order to design the control laws mentioned in Section 1.3 and the most relevant topics will be covered in the following section. This includes asymptotic stability, Lyapunov functions, Lyapunov's equation and backstepping.

Lyapunov Function

Lyapunov functions are essential for designing non-linear control laws and analysing the stability of non-linear systems. Their definition is given by (Khalil, 2014) in the following theorem.

Theorem 2.4.1. *Let $f(x)$ be a locally Lipschitz function defined over a domain $D \subset \mathbb{R}^n$, which contains the origin, and $f(0) = 0$. Let $V(x)$ be a continuously differentiable function defined over D such that*

$$V(0) = 0 \quad \textit{and} \quad V(x) > 0 \quad \textit{for all} \quad x \in D \quad \textit{with} \quad x \neq 0 \tag{2.25}$$

$$\dot{V}(x) \leq 0 \quad \textit{for all} \quad x \in D \tag{2.26}$$

Then, the origin is a stable equilibrium point of $\dot{x} = f(x)$. Moreover if

$$\dot{V}(x) < 0 \quad \textit{for allx} \in D \quad \textit{with} \quad x \neq 0 \tag{2.27}$$

then the origin is asymptotically stable. Furthermore, if $D = \mathbb{R}^n$ *and* (2.25) *and* (2.27) *hold for all* $x \neq 0$*, and*

$$||x|| \to \infty \implies V(x) \to \infty \tag{2.28}$$

then, the origin is globally asymptotically stable.

Theorem 2.4.1 states that if a function that satisfies (2.25) exists and its derivative is negative definite for all time and for all x in the function's domain, then the origin of $f(x)$ will be globally asymptotically stable. Now, the definition of asymptotic stability will be given.

Asymptotic Stability

Non-linear systems may exhibit several types of stability. They can be locally or globally stable and converge to the origin at different rates. Usually the rate of convergence is what defines different types of stability. In this work only the notion of asymptotically stability will be discussed, since this type of stability is usually achieved when using the backstepping method. The reader is directed to (Khalil, 2014) for more on stability of non-linear systems.

Before asymptotic stability is defined, the definitions of κ, ℓ and $\kappa\ell$ class functions will be given.

Definition 1. *A continuous function* α*:* $[0, a) \to [0, \infty)$ *is called a class-*κ *function if*

- $\alpha(0)$ = *0*
- *it is strictly increasing*

Definition 2. *A continuous function* ϕ*:* $[0, \infty) \to [0, \infty)$ *is called a class-*ℓ *function if*

- *it is decreasing*
- $\lim_{s\to\infty} \phi(s) = 0$

Definition 3. *A function* β*:* $[0, \infty) \times [0, \infty) \to [0, \infty)$ *is called a class-*$\kappa\ell$ *function if*

- *for each fixed s, the function* $\beta(r, s)$ *belongs to class* κ
- *for each fixed r, the function* $\beta(r, s)$ *belongs to class* ℓ

Uniform, global, asymptotic stability can now be defined in terms of κ, ℓ and $\kappa\ell$ class functions from 1, 2 and 3.

Definition 4. *The origin* $\boldsymbol{x} = 0$ *of* $\dot{\boldsymbol{x}} = \boldsymbol{f}(t, \boldsymbol{x})$*, with* $\boldsymbol{x}_0 = \boldsymbol{x}_0(t)$*, is uniformly Globally Asymptotically Stable (UGAS) if* $\exists \beta \in \kappa\ell$ *such that,* $\forall x_0 \in \mathbb{R}^n$*, the solution* $x(t, x_0)$ *satisfies*

$$|x(t, x_0)| \leq \beta(|x_0|, t), \qquad \forall t \geq 0 \tag{2.29}$$

For non-linear systems, asymptotic stability, as given by Definition 4, can be proved through Lyapunov's direct method (Khalil, 2014).

Theorem 2.4.2. *Assume the system $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, t)$ is forward complete.*

- *If there exists a smooth Lyapunov function for the system $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, t)$, with respect to the origin $\boldsymbol{x} = 0$, then the origin is uniformly globally stable*
- *If $\dot{V} \leq \alpha(|\boldsymbol{x}|)$, where V is a Lyapunov function candidate for $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, t)$, and $\alpha(|\boldsymbol{x}|)$ is a negative definite function, then the origin $\boldsymbol{x} = 0$ is uniformly globally asymptotically stable (UGAS).*

Hence, through Definition 4 and Theorems 2.4.3 and 2.4.2 asymptotic stability of non-linear systems can be proven. However, finding functions which satisfies Lyapunov's direct method is not always trivial, but there exist techniques that can achieve it for certain types of systems.

Lyapunov's Equation

The Lyapunov equation often occurs when analysing stability of non-linear systems. It is especially useful when designing controllers through backstepping and helps choosing gains that ensures asymptotic stability. By (Khalil, 2014), the Lyapunov equation is given as

$$\boldsymbol{A}^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}^T = -\boldsymbol{Q} \tag{2.30}$$

and connected with the following theorem.

Theorem 2.4.3. *A matrix $\boldsymbol{A}$ is Hurwitz if and only if for every positive definite symmetric matrix $\boldsymbol{Q}$ there exists a positive definite matrix $\boldsymbol{P}$ that satisfies the Lyapunov equation $\boldsymbol{A}^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}^T = -\boldsymbol{Q}$. Moreover, if $\boldsymbol{A}$ is Hurwitz, then $\boldsymbol{P}$ is the unique solution.*

Where a matrix $\boldsymbol{A}$ is Hurwitz if every eigenvalue has a strictly negative real part. With Lyapunov functions defined, the notion of asymptotic stability explored and given the Lyapunov equation, backstepping can be presented.

2.4.3 Backstepping

Although Lyapunov functions are an extremely powerful tool for stabilizing non-linear systems, the design procedure is not always straight forward. The backstepping method provides a systematic way of recursively designing control laws for strict-feedback systems of the form

$$\begin{aligned}
\dot{x} &= f_0(x) + g_0(x)z_1 \\
\dot{z}_1 &= f_1(x, z_1) + g_1(x, z_1)z_2 \\
\dot{z}_2 &= f_2(x, z_1, z_2) + g_2(x, z_1, z_2)z_3 \\
&\vdots \\
\dot{z}_{k-1} &= f_{k_1}(x, z_1, \cdots, z_{k_1}) + g_{k-1}(x, z_1, \cdots, z_{k-1})z_k \\
\dot{z}_k &= f_k(x, z_1, \cdot, z_k) + g_k(x, z_1, \cdots, z_k)u
\end{aligned} \tag{2.31}$$

where $x \in \mathbb{R}^n$, z_1 to z_k are scalars, $\boldsymbol{u}$ the input, f_0 to f_k vanish at the origin and

$$g_i(x, z_1, ..., z_i) \neq 0 \qquad \text{for} 1 \leq i \leq k \tag{2.32}$$

over the domain of interest (Khalil, 2014).

The recursive procedure start with a change of variables defined by

$$\begin{aligned} \boldsymbol{z}_1 &= \boldsymbol{y} - \boldsymbol{y}_d \\ \boldsymbol{z}_2 &= \boldsymbol{x}_2 - \boldsymbol{\alpha}_1 \\ &. \\ &. \\ &. \\ \boldsymbol{z}_n &= \boldsymbol{x}_n - \boldsymbol{\alpha}_n \end{aligned} \tag{2.33}$$

where $\boldsymbol{z}_i$ is the backstepping variables, $\boldsymbol{\alpha}_i$ a virtual controller that will be used to stabilize $\boldsymbol{z}_{i-1}$, $\boldsymbol{y}_d$ the system's output vector and $\boldsymbol{y}_d$ the desired that the system should follow.

The number of backstepping steps required are determined by the relative degree of the system. For each step a Lyapunov function is defined as

$$V_i = \Sigma_{j=1}^{i-1} V_j + \boldsymbol{z}_i^T \boldsymbol{P}_i \boldsymbol{z}_i \tag{2.34}$$

where $\boldsymbol{P}_i$ is a positive definite matrix satisfying the Lyapunov equation (2.30) and theorem 2.4.3. The virtual controller, $\boldsymbol{\alpha}_i$, is chosen such that $\dot{V}_i$ is rendered negative definite in terms of $\boldsymbol{z}_i$. This ensures asymptotic stability by Lyapunov's direct method (theorem 2.4.2). Terms related to $\boldsymbol{z}_{i+1}$ is ignored and carried over to the next step. This process is repeated until the control input is reached.

The final step is reached when the derivative of (2.34) contains the control input $\boldsymbol{u}$, which is chosen such that the final Lyapunov derivative is rendered negative definite. The final Lyapunov function is then given by the sum of Lyapunov functions for all steps and since each step is asymptotically stable, so is the total system.

However, backstepping only proves asymptotic stability for $\boldsymbol{z}$ and does not guarantee that stability is carried over to $\boldsymbol{x}$. Asymptotic stability is only preserved if the transformation $\boldsymbol{T}(x)$ from $\boldsymbol{x}$ to $\boldsymbol{z}$ is a global diffeomorphism (Khalil, 2014). This means that $\boldsymbol{T}(x)^{-1}$ must be defined over the domain of the system.

In summary, this chapter has provided the set of tools that will make it possible to construct a mathematical model of the knuckleboom crane and winch from Section 1.1 and design non-linear controllers for these systems.

Chapter 3

Simulation Models

In this chapter models of the knuckleboom crane and winch system from Section 1.1 will be developed using Lagrangian mechanics. The models will later in the thesis be used to design control laws for positioning the crane joints and soft landing. For each model, the modelling procedure will be described in detail and finish with a simulation study verifying its theoretical behaviour.

3.1 Crane Model

This section will use Lagrangian mechanics to derive the equations of motion for the knuckleboom crane presented in Section 1.1. The modelling procedure will start by defining a set of reference frames that will be used to describe each position in the system. Then, a set of generalized coordinates will be selected and the Jacobians of the mass centres derived. The Jacobians will be used to define the potential and kinetic energy of the system, which will be used to derive the system matrices. Afterwards, external forces will be identified and modelled. The section will finish by implementing the model in Simulink and running a simple simulation.

3.1.1 System Description

The first step is to properly define the system that is to be modelled. The knuckleboom crane consists of a rotating base and two arms. The orientation of the base is controlled by an electric motor and the arms are positioned by two electric actuators. In addition, a load that represent the crane cargo acts at the tip of the upper arm. The crane can therefore be represented as a series of linked, rigid bodies and is well suited to be modelled using Lagrangian mechanics. A sketch of the crane is presented in Figure 3.1.

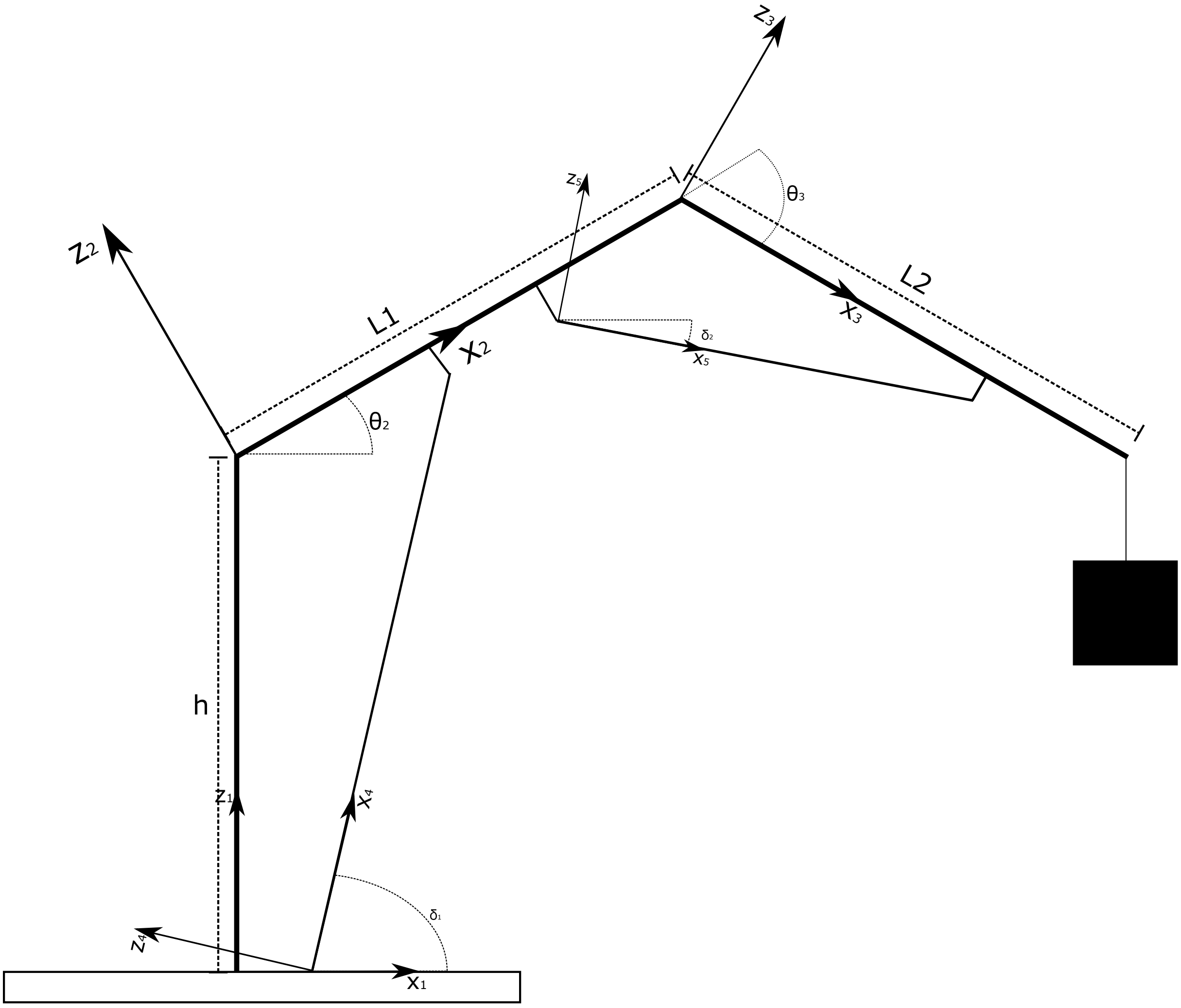


Figure 3.1: Sideview of the crane showing the placement of local reference frames.

3.1.2 Placement of Reference Frames

If the equations of motions are to be derived using Lagrange's equation, all positions and velocities in the crane system must be known and expressed in an inertial reference frame. These variables are essential when defining the kinetic and potential energy of the system. However, it is often more convenient to describe positions in a local reference frame and then use a series of coordinate transformations to transform the vector to the inertial frame. Hence, the reference frames that will be used in the modelling procedure must defined.

A simplified view of the crane seen from the side is presented in Figure 3.1. In addition, the placement of some reference frames and their relative orientation are shown.

The inertial reference frame for this system is placed at the base of the crane with its z-axis aligned with the base column. The frame is denoted $x_0y_0z_0$.

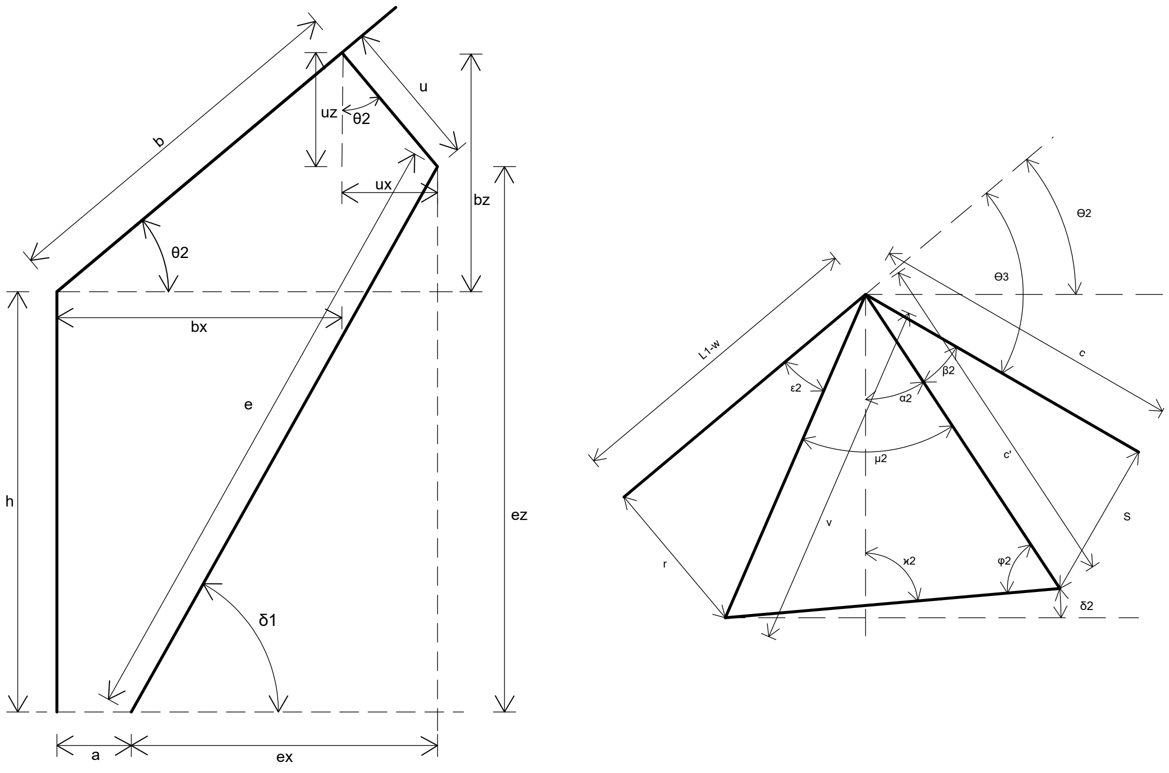
Next, a local reference frame for the base is placed with its origin coinciding with the inertial frame. However, its axes rotates together with the column, about the z-axis, and its relative orientation to the inertial frame is denoted θ_1 and axes $x_1y_1z_1$.

The second local reference frame is placed at connection point between the base and lower arm. Its local x-axis runs parallel to and rotates with the arm, about the local y-axis. Its axes are denoted $x_2y_2z_2$ and its orientation relative $x_1y_1z_1$ is θ_2.

The local reference frame for the upper arm, $x_3y_3z_3$ is placed at its connection point to the lower arm. The local x-axis runs parallel to the arm and it has an orientation θ_3 relative to $x_2y_2z_3$ about the y-axis.

Reference frames following the same principle as for the arms are placed at the lower connection point of each actuator. These are denoted $x_4y_4z_4$ for the lower and $x_5y_5z_5$ for the upper and their orientations are δ_1 and δ_2 relative to $x_1y_1z_1$.

From the previous discussion it is evident that all positions in the system can be described using the five reference frames and orientations θ_1, θ_2, θ_3, δ_1 and δ_2. However, the number of free variables can be reduced by noting that δ_1 and δ_2 are function of the joint orientations.



(a) Lower Actautor

(b) Upper Actuator

Figure 3.2: Upper and Lower Actuator geometries. From (Gyberg, 2017)

The geometry of the lower and upper actuators are shown in Figure 3.2. It can be shown that the orientation of the lower actuator can written as

$$\delta_1(\theta_2) = \sin^{-1}\left(\frac{h + \sin(\theta_2)b - \cos(\theta_2)u}{\sqrt{(\cos(\theta_2)b + \sin(\theta_2)u - a)^2 + (h + \sin(\theta_2)b - \cos(\theta_2)u)^2}}\right) \tag{3.1}$$

where all geometric parameters are defined in Figure 3.2a. A similar analysis of the upper actuator shows that its orientation can be written as

$$\delta_2(\theta_2, \theta_3) = \pi + \theta_2 + \theta_3 - \sin^{-1}\left(\frac{s}{c'}\right) - \cos^{-1}\left(\frac{\cos(\mu_2)v - c'}{\sqrt{v^2 + c'^2 - 2vc'\cos(\mu_2)}}\right) \tag{3.2a}$$

$$\mu_2 = \pi + \theta_3 - \sin^{-1}\left(\frac{s}{\sqrt{c^2 + s^2}}\right) - \sin^{-1}\left(\frac{r}{\sqrt{r^2 + (L_1 - w)^2}}\right) \tag{3.2b}$$

$$v = \sqrt{r^2 + (L_1 - w)^2} \tag{3.2c}$$

$$c' = \sqrt{c^2 + s^2} \tag{3.2d}$$

$$\tag{3.2e}$$

where all geometric parameters are defined in Figure 3.2b. (3.1) and (3.2) shows that the orientations of the actuators can be written as functions of the joint angles.

It has now been shown that all relevant orientations in the system can be expressed using the orientation of the base and arms. Since the crane arms are fixed in length, these orientations can be used to express all position in the system. Therefore, the generalized coordinates are chosen as

$$\boldsymbol{q} = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 \end{bmatrix} \tag{3.3}$$

and since $\boldsymbol{q} \in \mathbb{R}^3$, there will be three differential equations governing the system. The next step is to derive the transformations between the different reference frames and express the position and velocity of all mass centres in the inertial frame.

3.1.3 Relevant Positions

Now that generalized coordinates have been selected and reference frames defined, transformations between the local and inertial frames can be derived and all relevant positions in the system expressed. The relevant positions are the coordinates of the mass centres, origin of the local reference frames and crane tip. Expressing these positions in the inertial reference frame will make it possible to construct the Jacobians that will be used to define the system's kinetic and potential energy.

Most of the reference frames that were defined in the previous section rotates about the local y-axis, with exception of $x_1y_1z_1$ which rotates about the local z-axis. Hence, rotation matrices about the y- an z-axes must be used to transform coordinates. The rotation matrices are presented in Section 2.1.1.

The rotation matrix for each reference frame can be expressed as a series of rotations defined by

$$\boldsymbol{R}_1^0 = \boldsymbol{R}_z(\theta_1)^T \quad \boldsymbol{R}_2^0 = \boldsymbol{R}_1^0\boldsymbol{R}_y(\theta_2)^T \quad \boldsymbol{R}_3^0 = \boldsymbol{R}_2^0\boldsymbol{R}_y(\theta_3)^T \tag{3.4}$$

$$\boldsymbol{R}_4^0 = \boldsymbol{R}_1^0\boldsymbol{R}_y(\delta_1)^T \quad \boldsymbol{R}_5^0 = \boldsymbol{R}_1^0\boldsymbol{R}_y(\delta_2)^T \tag{3.5}$$

where $\boldsymbol{R}_i^k$ defines the rotation from frame i to k.

The origin of each local reference frame can be written as

$$\boldsymbol{r}_{1/0}^0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \boldsymbol{r}_{2/1}^1 = \begin{bmatrix} 0 \\ 0 \\ h \end{bmatrix} \quad \boldsymbol{r}_{3/2}^2 = \begin{bmatrix} 0 \\ 0 \\ L_1 \end{bmatrix} \quad \boldsymbol{r}_{4/1}^1 = \begin{bmatrix} a \\ 0 \\ 0 \end{bmatrix} \quad \boldsymbol{r}_{5/2}^2 = \begin{bmatrix} w \\ 0 \\ -r \end{bmatrix} \tag{3.6}$$

where the geometric parameters are defined in Figures 3.1 and 3.2.

The position of each reference frame relative to the inertial frame, expressed in the inertial frame, is given by

$$\boldsymbol{r}_{2/0}^0 = \boldsymbol{r}_{1/0}^0 + \boldsymbol{R}_1^0\boldsymbol{r}_{2/1}^1 \tag{3.7a}$$

$$\boldsymbol{r}_{3/0}^0 = \boldsymbol{r}_{2/0}^0 + \boldsymbol{R}_2^0\boldsymbol{r}_{3/2}^2 \tag{3.7b}$$

$$\boldsymbol{r}_{4/0}^0 = \boldsymbol{r}_{1/0}^0 + \boldsymbol{R}_1^0\boldsymbol{r}_{4/1}^1 \tag{3.7c}$$

$$\boldsymbol{r}_{5/0}^0 = \boldsymbol{r}_{2/0}^0 + \boldsymbol{R}_2^0\boldsymbol{r}_{5/2}^2 \tag{3.7d}$$

The same procedure can be repeated for the position of the mass centres. In the local frames the mass centres of each body can be written as

$$\boldsymbol{r}_{cg1/1}^1 = \begin{bmatrix} X_{cm1} \\ Y_{cm1} \\ 0 \end{bmatrix} \quad \boldsymbol{r}_{cg2/2}^2 = \begin{bmatrix} k \\ 0 \\ 0 \end{bmatrix} \quad \boldsymbol{r}_{cg3/3}^3 = \begin{bmatrix} n \\ 0 \\ 0 \end{bmatrix} \quad \boldsymbol{r}_{cg4/4}^4 = \begin{bmatrix} p \\ 0 \\ 0 \end{bmatrix} \quad \boldsymbol{r}_{cg5/5}^5 = \begin{bmatrix} o \\ 0 \\ 0 \end{bmatrix} \tag{3.8}$$

where X_{cm1}, Y_{cm1}, k, n, p and o are constant parameters.

The position of all centres of mass relative to the inertial reference frame, expressed in the inertial frame, can be written as

$$\boldsymbol{r}_{cg_1/0}^0 = \boldsymbol{r}_{1/0}^0 + \boldsymbol{R}_1^0\boldsymbol{r}_{cg_1/1}^1 \tag{3.9a}$$

$$\boldsymbol{r}_{cg_2/0}^0 = \boldsymbol{r}_{2/0}^0 + \boldsymbol{R}_2^0\boldsymbol{r}_{cg_2/2}^2 \tag{3.9b}$$

$$\boldsymbol{r}_{cg_3/0}^0 = \boldsymbol{r}_{3/0}^0 + \boldsymbol{R}_3^0\boldsymbol{r}_{cg_3/3}^3 \tag{3.9c}$$

$$\boldsymbol{r}_{cg_4/0}^0 = \boldsymbol{r}_{4/0}^0 + \boldsymbol{R}_4^0\boldsymbol{r}_{cg_4/4}^4 \tag{3.9d}$$

$$\boldsymbol{r}_{cg_5/0}^0 = \boldsymbol{r}_{5/0}^0 + \boldsymbol{R}_5^0\boldsymbol{r}_{cg_5/5}^5 \tag{3.9e}$$

Now that the origin of each reference frame and the position of each mass center relative to inertial frame have been expressed, the Jacobians can be defined.

3.1.4 Jacobians

In order to simplify the derivation of the Jacobians, the unit vectors and following transformations are defined

$$\boldsymbol{e}_1 = \begin{bmatrix}1\\0\\0\end{bmatrix} \quad \boldsymbol{e}_2 = \begin{bmatrix}1\\0\\0\end{bmatrix} \quad \boldsymbol{e}_3 = \begin{bmatrix}0\\0\\1\end{bmatrix} \tag{3.10a}$$

$$\boldsymbol{w}_1 = \boldsymbol{R}_1^0\boldsymbol{e}_3 \quad \boldsymbol{w}_2 = \boldsymbol{R}_2^0\boldsymbol{e}_2 \quad \boldsymbol{w}_3 = \boldsymbol{R}_3^0\boldsymbol{e}_2 \tag{3.10b}$$

meaning that $\boldsymbol{w}_1$ is the z-axis of frame 1, $\boldsymbol{w}_2$ the y-axis of frame 2 and $\boldsymbol{w}_3$ the y axis of frame 3, transformed to the inertial frame. Since there are three generalized coordinates, the Jacobians will be in $\mathbb{R}^{6\times3}$ and since the crane consists of five rigid bodies, at least five Jacobians must be defined. Hence, following the principles from section 2.3.2, the mass center Jacobians can be written as

$$\boldsymbol{J}_1 = \begin{bmatrix}\boldsymbol{w}_1 \times \boldsymbol{r}^1_{cg1/0} & \boldsymbol{0}_{3\times2}\\ \boldsymbol{w}_1 & \boldsymbol{0}_{3\times2}\end{bmatrix} \tag{3.11a}$$

$$\boldsymbol{J}_2 = \begin{bmatrix}\boldsymbol{w}_1 \times \boldsymbol{r}^1_{cg2/0} & \boldsymbol{w}_2 \times (\boldsymbol{r}^0_{cg2/0} - \boldsymbol{r}^0_{2/0}) & \boldsymbol{0}_{3\times1}\\ \boldsymbol{w}_1 & \boldsymbol{w}_2 & \boldsymbol{0}_{3\times1}\end{bmatrix} \tag{3.11b}$$

$$\boldsymbol{J}_3 = \begin{bmatrix}\boldsymbol{w}_1 \times \boldsymbol{r}^1_{cg3/0} & \boldsymbol{w}_2 \times (\boldsymbol{r}^0_{cg3/0} - \boldsymbol{r}^0_{2/0}) & \boldsymbol{w}_3 \times (\boldsymbol{r}^0_{cg3/0} - \boldsymbol{r}^0_{3/0})\\ \boldsymbol{w}_1 & \boldsymbol{w}_2 & \boldsymbol{w}_3\end{bmatrix} \tag{3.11c}$$

$$\boldsymbol{J}_4 = \begin{bmatrix}\boldsymbol{w}_1 \times \boldsymbol{r}_{cg4/0} & \boldsymbol{w}_2 \times \boldsymbol{r}^0_{cg4/0} - \boldsymbol{r}^0_{2/0} & \boldsymbol{0}_{3\times1}\\ \boldsymbol{w}_1 & \boldsymbol{w}_2 & \boldsymbol{0}_{3\times1}\end{bmatrix} \tag{3.11d}$$

$$\boldsymbol{J}_5 = \begin{bmatrix}\boldsymbol{w}_1 \times \boldsymbol{r}_{cg5/0} & \boldsymbol{w}_2 \times \boldsymbol{r}^0_{cg5/0} - \boldsymbol{r}^0_{2/0} & \boldsymbol{w}_3 \times \boldsymbol{r}^0_{cg5/0} - \boldsymbol{r}^0_{3/0}\\ \boldsymbol{w}_1 & \boldsymbol{w}_2 & \boldsymbol{w}_3\end{bmatrix} \tag{3.11e}$$

In addition, in order to follow the motion of the crane tip, the crane tip Jacobian is defined as

$$\boldsymbol{J}_6 = \begin{bmatrix}\boldsymbol{w}_1 \times \boldsymbol{r}^1_{tip/0} & \boldsymbol{w}_2 \times (\boldsymbol{r}^0_{tip/0} - \boldsymbol{r}^0_{2/0}) & \boldsymbol{w}_3 \times (\boldsymbol{r}^0_{tip/0} - \boldsymbol{r}^0_{3/0})\\ \boldsymbol{w}_1 & \boldsymbol{w}_2 & \boldsymbol{w}_3\end{bmatrix} \tag{3.12}$$

The next step in the modelling procedure is to define the kinetic and potential energy of the system.

3.1.5 Kinetic and Potential Energy

Now that the mass centre Jacobians (3.11) have been derived, the kinetic and potential energy of the system can expressed. In Section 2.3.3, it was shown that the kinetic energy can be written as a function of the Jacobians and mass matrices. For the knuckleboom crane, consisting of five rigid bodies, the total kinetic energy can be written as

$$T = \sum_{i=1}^{5} \frac{1}{2} \dot{\boldsymbol{q}}^T \boldsymbol{J}_i^T \boldsymbol{M}_i \boldsymbol{J}_i \dot{\boldsymbol{q}} \tag{3.13}$$

where $\dot{\boldsymbol{q}}$ is the generalized velocities, the time derivative of (3.3), $\boldsymbol{J}_i$ is the Jacobian for the ith rigid body and $\boldsymbol{M}_i$ the mass matrix for the ith body. Mass matrix i can be written as

$$\boldsymbol{M_i} = \begin{bmatrix} \boldsymbol{I}_{3\times3} m_i & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{R}_0^i \boldsymbol{I}_i (\boldsymbol{R}_0^i)^T \end{bmatrix} \tag{3.14}$$

where m_i and $\boldsymbol{I}_i$ are the mass and moment of inertia matrix of body i.

The system's potential energy depends upon the position of mass centres in the gravity field and other storage elements. For the crane system, the mass of the actuators and arms will contribute to the total potential energy. The base will not influence the energy since its position is fixed in the direction of gravity. Since there are no storage elements in the system, the total potential energy can be written as

$$V = \sum_{i=1}^{5} -g m_i z_i \tag{3.15}$$

where g is the acceleration of gravity, m_i the mass of body i and z_i its position in the gravity field, given by the z component of the mass center positions expressed (3.9).

With the kinetic and potential energy of the crane system expressed, the generalized inertia, Coriolis and centripetal matrices and gravity vector can be derived. The next section will derive these quantities and put the system on state-space form.

3.1.6 State-Space Formulation

It was shown in Section 2.3 that the generalized inertia, Coriolis and centripetal matrices and gravity vector of a system of rigid bodies can be derived from the system's kinetic and potential energy. These quantities were derived in the previous section, see (3.13) and (3.15).

Hence, the generalized inertia matrix of the crane can be written as

$$\boldsymbol{D} = \sum_{i=1}^{5} \boldsymbol{J}_i^T \boldsymbol{M}_i \boldsymbol{J}_i \tag{3.16}$$

where J_i is one of the Jacobians given by (3.11) and $\boldsymbol{M}_i$ is defined in (3.14).

The Coriolis matrix can then be derived directly from (3.16) by using the Christoffel symbols introduced in Section 2.3.4. For this system, the kjth element of the Coriolis matrix can be written as

$$c_{kj} = \sum_{i=1}^{3} \frac{1}{2} \left(\frac{\partial d_{kj}}{\partial q_j} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k} \right) \dot{q}_i \quad k = 1, ..., 3 \quad j = 1, ...3 \tag{3.17}$$

where d_{kj} is the k, jth element of (3.16) and q_i generalized coordinate i.

The gravity vector of the system can be derived directly from the potential energy (3.15)

$$\boldsymbol{G} = \begin{bmatrix} \frac{\partial V}{\partial q_1} & \frac{\partial V}{\partial q_2} & \frac{\partial V}{\partial q_3} \end{bmatrix}^T \tag{3.18}$$

where V is the potential energy.

Finally, the model can be put on state-space form using the formulation from section 2.3. The state space formulation of the three degrees of freedom knuckleboom crane can be written as

$$\boldsymbol{D}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\dot{\boldsymbol{q}}, \boldsymbol{q})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{\tau} \tag{3.19}$$

where $\boldsymbol{D}$ is given by (3.16), $\boldsymbol{C}$ by (3.17) and $\boldsymbol{G}$ by (3.18). The only term that has not been discussed is external forces represented by $\boldsymbol{\tau}$. This will be the topic of the next section.

3.1.7 Actuator Forces and Load

The previous section established the internal forces acting on the crane and put the crane dynamics on state space form (3.19). However, there are several external forces acting on the crane that has yet to be modelled. For instance, the force produced by the actuators and the load at the crane tip.

It is assumed that the load that is to be lifted by the crane acts at the crane tip and is constant during a single operation. This is of course a simplification since the load in reality acts at the tip of the wire and may also cause lateral disturbances due to pendulum motions. The force exerted on the crane tip by the load can be written as

$$\boldsymbol{F}_L = \begin{bmatrix} 0 & 0 & -m_L g \end{bmatrix}^T \tag{3.20}$$

where m_l is the weight of the cargo. This force must be expressed as a generalized force in order to derive its effect upon the crane dynamics. An introduction to generalized forces can be found in section 2.2.4.

The first step is to find the position where the load acts on the system. The position of the crane tip can be written as

$$\boldsymbol{r}^0_{tip/0} = \boldsymbol{r}^0_{3/0} + \boldsymbol{R}^3_0 \begin{bmatrix} L_2 \\ 0 \\ 0 \end{bmatrix} \tag{3.21}$$

where L_2 is the length of the upper arm. The generalized force from the load is then given by (2.14), and can for this system be written as

$$\boldsymbol{\tau}_L = \begin{bmatrix} \boldsymbol{F}_L^T \frac{\partial \boldsymbol{r}^0_{tip/0}}{\partial \theta_1} & \boldsymbol{F}_L^T \frac{\partial \boldsymbol{r}^0_{tip/0}}{\partial \theta_2} & \boldsymbol{F}_L^T \frac{\partial \boldsymbol{r}^0_{tip/0}}{\partial \theta_3} \end{bmatrix}^T \tag{3.22}$$

where $\boldsymbol{r}^0_{tip/0}$ is the position of the load in the inertial frame given by (3.21). Now, the procedure is repeated for the actuators.

The actuator forces act parallel to each actuator and have to be decomposed in order to be expressed as a generalized force. The decomposition angle for the lower actuator is δ_1, the orientation of reference frame $x_4y_4z_4$ given by (3.1). The second decomposition angle is δ_2, or the orientation of $x_5y_5z_5$ given by (3.2). In addition, the forces will have to be transformed to the inertial frame $x_0y_0z_0$ using the rotation θ_1. The forces produced by each actuator can be written as

$$\boldsymbol{F}_1 = F_{1,a} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \qquad \boldsymbol{F}_2 = F_{2,a} \boldsymbol{R}_0^1 \begin{bmatrix} \cos(\delta_1) \\ 0 \\ \sin(\delta_1) \end{bmatrix} \qquad \boldsymbol{F}_3 = F_{3,a} \boldsymbol{R}_0^1 \begin{bmatrix} \cos(\delta_2) \\ 0 \\ \sin(\delta_2) \end{bmatrix} \tag{3.23}$$

where $F_{i,a}$ is the amplitude of the force for the ith actuator.

The force from an actuator will result in an equal force at both the upper and lower connection (Gyberg, 2017). Therefore, the position of both connections points must defined. The connection points for the lower actuator are

$$\boldsymbol{r}^0_{a11/0} = \boldsymbol{R}_1^0 \begin{bmatrix} a & 0 & 0 \end{bmatrix}^T \qquad \boldsymbol{r}^0_{a12/0} = \boldsymbol{r}^0_{2/0} + \boldsymbol{R}_0^2 \begin{bmatrix} b & 0 & -u \end{bmatrix}^T \tag{3.24}$$

and

$$\boldsymbol{r}^0_{a21/0} = \boldsymbol{r}^0_{2/0} + \boldsymbol{R}_2^0 \begin{bmatrix} w & 0 & -r \end{bmatrix}^T \qquad \boldsymbol{r}^0_{a22/0} = \boldsymbol{r}^0_{3/0} + \boldsymbol{R}_0^3 \begin{bmatrix} c & 0 & -s \end{bmatrix}^T \tag{3.25}$$

for the upper actuator.

Now, the generalized force from each actuator can be determined from (2.14). By combining the actuator contributions into a matrix, the generalized actuators forces can be written as

$$\boldsymbol{\tau}_a = \begin{bmatrix} 1 & \boldsymbol{F}_2^T(-\frac{\partial \boldsymbol{r}_{a11}}{\partial \theta_1} + \frac{\partial \boldsymbol{r}_{a12}}{\partial \theta_1}) & \boldsymbol{F}_3^T(-\frac{\partial \boldsymbol{r}_{a21}}{\partial \theta_1} + \frac{\partial \boldsymbol{r}_{a22}}{\partial \theta_1}) \\ 0 & \boldsymbol{F}_2^T(-\frac{\partial \boldsymbol{r}_{a11}}{\partial \theta_2} + \frac{\partial \boldsymbol{r}_{a12}}{\partial \theta_2}) & \boldsymbol{F}_3^T(-\frac{\partial \boldsymbol{r}_{a21}}{\partial \theta_2} + \frac{\partial \boldsymbol{r}_{a22}}{\partial \theta_2}) \\ 0 & \boldsymbol{F}_2^T(-\frac{\partial \boldsymbol{r}_{a11}}{\partial \theta_3} + \frac{\partial \boldsymbol{r}_{a12}}{\partial \theta_3}) & \boldsymbol{F}_3^T(-\frac{\partial \boldsymbol{r}_{a21}}{\partial \theta_3} + \frac{\partial \boldsymbol{r}_{a22}}{\partial \theta_3}) \end{bmatrix} \tag{3.26}$$

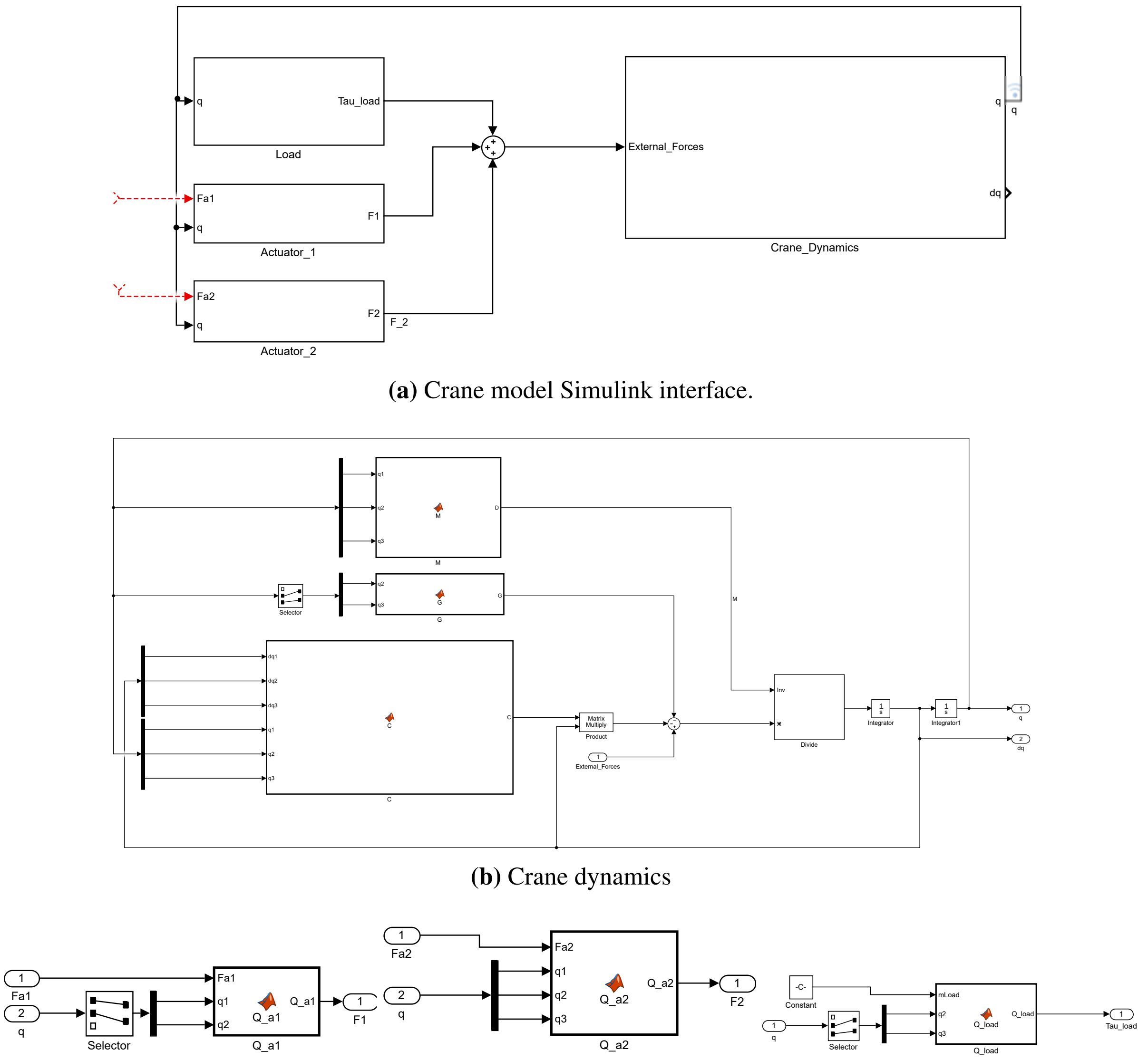
and by extracting the amplitude of each actuator force, (3.26) can be rewritten as

$$\boldsymbol{\tau_a} = \boldsymbol{Q} \begin{bmatrix} F_{a1} \\ F_{a2} \\ F_{a3} \end{bmatrix} = \boldsymbol{Q}\boldsymbol{u} \tag{3.27}$$

In summary, a model of the knuckleboom crane presented in Section 1.1 has been derived and the model is now ready for implementation in the Simulink.

3.1.8 Simulink Implementation

Deriving the system's generalized inertia and Coriolis matrices by hand is both time consuming and prone to errors. Therefore, the symbolic Math Toolbox(Mathworks, 2017) has been used to derive them and the script for generating the model can be found in Appendix A.1. Figure 3.3 shows the crane model (3.19) implemented in Simulink.



(a) Crane model Simulink interface.

(b) Crane dynamics

(c) Lower actuator connection.

(d) Upper actuator connection.

(e) Load connection

Figure 3.3: Overview of the crane model implemented in Simulink.

Figure 3.3a shows the interface of the simulation model, which has been divided into the crane dynamics, actuator connections and load. The outputs of the model are the generalized coordinates and velocities. The actuator connections have been configured to take the force that should be produced by the actuator as input. Meaning that different control laws can easily be connected to the simulation model.

A detailed overview of the crane dynamics are shown in Figure 3.3b, showing that the inertia, Coriolis and gravity vectors have been implemented as custom Matlab functions. An input port for external forces have also been provided.

The actuator and load connections are shown in Figure 3.3c, 3.3d and 3.3e. They have been implemented as Matlab functions containing the matrices that were developed in Section 3.1.7.

Now that the crane model has been implemented in Simulink, a simulation can be run to verify its behaviour.

3.1.9 Simulation Results

The final step in the modelling procedure is to run a simple simulation showing that the crane model behaves as expected. The simulation will be run without any actuator forces and without limits on the joint orientations. In this case, only gravitational forces acts on the system and it is expected that the response of the lower and upper arm will be similar to a gravity pendulum.

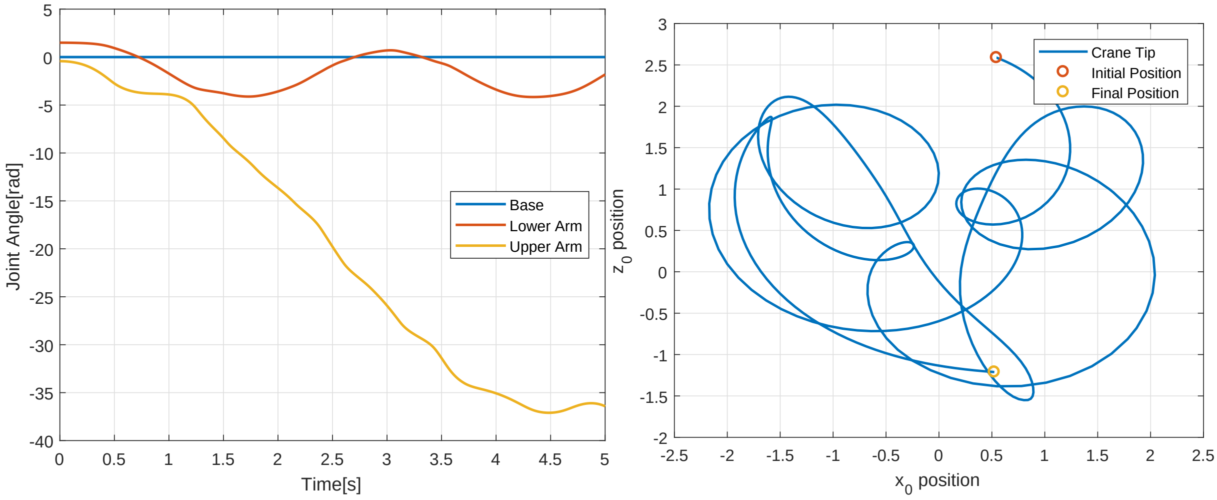
The simulation will be run with the mass and geometric parameters presented in Tables 3.1 and 3.2, which corresponds to the design parameters from (Gyberg, 2017).

Table 3.1: Mass Parameters

Parameter	Value	Parameter	Value	Parameter	Value
m_1	67 kg	$I_{1,z}$	1.8 kgm^2	$I_{4,z}$	2.3 kgm^2
m_2	23.3 kg	$I_{2,y}$	9.2 kgm^2	$I_{5,y}$	1.1 kgm^2
m_3	5.5 kg	$I_{2,x}$	9.4 kgm^2	$I_{5,z}$	1.1 kgm^2
m_4	8.5 kg	$I_{3,y}$	1.6 kgm^2		
m_5	6.8 kg	$I_{3,x}$	1.5 kgm^2		
m_l	16 kg	$I_{4,y}$	2.3 kgm^2		

Table 3.2: Geometrical Parameters

Parameter	Value	Parameter	Value
h	0.495 m	u	0.14m
L_1	1.251 m	r	0.14m
L_2	0.96 m	s	0.118m
a	0.1 m	k	0.533 m
b	0.461 m	n	0.42
c	0.1 m	o	0.324 m
w	0.461m	p	0.44 m



(a) Joint orientation.

(b) Crane tip position in the inertial frame.

Figure 3.4: Results from simulating the knuckleboom crane model without actuator forces and limits imposed on the joint orientations.

The results from the unforced simulation, Figure 3.4, shows that the crane behaved as an undamped gravity pendulum. The base did not move at all, since it was unaffected by gravity and no external forces were applied. The lower arm was affected only by gravity and the motion of the upper arm. Therefore, it behaved as a simple pendulum moving between its initial position and maximum stroke. The upper arm was driven by gravity and the motion of the lower arm.

It can be seen from the angle, Figure 3.4a, and pendulum movement, Figure 3.4b, of the upper arm that it started spinning In the real setup the angle of the upper arm and elongation of the actuators are restricted and will not be allowed to crash into or pass through the lower arm. Currently, this is a weakness of the simulation and limits to the angle can adding a stiff spring and a strong damper as a function of the orientation of the respective arm, added as external forces, or by imposing limitations in the Simulink model.

However, the most important observation is that the kinematics of the model behave as expected when the system is unaffected by external forces. This can be observed from the indefinite pendulum motion of the lower arm and the double pendulum motion of the tip. Thus, the most basic response of the crane dynamics have been shown to behave as expected.

As a summary, in this section the equations of motion for the knuckleboom crane from Section 1.1 crane lab has been derived by using Lagrangian mechanics. The resulting model has been put on state-space form and implemented in Simulink. A simulation of the unforced system has shown that the dynamics behaves as a double gravity pendulum when only affected by gravity.

In the next section, a model of the crane's winch system will be developed.

3.2 Winch Model

In the previous section, the dynamics of the knuckleboom crane from Section 1.1 was derived using Lagrangian mechanics. However, the model did not include the winch and simply modelled the load as a constant weight hanging from the crane tip. In this chapter, the dynamics of the winch will be derived such that the length of wire between the crane tip and cargo can be described. The motivation is to use the model to develop a control law for softly landing the load.

3.2.1 System Description

The section will start by giving a description of the winch system that is to be modelled. The main components of the winch are the reel, on which wire is stored, the electric motor controlling the speed of the reel and the belt connecting the motor and reel. An overview of the system is shown in Figure 3.5.

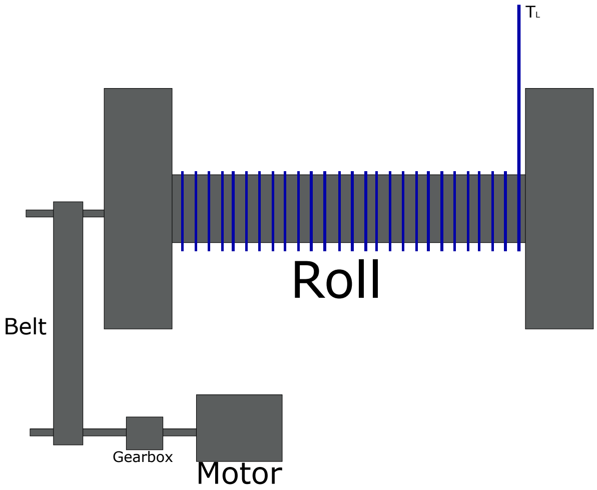


Figure 3.5: Figure showing the wire roll, the wire load, electrical motor and transmission belt.

In essence, the winch system is not different from an electric motor. The dynamics of the electric side will be equal to that of an electric DC motor, the challenge lies in accurately modelling the mechanical side of the winch. As the amount of wire on the reel changes, so will its inertia and diameter. This means that the moment acting on the reel from the load will also change with the amount of wire on the reel.

3.2.2 Winch Dynamics

Since the amount of wire stored on the reel is to be modelled, the system states are chosen as

$$\boldsymbol{x} = \begin{bmatrix} L \\ \dot{L} \\ i \end{bmatrix} \qquad \dot{\boldsymbol{x}} = \begin{bmatrix} \dot{L} \\ \ddot{L} \\ \dot{i} \end{bmatrix} \qquad u = V \tag{3.28}$$

where L is the amount of wire on the reel, $\dot{L}$ the velocity of wire leaving or entering the reel, i the current drawn by the electric motor and V the voltage across the motor.

The wire velocity is a function of the reel radius and angular velocity

$$\dot{L} = \frac{D(L)}{2}\omega \tag{3.29}$$

where $D(L)$ is the diameter of the reel, assumed to be a function of the amount of wire on the reel, and ω, the reel velocity. The acceleration of the wire can be found by differentiating (3.29) with respect to time. Since the reel diameter is dependent upon L its derivative will also appear in the expression. The wire acceleration can be written as

$$\ddot{L} = \frac{D^L(L)}{2}\dot{L}\omega + \frac{D(L)}{2}\dot{\omega} \tag{3.30}$$

where $D^L(L)$ is the derivative of $D(L)$ with respect to L and ω is the acceleration of the reel. An expression for the reel velocity has already been defined in (3.29) and inserting it into (3.30) yields

$$\ddot{L} = \frac{D^L(L)}{D(L)}\dot{L}^2 + \frac{D(L)}{2}\dot{\omega} \tag{3.31}$$

The next step is to find an expression for $\dot{\omega}$. The reel acceleration can be modelled by Newton's second equation and can be written as

$$\dot{\omega}I(L) = -\omega B + K_t i + \tau_L \tag{3.32}$$

where $I(L)$ is the inertia of the reel, assumed to be a function of L, B the reel damping, K_t the torque constant of the electric motor and τ_L the load from the cargo.

The torque produced by the electric motor acts at the motor's shaft and must be transformed through the gearbox and belt connection to find its magnitude at the reel. This will be modelled by multiplying K_t with an efficiency. It will be assumed that the friction acting on the reel is small, meaning that $B = 0$.

The load acting on the crane tip is transferred to the reel using a series of pulleys, see Section 1.1, and the moment it exerts on the reel depends upon the reel radius. The magnitude of the force at the reel can be modelled using the Capstan equation.

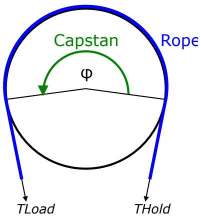


Figure 3.6: Pulley showing the area covered by the rope, the tension exerted by the load and reaction tension at the hold side. From (Commons, 2017).

Figure 3.6 shows a single pulley covered by a rope spanning the angle ϕ. The relationship between the force at the load and hold side is given by

$$T_{load} = T_{hold} e^{\mu \phi} \tag{3.33}$$

where μ is the coefficient of friction between the wire and pulley. The crane supports the rope using three pulleys, meaning that the magnitude of the load at the reel is given by

$$\tau_L = \frac{D(L)}{2} \frac{m_L g}{e^{\mu(\phi_1+\phi_2+\phi_3)}} \tag{3.34}$$

where m_L is the weight of the load, ϕ_1, ϕ_2 and ϕ_3 the spanning angle of each pulley, and g the gravitational acceleration.

By inserting (3.34) into (3.32), the final expression of the reel acceleration is obtained

$$\dot{\omega} = \frac{K_t \eta_t}{I(L)} i + \frac{D(L)}{2I(L)} \frac{m_L g}{e^{\mu(\phi_1+\phi_2+\phi_3)}} \tag{3.35}$$

The final equation for the wire acceleration is found by inserting (3.35) into 3.31, yielding

$$\ddot{L} = \frac{D^L(L)}{D(L)} \dot{L}^2 + \frac{D(L)}{2I(L)} K_t \eta_t i + \frac{D(L)^2}{4I(L)} \frac{m_L}{e^{\mu(\phi_1+\phi_2+\phi_3)}} g \tag{3.36}$$

The final step is to model the electric part of the motor. It is assumed that the DC brushless motor controlling the winch can be modelled by using the simple DC motor model from (Balchen et al., 2003). This model reduced the electric part of the motor to a simple RI circuit, meaning that the current dynamics can be written as

$$\dot{i} = -\frac{K_v}{L_a} \omega_m - \frac{R_a}{L_a} i + \frac{V}{L_a} \tag{3.37}$$

where K_v is the motor's speed constant, ω_m the shaft speed, R_a the resistance and L_a the anchor inductance. Again, the angular velocity in (3.37) is given at the motor shaft. However, ω is not a state in the system and must be transformed to the wire velocity $\dot{L}$. The transformation from shaft to reel velocity is given by the motor gearbox and belt transmission and will, as with the motor torque, be modelled by an efficiency. Hence, ω_m can be written as

$$\omega_m = \eta_\omega \frac{D(L)}{2} \dot{L} \tag{3.38}$$

where η_ω is the efficiency. The final equation for the current dynamics is obtained by inserting (3.38) into (3.37)

$$\dot{i} = -\eta_\omega \frac{K_v}{L_a}\frac{D(L)}{2}\dot{L} - \frac{R_a}{L_a}i + \frac{V}{L_a} \tag{3.39}$$

There are now only two terms that must be discussed before the final winch system can be presented: the inertia and diameter of the reel. As mentioned both of these parameters will be functions of the amount of wire and the reel. This thesis will use the equations developed by (Skjong. and Pedersen, 2014) as part of modelling a hydraulic winch system. According to this study the diameter of the reel is given by

$$D(L) = \sqrt{\frac{LD_w^2}{w_r f_w} + D_{r,0}^2} \tag{3.40}$$

where D_w is the diameter of the wire, w_r the width of the reel, $D_{r,0}$ the diameter of an empty reel and f_w a packing efficiency. The packing efficiency depends upon how the wire is stored at the reel. According to (Skjong. and Pedersen, 2014) the most efficient packing efficiency is $f_w = 0.9069$.

Next is the inertia of the reel. The inertia is given as a sum of the static elements of the reel, such as the flange and reel, and the amount of wire stored on the reel. According to (Skjong. and Pedersen, 2014) the inertia the winch can be written as

$$I(L) = \frac{1}{2}m_{reel}\frac{D_{r,0}^2}{4} + m_{flange}\frac{D_{r,full}^2}{4} + \frac{1}{2}\rho_w L\left(\frac{D(L)^2 - D_{r,0}^2}{4}\right) \tag{3.41}$$

where m_{reel} is the mass of the reel, m_{flange} of the flanges, ρ_w the density of the wire and $D_{r,full}$ the diameter of a full reel.

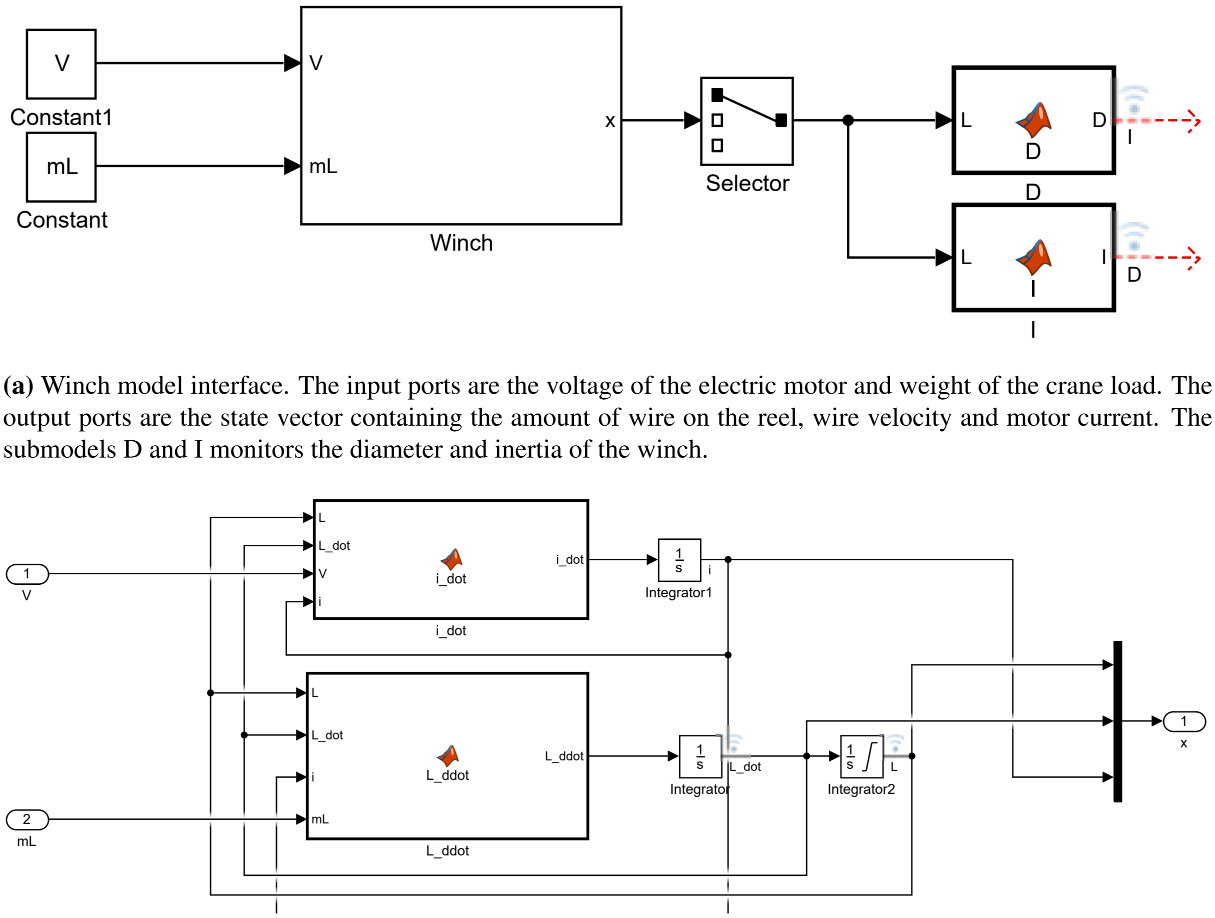
Now that the winch model has been fully developed the final system can be presented.

$$\begin{bmatrix} \dot{L} \\ \ddot{L} \\ \dot{i} \end{bmatrix} = \begin{bmatrix} \dot{L} \\ \frac{D^L(L)}{D(L)}\dot{L}^2 + \frac{D(L)}{2I(L)}K_t\eta_t i \\ -\eta_\omega \frac{K_v}{L_a}\frac{D(L)}{2}\dot{L} - \frac{R_a}{L_a}i \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L_a} \end{bmatrix} V + \begin{bmatrix} 0 \\ -\frac{D(L)^2}{4I(L)}\frac{m_L}{e^{\mu(\phi_1+\phi_2+\phi_3)}} \\ 0 \end{bmatrix} g \tag{3.42}$$

In summary, a dynamical model of the winch system shown in Figure 3.5 has been developed. The model has been designed to account for the changes in reel diameter and inertia, caused by wire entering or leaving the reel. The resulting model is presented in 3.42 and is non-linear due to the inertia and reel diameter. In the next section the model will be implemented in Simulink.

3.2.3 Simulink Implementation

The winch dynamics that were developed in the previous section, (3.42), has been implemented in Simulink and made ready for simulation. The system interface and dynamics are presented in Figure 3.7 and the Matlab code for generating the model can be found in Appendix A.3.



(a) Winch model interface. The input ports are the voltage of the electric motor and weight of the crane load. The output ports are the state vector containing the amount of wire on the reel, wire velocity and motor current. The submodels D and I monitors the diameter and inertia of the winch.

(b) Winch model dynamics. The amount of wire stored on the reel has been limited between zero and maximum wire length.

Figure 3.7: Winch model interface and dynamics as implemented in Simulink.

The model interface is shown in Figure 3.7a and has been implemented with the motor voltage and crane load as inputs. Meaning that different control laws and load models can be easily connected to the system. The outputs of the model is the state vector defined by (3.28). In addition, the equations calculating reel inertia and diameter, (3.41) and (3.40), have been implemented as Matlab functions in order to monitor these quantities.

The interior of the winch dynamics subsystem is shown in Figure 3.7b. The current and wire length dynamics have been implemented as Matlab functions. The amount of wire is calculated by integrating the wire velocity. Initial conditions for the wire and limitations on the amount of wire on the reel have been implemented in the integrator block.

Now that the winch model has been derived and implemented in Simulink, test simulation can be run.

3.2.4 Simulation Results

The objective of the simulation is to verify that the winch dynamics behaves as expected and that the reel and inertia equations have been correctly implemented. The simulation will be run with the motor parameters given in Table 3.3 and reel properties in Table 3.4. The transmission and wire properties are given in Tables 3.5 and 3.6. They are based on design parameters from (Gyberg, 2017). The simulation will be run with a load of 16kg attached to the crane.

Table 3.3: Winch motor parameters

Parameter	Symbol	Value	Unit
Torque constant	K_t	53.4	$\mathrm{mN\,m\,A^{-1}}$
Speed constant	K_v	179	$\mathrm{RPM\ V^{-1}}$
Anchor resistance	R_a	0.307	Ω
Anchor inductance	L_a	0.118	mH

Table 3.4: Reel properties

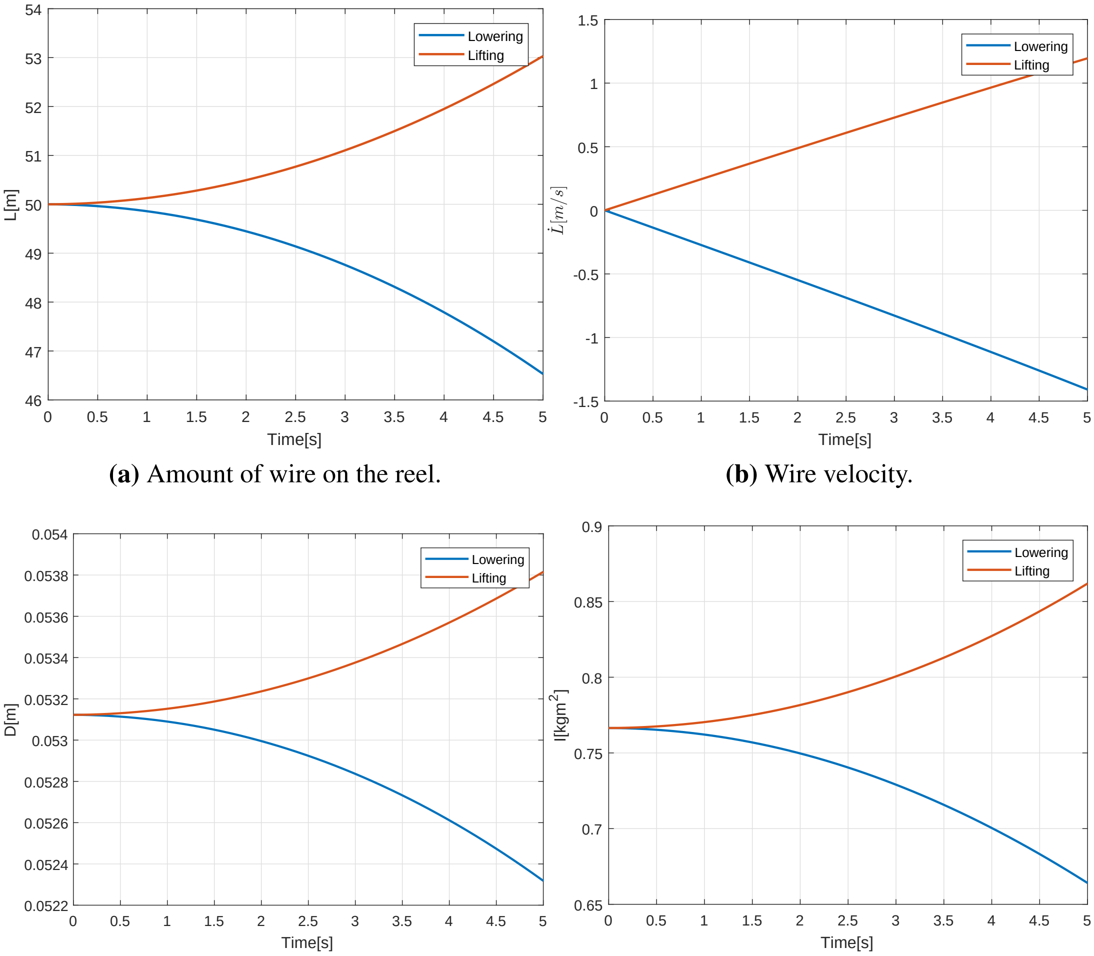
Parameter	Symbol	Value	Unit
Reel mass	m_{reel}	1	kg
Flange mass	m_{flange}	1	kg
Diameter of empty reel	$D_{r,0}$	0.04	m
Diameter of full reel	$D_{r,full}$	0.1	m
Reel width	w_r	0.13	m
Packing factor	f_r	0.9069	-

Table 3.5: Transmission properties

Parameter	Symbol	Value	Unit
Gearbox reduction	η_g	43	-
Belt diameter at reel	$D_{r,pw}$	0.12	m
Belt diameter at gearbox	$D_{g,pw}$	0.6	m
Wire pulley span	ϕ	$\frac{\pi}{4}$	rad

Table 3.6: Wire properties

Parameter	Symbol	Value	Unit
Total wire length	L_t	100	m
Wire diameter	D_w	1.7	mm
Wire density	ρ_w	100	$\mathrm{kg/m^3}$



(a) Amount of wire on the reel.

(b) Wire velocity.

(c) Reel diameter.

(d) Reel inertia

Figure 3.8: Results of lifting and lowering with the winch using a supply voltage of ±5V. A load of 16kg was attached to the wire.

The wire velocity was slightly higher when lowering, Figure 3.8b, since the gravitational forces on the load assists motions in this direction. The velocity did not saturate since friction has not been included in the motor model. The inertia and diameter of the reel, Figures 3.8c and 3.8d, showed the expected behaviour. They increased when wire entered the reel and decreased when lowering the load.

The uncertainty of the model lies in the values that has been chosen for the pulley span, and the fact that friction has been neglected. The actually pulley span is expected to change as the orientation of the crane changes, but can be assumed constant when the crane arms are not moving. However, whether the chosen values represents the crane in Section 1.1 is not known, and a parameter study should be done to determine them.

In summary, this section has developed a dynamic model of the winch of the knuckleboom crane. A simulation has been run to verify that the model behaves similarly to an electric DC motor and that the function for calculating inertia and diameter have been implemented correctly. The uncertainty of the model lies in the selection of parameters.

Chapter 4

Control Laws

The previous chapter developed models of the knuckleboom crane from Section 1.1 and its winch system. The limitations and simplifications of the models have been discussed and simulations have been run to verify their theoretical behaviour. In this chapter, the knuckleboom crane model will be used to design a control law for positioning the joints of the crane and the winch model to design a control law for soft landing.

4.1 Crane Position Control Law

The motivation for developing a position control law for the crane is to implement it in the crane lab from Section 1.1 and investigate the response of the crane when subject to model based control laws. Currently, a hands-on control system, using low-level motor controllers, are implemented, making it possible to control the crane manually using two joysticks. The low-level controllers, however, do not utilize feedback from the sensors in the lab and only controls the velocity of the crane joints. Hence, positioning the crane at a desired orientation must be done by hand and following a desired reference signal is difficult and inaccurate, if not impossible. Designing a control law for positioning the crane joints will therefore improve the state of the lab by enabling the user to position the crane at a desired position and to follow reference signals.

The modelling procedure will start by defining the system that is to be stabilized and the control objective. The control law itself will be designed using the recursive backstepping technique, see Section 2.4.3. After the control law has been designed it will be implemented in Simulink and tested and tuned with the simulation model from Section 3.1.

4.1.1 Problem Formulation

The system that is to be stabilized is the knuckleboom crane model from Section 3.1. The states of the model are the joint orientations and velocities and the input is the force produced by each actuator. It is assumed that full state-feedback is available, meaning that both the joint angles and velocities can be measured. Hence, the model states and inputs can be written as

$$\boldsymbol{x}_1 = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} \qquad \boldsymbol{x}_2 = \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} \qquad \boldsymbol{u} = \begin{bmatrix} \boldsymbol{F}_1 \\ \boldsymbol{F}_2 \\ \boldsymbol{F}_3 \end{bmatrix} \tag{4.1}$$

where $\boldsymbol{x}_1$ is the joint angles, $\boldsymbol{x}_2$ the joint velocities and $\boldsymbol{F}_1$, $\boldsymbol{F}_2$,$\boldsymbol{F}_3$ the forces produced by the actuators. The system dynamics were derived in Section 3.1 and can be written as

$$\dot{\boldsymbol{x}}_1 = \boldsymbol{x}_2 \tag{4.2a}$$

$$\boldsymbol{M}(\boldsymbol{x}_1)\dot{\boldsymbol{x}}_2 = -\boldsymbol{C}(\boldsymbol{x}_1, \boldsymbol{x}_2)\boldsymbol{x}_2 - (\boldsymbol{G}(\boldsymbol{x}_1) + \boldsymbol{\tau}_L)g + \boldsymbol{Q}_u\boldsymbol{u} \tag{4.2b}$$

with the states defined in (4.1). From (4.2) it can be noted that the system is subject to a constant gravitational disturbance from the crane arms and load. Hence, the control law be should be designed to include integral to remove stationary deviations.

The goal of the controller is to make joint orientations converge to a desired reference. Meaning that the control objective can be stated as

$$\lim_{t \to \infty} |\boldsymbol{x}_1 - \boldsymbol{x}_d| = 0 \tag{4.3}$$

where $\boldsymbol{x}_1 \in \mathbb{R}^3$ contains the three joint angles of the crane. In order to achieve this objective a controller with integral action will be designed by following the procedure of (Skjetne and Fossen, 2004). This design method will ensure an asymptotically stable controller that is able to reject the gravitational disturbance.

4.1.2 Control Law Design

Now that the system that is to be stabilized has been formulated and the control objective defined, the control law can be designed. The recursive backstepping technique will be used to design the control law, see Section 2.4.3 for more details. The method is used to design asymptotically stable controllers for systems of strict-feedback form. Comparing (4.1) with (2.31) shows that the system is on strict-feedback form. Since (4.1) is of relative degree two, the design procedure will consist of two steps.

Before starting the design, a new set of states are defined as

$$\boldsymbol{z}_1 = \boldsymbol{x}_1 - \boldsymbol{x}_d \tag{4.4a}$$

$$\boldsymbol{z}_2 = \boldsymbol{x}_2 - \boldsymbol{\alpha}_1 \tag{4.4b}$$

where $\boldsymbol{x}_d$ is the reference signal and $\boldsymbol{\alpha}_1$ is a virtual controller that will be used to stabilize $\boldsymbol{z}_1$. The dynamics of the new states are found by differentiating (4.4) with respect to time and inserting (4.1). The new dynamics can be written as

$$\dot{\boldsymbol{z}}_1 = \boldsymbol{x}_2 \tag{4.5a}$$

$$\dot{\boldsymbol{z}}_2 = \boldsymbol{M}^{-1}[\boldsymbol{C}\boldsymbol{x}_2 - (\boldsymbol{G} + \boldsymbol{\tau}_L)g + \boldsymbol{Q}_u\boldsymbol{u}] - \dot{\boldsymbol{\alpha}}_1 \tag{4.5b}$$

where it has been assumed that $\boldsymbol{x}_d$ is constant.

Step 1:
In the first step, a Lyapunov function of $\boldsymbol{z}_1$ is written as

$$V_1 = \frac{1}{2}\boldsymbol{z}_1^T\boldsymbol{P}_1\boldsymbol{z}_1 \tag{4.6}$$

$$\dot{V}_1 = \boldsymbol{z}_1^T\boldsymbol{P}_1(\boldsymbol{z}_2 + \boldsymbol{\alpha}_1) \tag{4.7}$$

and an $\boldsymbol{\alpha}_1$ is chosen such that (4.7) is rendered negative definite. However, terms including $\boldsymbol{z}_2$ are ignored and carried over to the next step. The virtual controller is chosen as

$$\boldsymbol{\alpha}_1 = \boldsymbol{A}_1\boldsymbol{z}_1 \tag{4.8}$$

$$\dot{\boldsymbol{\alpha}}_1 = \boldsymbol{A}_1\boldsymbol{x}_2 \tag{4.9}$$

where $\boldsymbol{A}_1$ and $\boldsymbol{P}_1$ satisfies Lyapunov's equation (2.30). The time derivative of $\boldsymbol{\alpha}_1$ will be needed in the next step. Inserting (4.8) into (4.7) yields

$$\dot{V}_1 = -\boldsymbol{z}_1^T\boldsymbol{Q}_1\boldsymbol{z}_1 + \boldsymbol{z}_1^T\boldsymbol{P}_1\boldsymbol{z}_2 \tag{4.10}$$

showing that $\dot{V}_1$ is negative definite with respect to $\boldsymbol{z}_1$. The objective of the first step has been reached and the second step can be initiated.

Step 2:
The second step Lyapunov function is chosen to be

$$V_2 = V_1 + \frac{1}{2}\boldsymbol{z}_2\boldsymbol{P}_2\boldsymbol{z}_2 \tag{4.11}$$

and its time derivative can be written as

$$\dot{V}_2 = -\boldsymbol{z}_1^T\boldsymbol{Q}_1\boldsymbol{z}_1 + \boldsymbol{z}_1^T\boldsymbol{P}_1\boldsymbol{z}_2 + \boldsymbol{z}_2^T\boldsymbol{P}_2(\boldsymbol{M}^{-1}(-\boldsymbol{C}\boldsymbol{x}_2 - \boldsymbol{G} + \boldsymbol{Q}_u\boldsymbol{u}) - \dot{\boldsymbol{\alpha}}_1) \tag{4.12a}$$

$$\dot{V}_2 = -\boldsymbol{z}_1^T\boldsymbol{Q}_1\boldsymbol{z}_1 + \boldsymbol{z}_2^T\boldsymbol{P}_2(\boldsymbol{P}_2^{-1}\boldsymbol{P}_1\boldsymbol{z}_1 + \boldsymbol{M}^{-1}(-\boldsymbol{C}\boldsymbol{z}_2 - \boldsymbol{C}\boldsymbol{\alpha}_1 - \boldsymbol{G} + \boldsymbol{Q}_u\boldsymbol{u}) - \dot{\boldsymbol{\alpha}}_1) \tag{4.12b}$$

Since $\boldsymbol{u}$ is included in (4.12), the control input of the system has been reached and can be defined. Again, the goal is to render (4.12) negative definite. However, the gravitational terms will be ignored for now and included later in the integral state. Hence, $\boldsymbol{u}$ can be written as

$$
\begin{aligned}
\boldsymbol{u} &= \boldsymbol{Q}^{-1}(\boldsymbol{C}\boldsymbol{\alpha}_1 + \boldsymbol{C}\boldsymbol{z}_2 + \boldsymbol{M}(\dot{\boldsymbol{\alpha}}_\mathbf{1} - \boldsymbol{P}_2^{-1}\boldsymbol{P}_1\boldsymbol{z}_1 + \boldsymbol{A}_2\boldsymbol{z}_2)) && \text{(4.13a)} \\
\boldsymbol{u} &= \boldsymbol{Q}^{-1}\boldsymbol{\psi} && \text{(4.13b)}
\end{aligned}
$$

where $\boldsymbol{A}_2$ is a matrix that together with $\boldsymbol{P}_2$ satisfies Lyapunov's equation (2.30).

$$
\dot{V}_2 = -\boldsymbol{z}_1^T\boldsymbol{Q}_1\boldsymbol{z}_1 - \boldsymbol{z}_2^T\boldsymbol{Q}_2\boldsymbol{z}_2 \tag{4.14}
$$

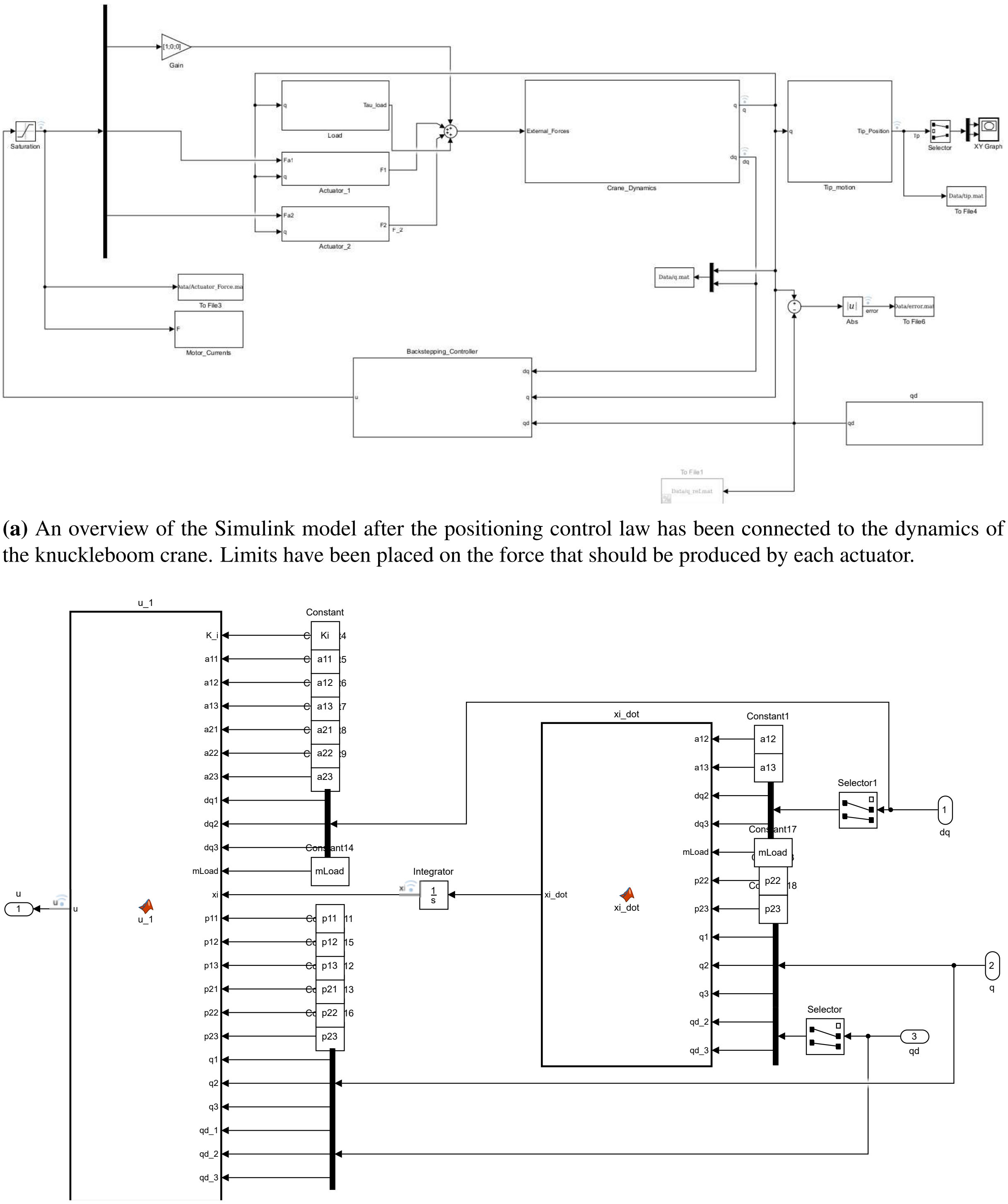
(4.13) is called the nominal controller and is a control law that would stabilize the system without disturbances. (Skjetne and Fossen, 2004) showed that the nominal controller can be used to design an integral state rendering the system asymptotically stable with the disturbances included. For this system, the final control law and integral states can be written as

$$
\begin{aligned}
\dot{\boldsymbol{\xi}} &= \boldsymbol{M}^{-1}(\boldsymbol{Q}_u - \boldsymbol{G})^T(\boldsymbol{z}_2^T\boldsymbol{P}_2)^T && \text{(4.15)} \\
\boldsymbol{u} &= \boldsymbol{Q}^{-1}(\boldsymbol{\psi} - \boldsymbol{M}^{-1}(\boldsymbol{Q}_u - \boldsymbol{G})\boldsymbol{K}_i\boldsymbol{\xi}) && \text{(4.16)}
\end{aligned}
$$

The controller is tuned by choosing values for $\boldsymbol{K}_i$, $\boldsymbol{Q}_1$, $\boldsymbol{Q}_2$, $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ and solving Lyapunov's equation for $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$. The control law has now been designed and can be implemented and tested with the simulation model derived in section 3.1

4.1.3 Simulink Implementation

The control law for positioning the crane joints has been implemented in Simulink and connected to the dynamics developed in Section 3.1. The results are shown in Figure 4.1 and the Matlab script that was used to generate the model can be found in Appendix A.1.



(a) An overview of the Simulink model after the positioning control law has been connected to the dynamics of the knuckleboom crane. Limits have been placed on the force that should be produced by each actuator.

(b) Interior of the Control Law subsystem in Simulink. The nominal control law and integral action has been implemented as seperate matlab functions. The integral action outputs the derivative of ξ which is integrated and passed to the control law.

Figure 4.1: Overview of the Simulink implementation of the position control law.

4.1.4 Simulation Results

The control law for positioning the crane joints has now been designed, implemented in Simulink and connected to the crane dynamics developed in Section 3.1. The next step is to run a series of simulations that will evaluate the controller's performance in different conditions. The tests will include lifting, hoisting and a heave compensation like test where the crane will be made to follow a sine shaped reference signal.

The control law has been tuned in preparation for the simulations and the tuning parameters are presented in Table 4.1. The mass and geometric parameters are the same as those in Section 3.1.9 and are given in Table 3.1 and 3.2. In addition, the limits given in Table 4.3 have been placed on the joint angles.

The efficiencies transforming the force that is produced by the actuators to torque produced by the electric motors have been calculated based on the design parameters of the actuators, found in (Gyberg, 2017), and the parameters of the motors in the crane lab. The results are given in Table 4.2. The efficiencies will be used to estimate the current that must be drawn by each motor to produce the force given by the control law.

Table 4.1: Controller Parameters

Parameter	Value	Parameter	Value
P_1	$\begin{bmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.01 \end{bmatrix}$	P_2	$\begin{bmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.01 \end{bmatrix}$
Q_1	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	Q_2	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
A_1	$\begin{bmatrix} -50 & 0 & 0 \\ 0 & -50 & 0 \\ 0 & 0 & -50 \end{bmatrix}$	A_2	$\begin{bmatrix} -50 & 0 & 0 \\ 0 & -50 & 0 \\ 0 & 0 & -50 \end{bmatrix}$
K_i	400		

Table 4.2: Actuator Efficiency Parameters

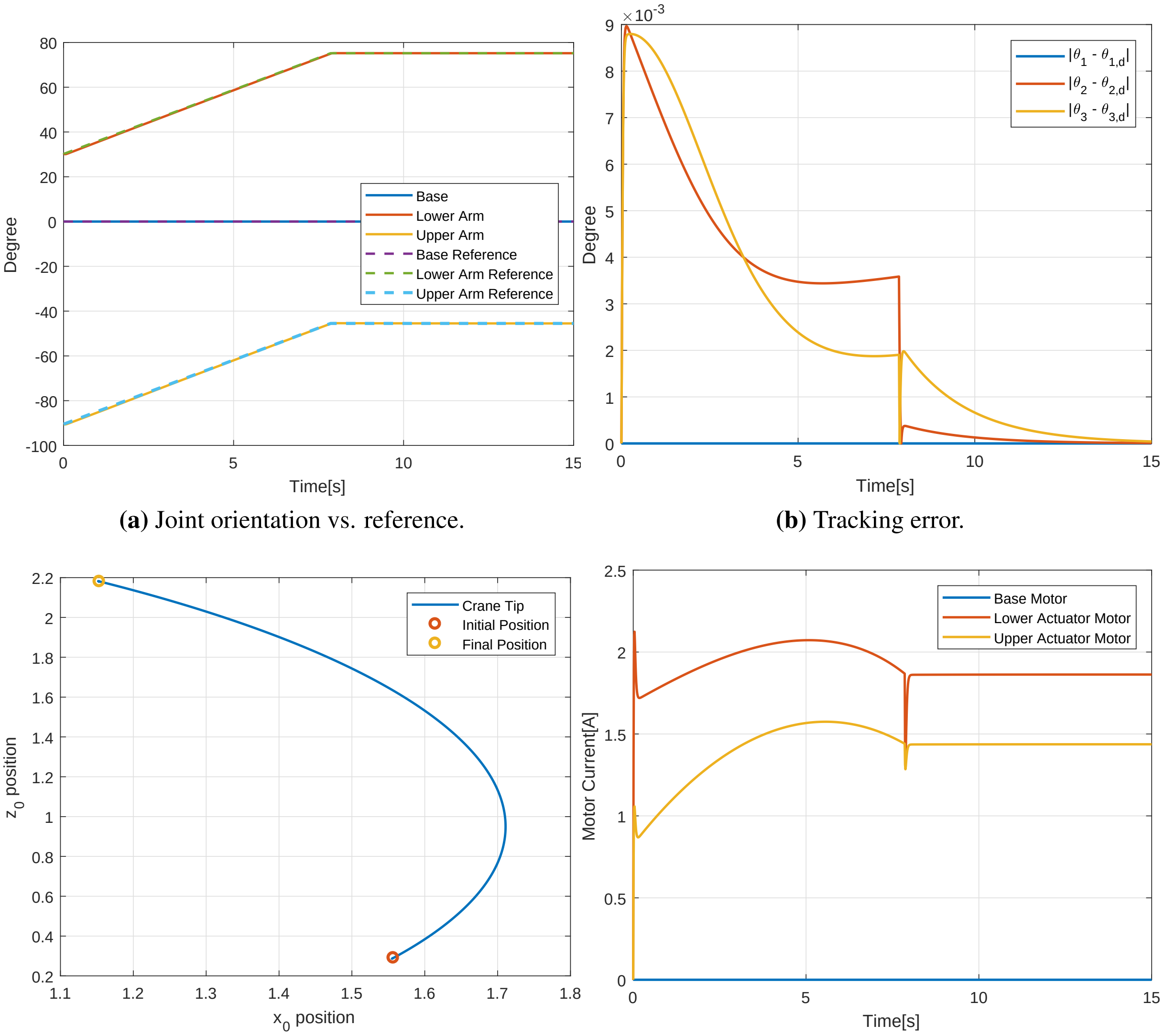
Parameter	Base Actuator	Lower Actuator	Upper Actuator
η	0.011 T /N	$7.3751e^{-5}$ T /N	$1.2214e^{-4}$ T /N
K_t	60.4 mN A^{-1}	60.4 mN A^{-1}	60.4 mN A^{-1}

Table 4.3: Joint limits.

Joint	Lower Limit	Upper Limit
θ_1	-	-
θ_2	$30.2°$	$86°$
θ_3	$-90.5°$	$-24.4°$

Lifting

The first simulation tested the controller's capability of lifting the lower and upper arm from their lower to upper limits. The reference signal was a ramp moving from the lower to upper position in seven seconds. A load of 16kg was attached to the tip of the crane.



(a) Joint orientation vs. reference.

(b) Tracking error.

(c) Crane tip position in the inertial reference frame.

(d) Motor currents demanded by the control law.

Figure 4.2: Results from the simulation where the lower and upper arms where lifted from the lower to upper limits. A load of 16kg was attached to the crane tip.

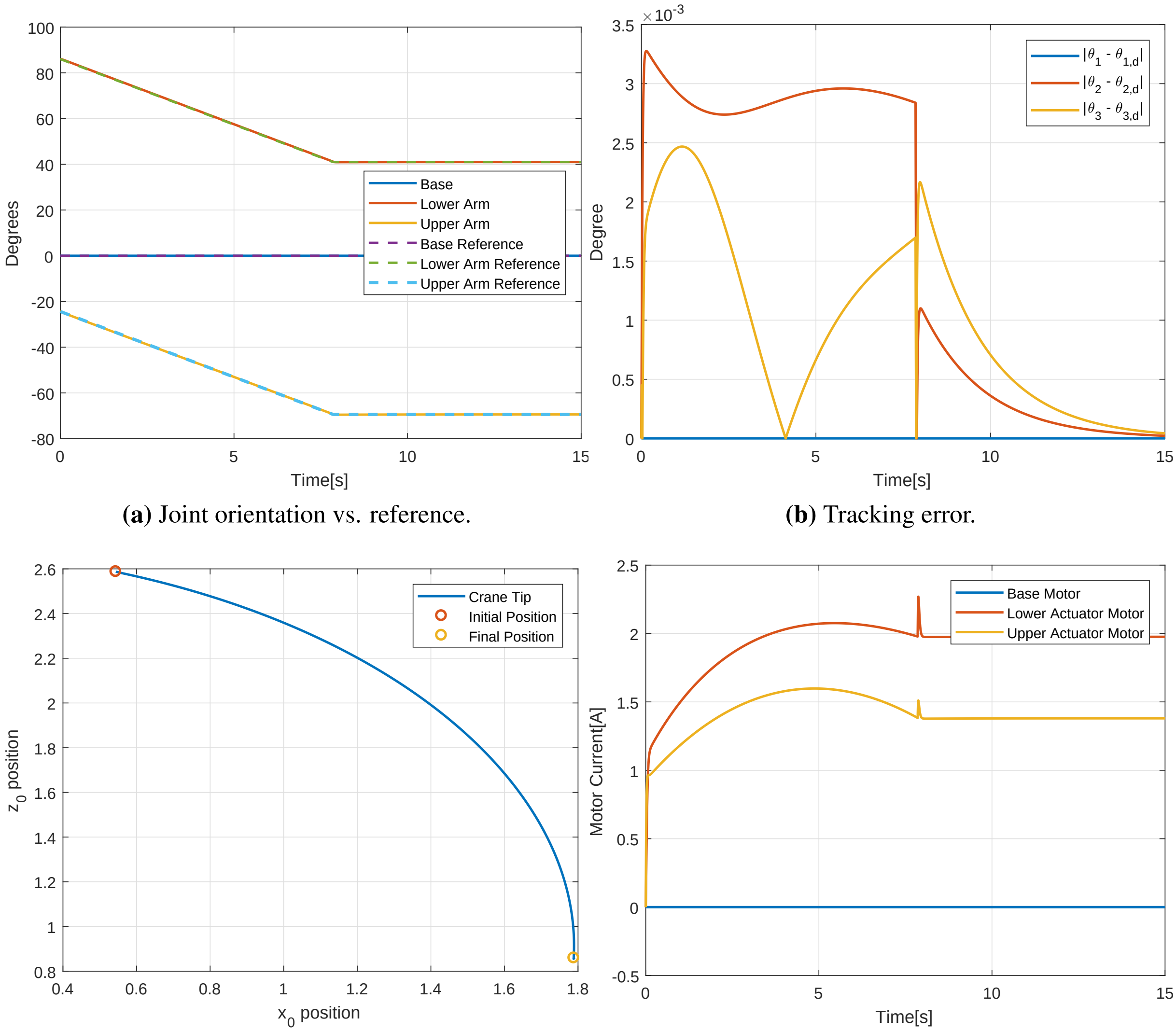
The results from the lifting test are shown in Figure 4.2. It can be observed in Figure 4.2a that both the lower and upper arm were able to follow the reference signal. The tracking error, displayed in Figure 4.2b, was in the scale of 10^{-3}, and the constant deviation, caused by gravity, was removed by the integrator term in about five seconds.

The motion of the crane tip is shown in Figure 4.2c, which shows a smooth transition from the initial to final position. During the simulation the maximum current demanded from the actuators, Figure 4.2d, were about 2.3A for the lower actuator and 1.2A for the lower. These values are well within the maximum current of 7A that can be drawn by the actuators in the crane lab. It can also be observed that both actuators stabilize at a constant current when the desired position is reached, to hold the arms in place.

In summary, the control law designed in Section 4.1 is able to position the crane arms, governed by the crane dynamics from Section 3.1, at an orientation above their initial orientation. The estimated currents drawn by the actuator motors have been found to be well within the limitations of the actuators in the crane lab from Section 1.1. The next simulation will test the controller's ability to lower the crane arms.

Lowering

The second simulation tested the controller's capability of lowering the lower and upper arm from their upper to lower limits. The reference signal was a ramp moving from the upper to lower position in seven seconds. A load of 16kg was attached to the tip of the crane.



(a) Joint orientation vs. reference.

(b) Tracking error.

(c) Crane tip position in the inertial reference frame.

(d) Motor currents demanded by the control law.

Figure 4.3: Results from the simulation where the lower and upper arms where lowered from the upper to lower limits. A load of 16kg was attached to the crane tip.

The results of the lowering simulation are shown in Figure 4.3 and in general the same as for the lifting simulation.

The joint angles, Figure 4.3a, is able to follow the desired reference signal. The tracking error, Figure 4.3b, is again in the scale 10^{-3} and any constant deviation is quickly removed by the integral term.

The transition from the initial position of the tip to the final, Figure 4.3c, is smooth. The currents drawn by the actuator motors, Figure 4.3c, are smaller since the motions are helped by gravity.

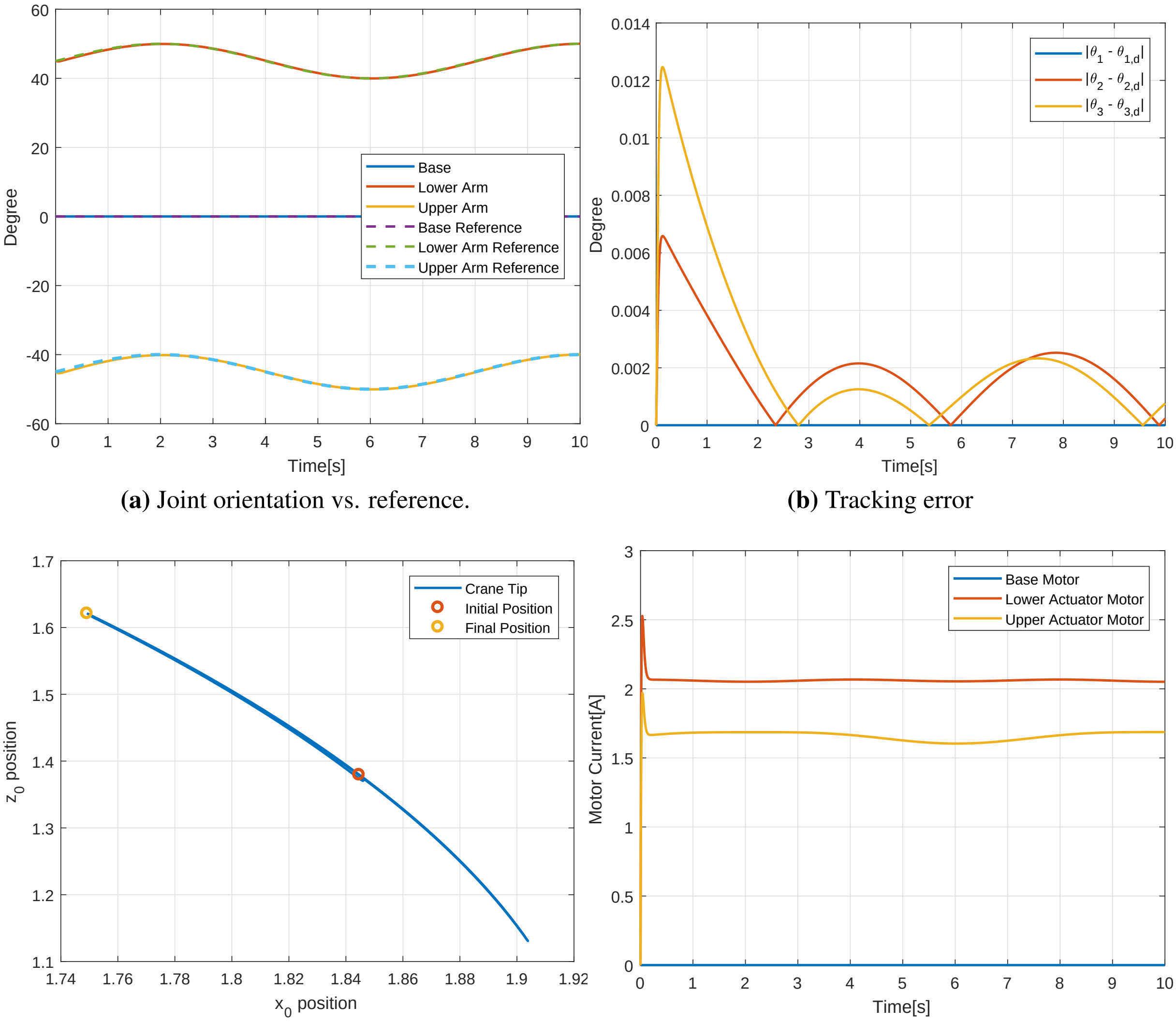
However, the similar peaks in current can be observed when the reference is reached.

In summary, the lowering simulation has shown that the control law designed Section 4.1 is able to position the crane arms, governed by the crane dynamics from Section 3.1, at an orientation below their initial orientation.

Sine Tracking

The last simulation was a heave-like simulation scenario where the lower and upper arms were made to follow a sine shaped reference signal. The reference signal for both arms were given by

$$\theta_d = \frac{5\pi}{180}\sin(\frac{\pi}{4}t) \tag{4.17}$$



(a) Joint orientation vs. reference.

(b) Tracking error

(c) Crane tip position in the inertial reference frame.

(d) Motor currents demanded by the control law.

Figure 4.4: Results from the simulation where the lower and upper arms where made to follow a sine shaped reference signal. A load of 16kg was attached to the crane tip.

The results from the sine tracking simulation are presented in Figure 4.4. Both the lower and upper arm were able to follow the reference signal given by (4.17) for the duration of the simulation. It can be observed in Figure 4.4a that both arms used about two seconds to converge to the reference signal.

The same result can be observed in the tracking error, Figure 4.4b, and after converging, the error of both arms followed a sine shape. Meaning that the orientations were off by a small amount. However, the error was in the scale of 10^{-3}, same as for the lifting and lowering simulations, and can be considered negligible.

The motion of the crane tip, Figure 4.4c, was smooth, oscillating between a maximum and minimum amplitude. The simulation was ended when the crane tip was in the top position. The motions along the vertical axis was larger then along the horizontal. The tip shifted between a height 1.6m to 1.1m while the horizontal position only shifted between 1.9m and 1.74m. However, this is to be expected since the vertical position of the tip more amplified by small motions than the horizontal position.

The motor currents, Figure 4.4d, peaked when the reference signal was initialized and stabilized at about 2.1A for the lower actuator and 1.7A for the upper. Small fluctuations in can be observed, since the actuators had to keep up with the reference signal. However, the maximum currents are well within the maximum values of the motors in the lab.

4.2 Soft Landing Winch Controller

One of the main goals of this thesis is to design a soft landing control law. The controller should ensure that cargo suspended by the crane is lowered safely and lands without being damaged. The trivial solution to this problem is to use a very slow lowering speed, but this is time consuming and not very efficient. A better solution is to try and keep maximum lowering velocity for a given duration of the operation and initiate a deceleration phase as the cargo approaches the ground. This means that the there are at least two objectives: reaching the desired position and following a desired speed profile.

4.2.1 Problem Formulation

The dynamics to be stabilized is the winch dynamics that were developed in Section 3.2. The states of this system is the amount of wire left on the winch reel, the wire velocity and the current drawn by the electric motor. The control input is the voltage of the electric motor. Hence, the system states and input can be written as

$$x_1 = L \qquad x_2 = \dot{L} \qquad x_3 = i \qquad u = V \tag{4.18}$$

where L is the amount of wire on the reel, i the motor current and V the input voltage.

In Section 3.2 it was shown that the dynamics of this system can be written as

$$\tag{4.19a}$$

$$\dot{x}_1 = x_2 \tag{4.19b}$$

$$\dot{x}_2 = \frac{D^{x_1}(x_1)}{D(x_1)}x_2^2 + \frac{D(x_1)}{2I(x_1)}K_t\eta_t x_3 - \frac{D(x_1)^2}{4I(x_1)}\frac{m_L g}{e^{\mu(\phi_1+\phi_2+\phi_3)}} \tag{4.19c}$$

$$\dot{x}_3 = -\eta_\omega \frac{K_v}{L_a}\frac{D(x_1)}{2}x_2 - \frac{R_a}{L_a}x_3 + \frac{1}{L_a}u \tag{4.19d}$$

The same procedure as in Section 4.1 will be used to design the soft landing controller. However, the control objective will be different since it is important that the wire velocity is small when approaching the desired position. Therefore, the control objective and following design will be treated as a tracking problem.

The tracking problem is formalized and detailed in (Skjetne et al., 2004) and is usually formulated by defining one or several secondary control objectives. These may be restrictions in velocity, time or acceleration. The tracking problem for the soft landing controller will be defined as

$$\lim_{t\to\infty} |L(t) - L_d(s(t))| = 0 \tag{4.20a}$$

$$\lim_{t\to\infty} |\dot{s}(t) - v(s(t), t)| = 0 \tag{4.20b}$$

where $L_d(s(t))$ is the desired amount of wire that should be stored on the reel, $v(s(t), t))$ the speed assignment and s a variable parametrizing the desired position and velocity. This means that (4.20a) expresses that the amount of wire left on the reel should follow a desired path that is generated by s and (4.20b) that the speed at which the path is generated should follow a defined velocity profile. The next section will discuss the selection of a desired path and velocity profile.

4.2.2 Path and Speed Profile

The cargo should follow a straight line towards the ground while its speed should follow some desired speed. A one dimensional linear path parametrized by s can be written as

$$L_d(s) = as + L_0 \tag{4.21}$$

where L_0 is the initial amount of wire on the reel.

The speed profile should be chosen such that the winch reaches maximum speed as soon as possible and keeps this speed until a certain point in the path. When reaching this point, the velocity should slow down such that the cargo is slowly lowered during the last phase.

A speed profile that is inspired by the work of (Skjetne et al., 2004) will be used in this thesis. The chosen speed profile was originally used to make the tip of a robotic manipulator follow a triangular path. This required a speed profile that slowed down when the tip approached the vertices of the triangle. The profile is given by the $\arctan$ function and can be written as

$$v(s) = \begin{cases} \frac{u_{max}}{\pi|y_d^s|} \arctan(\frac{s-s_k-a_1}{a_2}) + \frac{u_{max}}{2|y_d^s|}; & s \in [s_k, +f(s_{k+1}-s_k)] \\ \frac{u_{max}}{\pi|y_d^s|} \arctan(\frac{s_{k+1}-a_3-s}{a_2}) + \frac{u_{max}}{2|y_d^s|}; & s \in [s_k + f(s_{k+1}-s_k), s_{k+1}] \end{cases} \tag{4.22}$$

where u_{max} is the maximum velocity, y_d^s the derivative of the path with respect to s, a_1, a_2 and a_3 shaping parameters and f determines at which point the deceleration phase is initiated.

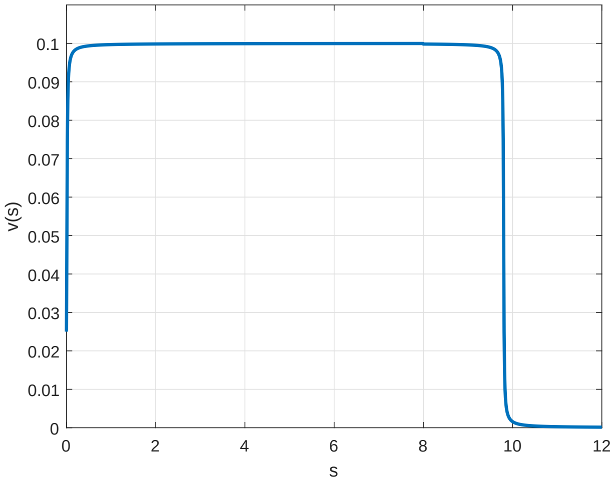


Figure 4.5: Soft landing speed profile.

A plot of the speed profile given by (4.22) is shown in Figure 4.5 and has been generated with the parameters in Table 4.4. It shows that maximum speed is given for most of the path, except towards the end where the speed slowly decelerates.

Table 4.4: Speed Profile Parameters

Parameter	Value	Parameter	Value
s_1	0	s_2	12
a_1	0.01	a_2	0.01
a_3	0.2	y_d^s	-1
f	$\frac{4}{5}$	u_{max}	0.1

Now that the path and speed profile has been selected, it is time to design the control law.

4.2.3 Control Law Derivation

The soft landing control law will de designed by using the same backstepping technique as in Section 4.1. The main difference is of course the dynamics of the winch and also the control objective. The states that will be used for backstepping is written as

$$z_1 = L - L_d(s) \tag{4.23a}$$

$$z_2 = \dot{L} - \alpha_1 \tag{4.23b}$$

$$z_3 = i - \alpha_2 \tag{4.23c}$$

where $L_d(s)$ is a general path parametrized by s, α_1 a virtual controller that will stabilize z_1 and α_2 a virtual controller to stabilize z_2. The path speed $\dot{s}$ is simply set equal to the speed profile

$$\dot{s} = v(s) \tag{4.24}$$

The winch system is of relative degree three, meaning that the design procedure will consist of three steps.

Step 1
The Lyapunov equation for the first step is written as

$$V_1 = \frac{1}{2} z_1^2 p_1 \tag{4.25}$$

where z_1 is defined in (4.23a) and p_1 is a positive value. The time derivative of (4.25) can be written as

$$\dot{V}_1 = z_1 p_1 (z_2 + \alpha_1 - L_d^s \dot{s}) \tag{4.26}$$

where L_d^s is the derivative of $L_d(s)$ with respect to s. In order to stabilize z_1, α_1 must be chosen to render (4.26) negative definite. This is done by choosing α_1 as

$$\alpha_1 = L_d \dot{s} + a_1 z_1 \tag{4.27}$$

$$\dot{\alpha}_1 = \alpha_1^L \dot{L} + \alpha_1^s \dot{s} + \alpha_1^t \tag{4.28}$$

where a_1 must be negative. Inserting (4.27) into (4.26) yields

$$\dot{V}_1 = -z_1^2 a_1 + z_1 p_1 z_2 \tag{4.29}$$

showing that $\dot{V}_1$ is negative definite with respect to the z_1 terms. Terms including z_2 are carried over to the next step.

Step 2
The Lyapunpov function for the second step is written as

$$V_2 = V_1 + \frac{1}{2} z_2^2 p_2 \tag{4.30}$$

where V_1 is the Lyapunov function from the first step (4.25) and p_2 a positive value. Taking the time derivative of V_2 and inserting the $\dot{L}$ and $\ddot{L}$ dynamics (4.19b) and (4.19c) gives

$$\dot{V}_2 = z_1^2 q_1 + z_1 p_1 z_2 + z2p_2(\frac{D^L(L)}{D(L)}\dot{L}^2 + \frac{D(L)}{2I(L)}K_t\eta_t i \tag{4.31a}$$
$$- \frac{D(L)^2}{4I(L)}\frac{m_L}{e^{\mu(\phi_1+\phi_2+\phi_3)}} - \dot{\alpha}_1)$$
$$\dot{V}_2 = z_1^2 q_1 + z_1 p_1 z_2 + z2p_2(\frac{D^L(L)}{D(L)}(z_2+\alpha_1)^2 + \frac{D(L)}{2I(L)}K_t\eta_t (z_3+\alpha_2) \tag{4.31b}$$
$$- \frac{D(L)^2}{4I(L)}\frac{m_L}{e^{\mu(\phi_1+\phi_2+\phi_3)}} - \dot{\alpha}_1)$$

This time α_2 must be chosen to render $\dot{V}_2$ negative definite. This is achieved by choosing α_2 to be

$$\alpha_2 = \frac{2I(L)}{D(L)}(a_2 z_2 - z_1\frac{p_1}{p_2} - \frac{D^L(L)}{D(L)K_t\eta_t}(z_2+\alpha_1)^2 + \dot{\alpha}_1) \tag{4.32a}$$
$$\dot{\alpha}_2 = \alpha_2^L \dot{L} + \alpha_2^{\dot{L}}\ddot{L} + \alpha_2^s \dot{s} + \alpha_2^t \tag{4.32b}$$

where a_2 must be negative. Inserting (4.32a) into (4.31b) results in

$$\dot{V}_2 = -z_1^2 a_1 - z_2^2 a_2 + z_2 p_2 \frac{D(L)}{2I(L)} K_t \eta_t z_3 \tag{4.33}$$

which shows that the $\dot{V}_2$ is rendered negative definite with respect to z_2, excluding terms including z_3. In the third, and last, step the final control law for the motor voltage will be chosen.

Step 3
The Lyapunov function for the third step is chosen as

$$V_3 = V_2 + \frac{1}{2} z_3^2 p_3 \tag{4.34}$$

where V_2 is the Lyapunov equation from the second step and p_3 a positive value. Taking the time derivative of (4.34) results in

$$\dot{V}_3 = -z_1^2 q_1 - z_2^2 q_2 + z_2 p_2 \frac{D(L)}{2I(L)} K_t\eta_t z_3 + z_3 p_3(-\eta_\omega \frac{K_v}{L_a}\frac{D(L)}{2}\dot{L} \tag{4.35a}$$
$$- \frac{R_a}{L_a} i + \frac{V}{L_a} - \dot{\alpha}_2)$$
$$\dot{V}_3 = -z_1^2 q_1 - z_2^2 q_2 + z_2 p_2 \frac{D(L)}{2I(L)} K_t\eta_t z_3 + z_3 p_3(-\eta_\omega \frac{K_v}{L_a}\frac{D(L)}{2}(z_2+\alpha_1) \tag{4.35b}$$
$$- \frac{R_a}{L_a}(z_3+\alpha_2) + \frac{V}{L_a} - \dot{\alpha}_2)$$

The control law for the motor voltage is defined as

$$V = L_a(\frac{R_a}{L_a}(z_3 + \alpha_2) + \eta_w \frac{K_v}{L_a}\frac{D(L)}{2}(z_2 + \alpha_1) - z_2 \frac{p_2}{p_3}\frac{D(L)}{2I(L)}K_t \eta_t + a_3 z_3) \tag{4.36}$$

where a_3 must be negative. Inserting (4.36) into (4.35) yields

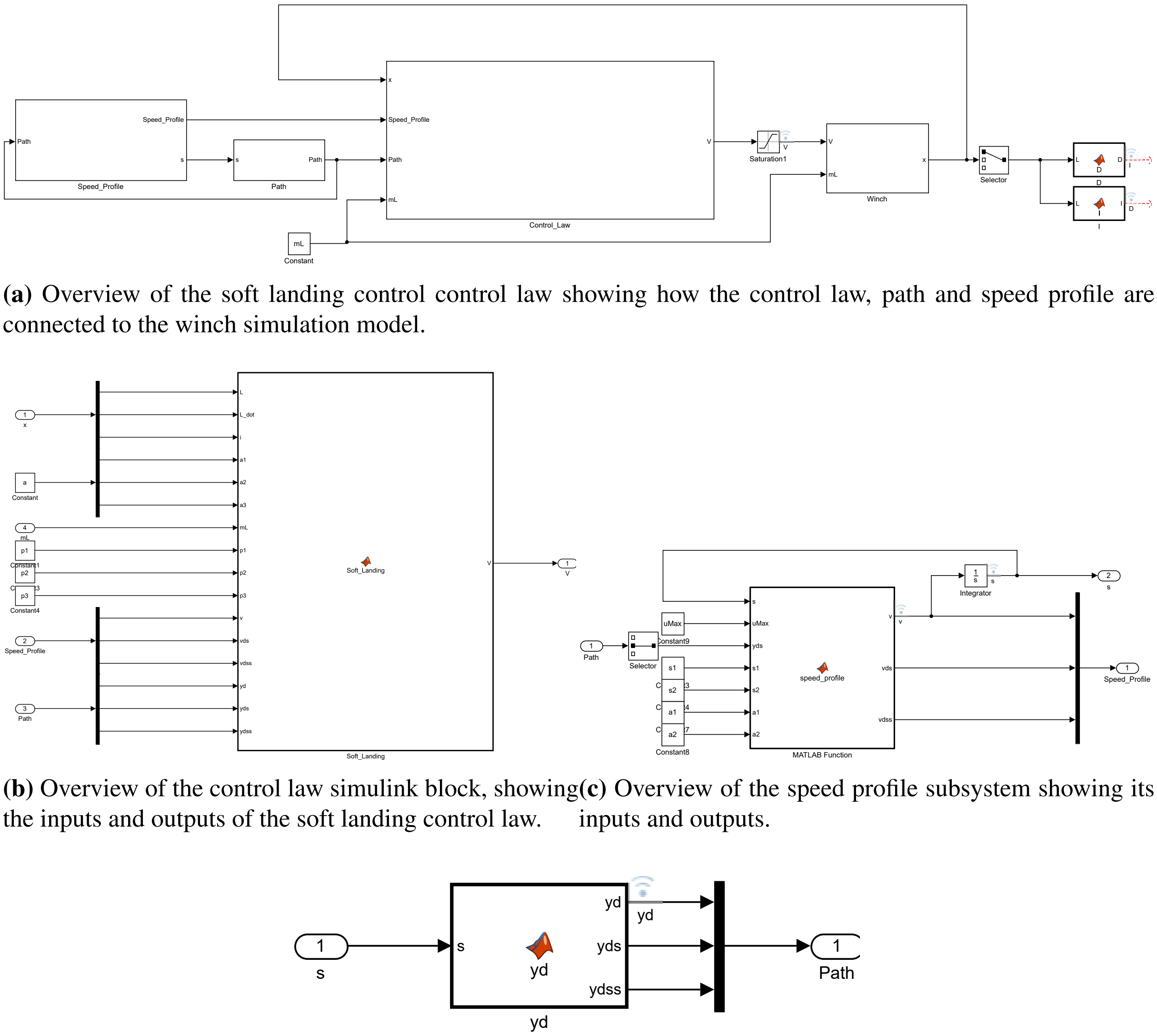
$$\dot{V}_3 = -a_1 z_1^2 - a_2 z_2^2 - a_3 z_3^2 \tag{4.37}$$

showing the $\dot{V}_3$ is negative definite and the system is globally asymptotically stable by Lyapunov's direct method 2.4.2.

The control law for soft landing has now been designed and can be implemented in Simulink and connected to the winch dynamics developed in Section 3.2.

4.2.4 Simulink Implementation

The control law designed in the previous section (4.36) has been implemented in Simulink and connected with the simulation model developed in Section 3.2. The resulting model is shown in Figure 4.6.



(a) Overview of the soft landing control control law showing how the control law, path and speed profile are connected to the winch simulation model.

(b) Overview of the control law simulink block, showing the inputs and outputs of the soft landing control law.

(c) Overview of the speed profile subsystem showing its inputs and outputs.

(d) Overview of the path profile subsystem showing its inputs and outputs.

Figure 4.6: Soft landing control law Simulink implementation.

Figure 4.6a shows the interfacing between the simulation model and control law. The controller has been split into the control law, path and speed profile.

The detailed overview of the path generator, Figure 4.6d shows that its output also includes the first and second order derivatives of the path with respect to s as these are needed by the control law. The same is true for the speed profile, Figure 4.6c. The control law itself is shown in Figure 4.6b and can be customized to include different cargo weights. The Simulink interface allows any path and speed profile to be used with the control law.

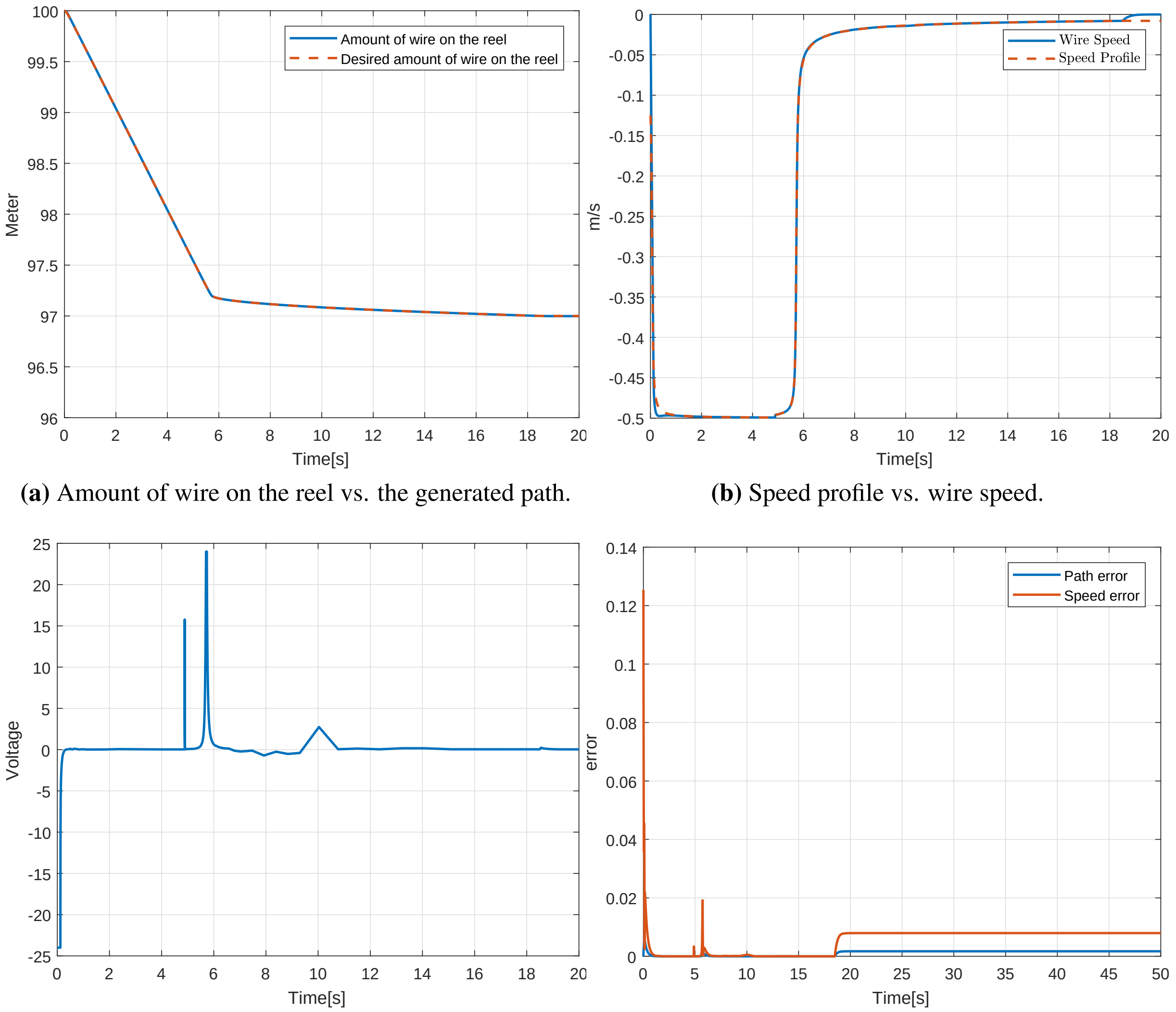
4.2.5 Simulation Results

The control law for soft landing that was designed in Section 4.2 has been implemented in Simulink and connected to the winch dynamics that were derived in Section 3.2. Hence, the performance of the control law can be now be estimated by running a simulation scenario. The simulation will be run with the speed profile given by (4.22) and the straight line path given by (4.21).

The winch and wire parameters are the same as in the simulation from Section 3.2.4, given in Tables 3.3, 3.4, 3.5 and 3.6. The controller has been tuned and the parameters given in Table 4.5 have been found to give a satisfactory performance. The speed profile are given by the parameters in Table 4.4, except that $s \in [0, 3]$.

Table 4.5: Soft Landing Tuning Parameters

Parameter	Value
p_1	0.01
p_2	0.1
p_3	0.0001
a_1	-50
a_2	-5
a_3	-5000



(a) Amount of wire on the reel vs. the generated path.

(b) Speed profile vs. wire speed.

(c) Voltage applied to the winch motor.

(d) Path and speed error.

Figure 4.7: Results from simulating the soft landing controller with the winch dynamics.

The results from simulating the soft landing control law with the winch dynamics are given in Figure 4.7. It can be observed that the amount of wire on the reel was able to follow the desired reference almost perfectly. The largest path errors, shown in Figure 4.7d, can be observed when the path was initialized and when the speed velocity entered the slower phase.

The wire velocity was able to follow the desired speed profile. The largest deviations occurred when the speed profile was initialized and when the slowdown phase began, at approximately six seconds. A constant deviation can be observed after the path reached its final destination, but this is because the speed profile was not set to zero when the path stopped.

The voltage supplied to the winch motor can be observed in Figure 4.7d. A limit of ±24V was imposed on the control signal, as to better represent the motor that is used in the crane lab. It can be seen that maximum voltage was applied when the path was initialized. However, very little voltage is needed to keep wire moving. This is most likely due to low friction in the simulation model and that the load helps pulling the wire. Voltage spikes occurred when the path slowed down, at six seconds. Afterwards, a small voltage was needed to slow the wire down, and the final spike appeared when the load reached its final destination.

In summary, the soft landing controller designed in Section 4.2 was able to follow the path given by (4.21) and speed profile given by (4.22).

4.3 Chapter Summary and Discussion

In this chapter a control law for positioning the joints of the knuckleboom crane from Section 1.1 was developed. The control law was based on the crane dynamics developed in Section 3.1. In addition, a soft landing control law for automatically lowering crane cargo was designed. This control law was based upon the winch dynamics developed in Section 3.2.

The position control law was simulated with the crane dynamics and was able to lift and lower the crane arms to a desired reference, and make the arms simultaneously follow sine shaped reference signal. The results are presented in Section 4.1.4. However, how the control law will perform when tested with the crane in the lab will be determined by the accuracy of the model and robustness of the control law.

For instance, actuator dynamics have largely been neglected when developing the crane model and the control law. It has been assumed that their response will be fast and the estimate of current drawn by the motors have been calculated using steady-state parameters. However, the actuator housing may affect the dynamics in ways that have not been predicted. Their friction parameters have not been well documented and it may be that friction will have a larger contribution than assumed. Especially, the transition from static to dynamic friction.

In addition, a soft landing control law was developed from the winch dynamics derived in Section 3.2. The soft landing problem has been defined as a tracking problem, and the goal is to lower the load in a straight path by following a pre defined velocity profile. The velocity profile was chosen to keep maximum speed for most of the operation, until a defined point in the path is reached, after which the velocity is lowered and gently converges to zero.

Simulation showed that the soft landing controller was able to follow the desired speed profile perfectly. However, there are uncertainties regarding the parameters used in the simulation, especially regarding friction and pulley span.

Chapter 5

Lab Setup

So far simulation models of the knuckleboom crane and winch systems from Section 1.1 have been developed. In addition, control laws for positioning the crane arms and softly landing the crane load was designed in Chapter 4 and initial tuning parameters have been found by running the control laws with the simulation models.

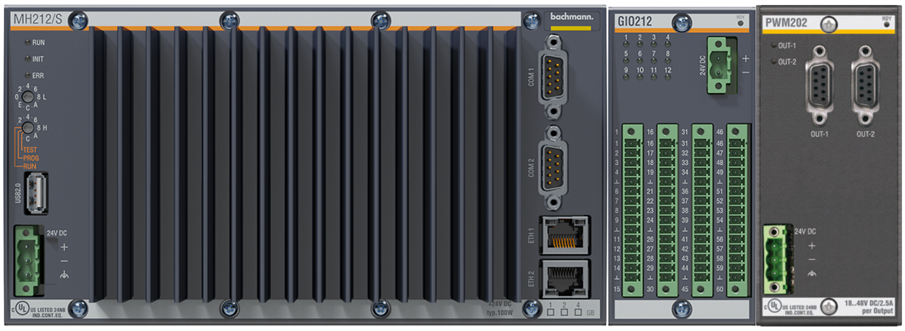
This means that the control laws are almost ready to be tested in the crane lab. However, the setup of the lab must first be reviewed. In addition, one of the goals of this thesis is to finish the lab design envisioned by (Gyberg, 2017) and design sensor filters. Therefore, this section will review the lab equipment, design a low-pass filter for removing high frequency noise and design a Kalman filter for estimating gyro bias.

5.1 Equipment

The chapter will start by reviewing the most essential equipment that will be used when testing the control laws from Chapter 4. This will cover the lab's main controller, low-level motor controllers, joysticks and sensors.

5.1.1 Main Programmable Logic Controller

A PLC from Bachmann electronics will be used as the main controller in the lab and will be responsible for executing control laws and handling low-level I/O. The controller consists of the MH212/S processor module and is equipped with a rail-like interface that supports additional modules. In this thesis the general GIO212 I/O module and the PWM202 module will be used. The first to handle analogue and digital I/O signals and the latter to control the winch brake. The controller and its modules are depicted in Figures 5.1a, 5.1b and 5.1c. For for more documentation about the individual modules see (Bachmann, 2017)



(a) MH212/S (b) GIO212 (c) PWM202

Figure 5.1: Bachmann controller modules used in the lab.

5.1.2 Motor Controllers

The Bachmann controller is not capable of handling the currents drawn by the actuator and winch motors during operation. Therefore, external motor controllers will be used to control the velocity and torque of the actuators and winch.

The ESCON Module 50/5 4-Q motor controller will be used for the actuator motors. This controller can be configured to run either a speed or current controller. In the experiments in this thesis, it will be configured to rune the current controller, since the output of the position control law from Section 4.1 outputs the force that should be produced by each actuator. The current setpoint is provided as an analogue voltage between $\pm 10\text{V}$.

The winch motor will use 1-Q-EC Amplifier DEC Module 50/5 the motor controller. This controller will be configured to run a speed controller, where the desired speed is directly proportional to an analogue voltage signal. This mode will be used for hands-on winch control.

5.1.3 Sensors

Three types of sensors will be used in the experiments: gyroscopes, accelerometers and stop sensors.

The gyroscopes are of the type ADXR2624, produced by Analog devices, documentation can be found at (Devices, 2017). It was planned to use three gyroscopes, one for each degree of freedom of the crane. However, the sensor intended for the base was defective upon delivery and a new one could not be obtained in time for the experiments. Therefore it was prioritized to measure the angular rate of the lower and upper arms. The gyroscope parameters provided by the manufacturer are given in Table 5.1.

Table 5.1: Gyroscope parameters provided by the manufacturer

Parameter	Value	Unit
Sensitivity	25	$\text{mV}/^\circ\,\text{s}^{-1}$
Offset	2.5	V

The accelerometers are of the type MXR9500G/M, produced by MEMSIC, documentation can be found at (Memsic, 2017). They will be used to measure the absolute orientation of the crane arms and to estimate the bias of the gyroscopes. The sensors have been placed as close as possible to the mass center of each arm. The parameters from the manufacturer are given in Table 5.2.

Table 5.2: Accelerometer Parameters

Parameter	Value	Unit
Sensitivity	500	mV/g
Zero g offset	1.5	V

The stop sensors are inductive sensors made by Contrinex with the model number DW-AD-623-04, see (Contrinex, 2017) for more information. They register the actuator as it approaches its limits and outputs a digital signal. The stop sensors for the upper actuator are shown in figure 5.2 and are placed similarly for on the lower actuator. They will be used to stop the electric motors when the actuator's approach their maximum or minimum elongation.

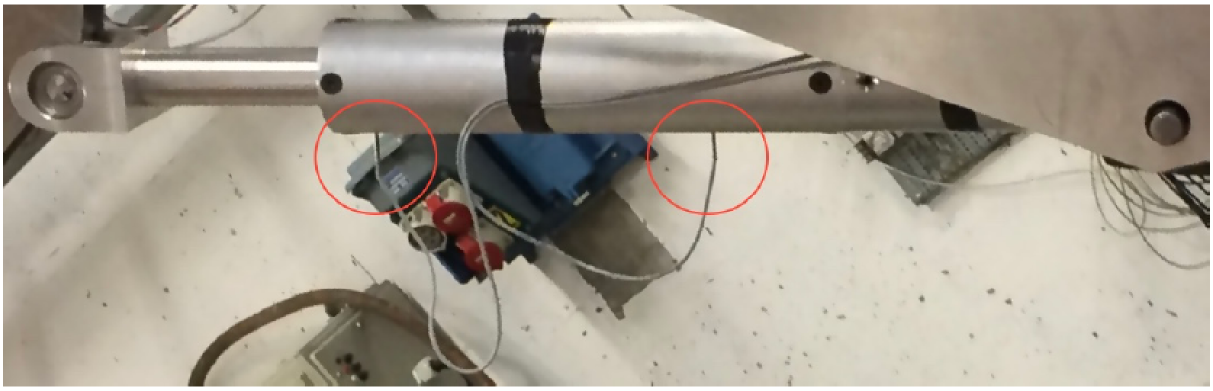


Figure 5.2: Proximity sensors in the upper actuator. Taken from (Gyberg, 2017).

5.2 Signal and Sensor Processing

This section will describe the processing that is necessary in order to read measurement signals from the gyroscopes and accelerometers from Section 5.1.3. This will include digital to analogue conversion, absolute orientation measurements by using the accelerometer, low-pass filter design and gyroscope bias estimation using a Kalman filter.

5.2.1 Digital to Analogue Conversion

The analogue channels of the Bachmann controllers reads or writes voltages between ±10V. For output channels, the desired voltage is set by a 16-bit signed integer and for the input channel the voltage is read with the same resolution. This means that the read value must be processed in order to either recover or produce the desired voltage. The maximum values that can be stored by a 16-bit signed integer is 32767 in the positive range and -32768 in the negative. Amounting to a maximum of 65535 values. Hence, the conversion from a 16-bit signed integer to a voltage is given by

$$V = \frac{V_{range}}{65535} SINT16 \tag{5.1}$$

where V is the read voltage, V_{range} the voltage range and $SINT16$ the 16-bit signed integer. To find the integer that produces a desired voltage simply take the inverse of (5.1).

5.2.2 Measuring Absolute Position

It will now be shown how the accelerometers presented in Section 5.1.3 can be used to measure the absolute orientation of the crane. Being able to measure absolute orientation is convenient since it removes the need of defining the crane's initial position manually and will also make it possible to estimate the drift of the gyroscopes.

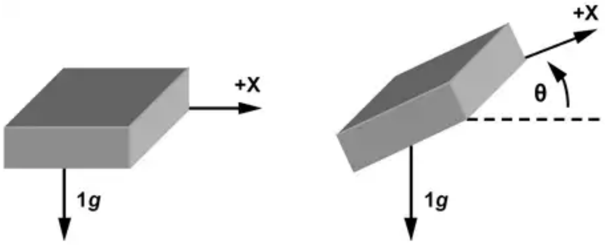


Figure 5.3: Accelerometer tilted an angle θ relative to the horizontal axis.

An accelerometer that is tilted at an angle θ relative to the horizontal axis is shown in Figure 5.3. When $\theta = 90°$, the accelerometer will measure $1g$ and $0g$ when $\theta = 0°$. Hence, the gravitational acceleration measured by the accelerometer varies with orientation. This means that the measurements from thee accelerometer can be used to determine its orientation.

From Figure 5.3, it can be observed that the orientation of the accelerometer is given by

$$\theta = \arcsin(a_0) \tag{5.2}$$

where θ is the orientation of the accelerometer relative to the horizontal axis and a_0 the measured acceleration in g.

5.2.3 Low-Pass Filter

Measurements from the gyroscopes will be corrupted by high frequency noise and will, if passed directly into the control law, cause unnecessary wear and tear of the actuators and in a worst case scenario lead to instabilities. A low-pass filter will therefore be designed to remove high frequency components and provide a smooth measurement signal

A first-order low-pass filter can be written as

$$\frac{y(s)}{x(s)} = \frac{K}{\tau s + 1} \tag{5.3}$$

in the Laplace notation, where $y(s)$ is the output, $x(s)$ the input, K the passband gain and τ the time constant that determines the cutoff frequency. In the time domain the filter can be written as

$$\dot{y} = \frac{Kx - y}{\tau} \tag{5.4}$$

(5.4) can be discretized by assuming that the timestep is significantly smaller than the cutoff period. The discretized low-pass filter can be written as

$$y_{k+1} = y_k + (Kx - y_k)\frac{\Delta t}{\tau} \tag{5.5}$$

where Δt is the timestep. When implementing the discrete low-pass filter the cutoff frequency must be selected such that most of the high frequency noise components of the signal are removed. Small values of τ will result in a slow filter and may negate the assumption that Δt is much smaller than τ. However, the low-pass filter will not remove the constant bias that affects most gyroscopes.

5.2.4 Kalman Filter for Estimating Gyro Drift

A Kalman filter that combines measurements from the gyroscopes and accelerometer will be designed to remove noise from the acceleration measurements and estimate the gyroscope bias. If the bias is not removed, it will result in large integration errors when calculating orientation from the gyroscopes measurements. This will cause the system to deviate from its desired position over time.

A simple discrete model for the orientation and bias can be written as

$$\begin{bmatrix} \theta_k \\ b_k \end{bmatrix} = \begin{bmatrix} 1 & -\Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \theta_{k-1} \\ b_{k-1} \end{bmatrix} + \begin{bmatrix} \Delta t \\ 0 \end{bmatrix} \dot{\theta}_k \tag{5.6}$$

$$y_k = \theta_k + v_k \tag{5.7}$$

where θ_k is the orientation, b_k the bias, Δt the timestep, $\dot{\theta}_k$ the velocity measured by the gyro, k the timestep index and v_k is assumed to be zero mean white noise. The model given by (5.6) states that the orientation is given by the orientation of the previous timestep plus the measured velocity, minus the bias, multiplied by the timestep. This model will be used to calculate the *a priori* states in the Kalman filter and can be written more compactly as

$$\hat{\boldsymbol{x}}_{k|k-1} = \boldsymbol{F}_k \hat{\boldsymbol{x}}_{k-1|k-1} + \boldsymbol{B}_k \boldsymbol{u}_k \hat{\boldsymbol{y}}_k = \boldsymbol{H}_k + \boldsymbol{v}_k \tag{5.8a}$$

where the system matrices are given by

$$\boldsymbol{F}_k = \begin{bmatrix} 1 & -\Delta t \\ 0 & 1 \end{bmatrix} \qquad \boldsymbol{B}_k = \begin{bmatrix} \Delta t \\ 0 \end{bmatrix} \qquad \boldsymbol{H}_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{5.9}$$

The predicted a *a priori* covariance estimate is given by

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{F}_k \boldsymbol{P}_{k-1|k-1} \boldsymbol{F}_k^T + \boldsymbol{Q}_k \Delta t \tag{5.10}$$

Since the bias of the gyroscope cannot be measured, the orientation is the only measured state. The residual term of the Kalman filter is therefore given by

$$\tilde{y} = \theta_k - \hat{\theta}_{k|k-1} \tag{5.11}$$

The residual covariance will also be a vector given by

$$S_k = \boldsymbol{H}_k \boldsymbol{P}_{k|k-1} \boldsymbol{H}^T + R_k \tag{5.12}$$

and the Kalman gain by

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-|} \boldsymbol{H}^T S_k^{-1} \tag{5.13}$$

the update state

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k \tilde{\boldsymbol{y}}_k \tag{5.14}$$

and finally the updated covariance matrix

$$\boldsymbol{P}_{k|k} = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}) \boldsymbol{P}_{k|-k-1} \tag{5.15}$$

This finishes the designs of the Kalman filter.

The lab equipment has now been reviewed and the low-pass and Kalman filters designed. This means that the control laws can be tested in the marine crane.

Chapter 6

Lab Results

In the previous chapter, the marine crane lab was presented. The equipment that will be used to implement and test the control laws developed in Chapter 4 was reviewed and the signal processing necessary to achieve communication between the sensors, PLC and low-level motor controllers were described. A low-pass filter for removing high frequency noise from the gyroscopes and a Kalman filter for removing accelerometer noise and estimating bias were designed. In addition, a hands-on control system for the winch and actuators were implemented.

This means that the preparations for implementing and testing the position control law are finished. Tuning the low-pass and Kalman filter is the only remaining step before the control law can be tested. Hence, this chapter will present the final results of this thesis which is the implementation, tuning and testing of the position control law.

6.1 Filters

The low-pass and Kalman filters designed in Section 5.2 must be tuned before being used with the control law. For the low-pass filter this will consist of finding a cutoff frequency that removes most of the high frequency noise, while ensuring that the filter is fast enough to keep up with the system dynamics. As mentioned in Section 5.2, the cutoff must be chosen such that the assumption that $\tau = \frac{1}{\omega_{cuttoff}} >> \Delta t$ is valid. The time between control cycles in the Bachmann controller is approximately 10ms.

For the Kalman filter, the tuning procedure will consist of finding values for R and $\boldsymbol{Q}$, which are the covariance matrices of the output and system noise. An approximation of R can be found by analysing a timeseries of the measurements and $\boldsymbol{Q}$ is difficult to determine, but can be tuned by trial and error.

Table 6.1: Low-pass filter tuning parameters

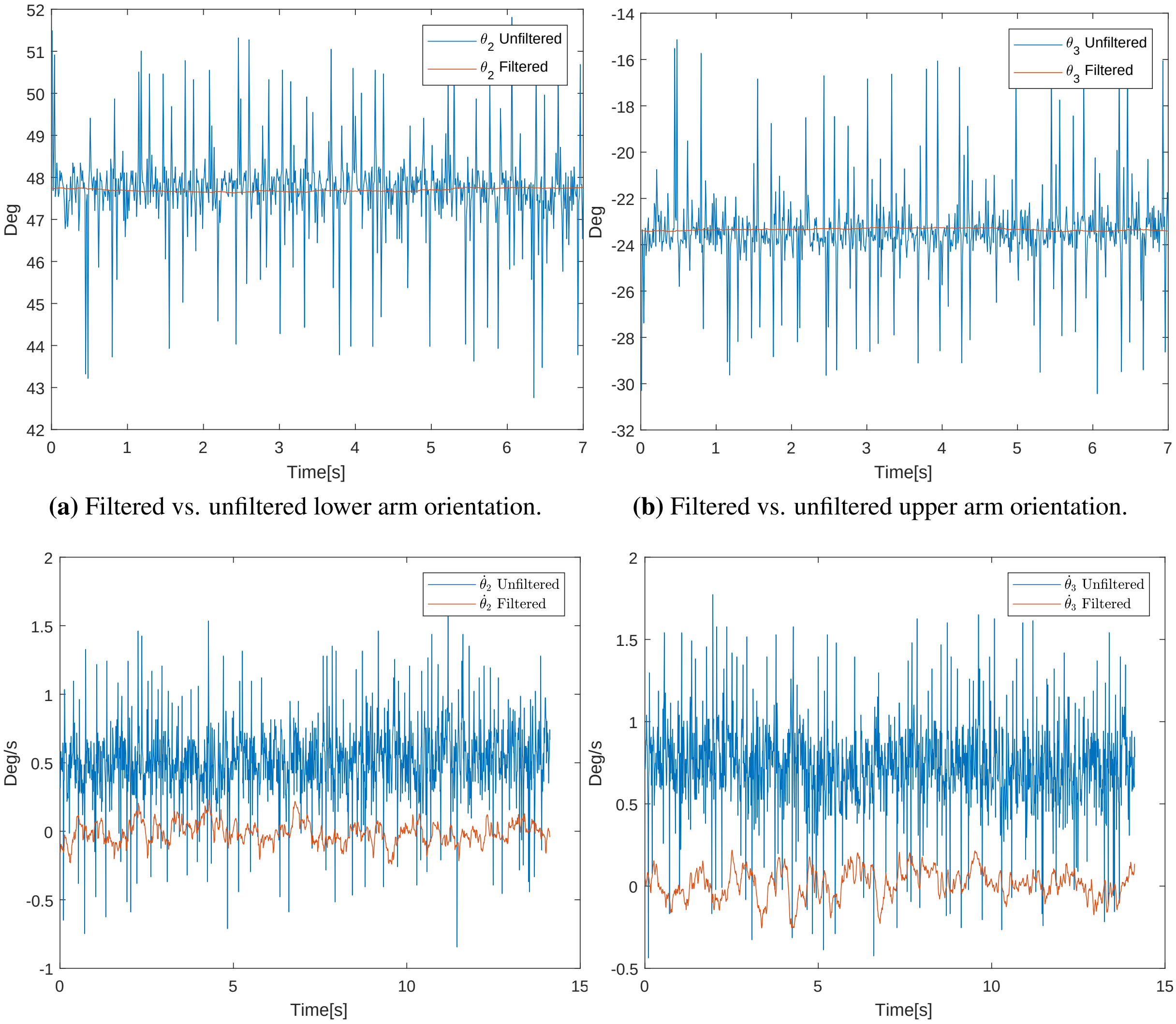
Gyro 1		Gyro 2	
Parameter	Value	Parameter	Value
K	1	K	1
τ	0.1	τ	0.1

Table 6.2: Kalman filter tuning parameters

Gyro 1		Gyro 2	
Parameter	Value	Parameter	Value
R	5	R	5
Q	$\begin{bmatrix} 0.0001 & 0 \\ 0 & 0.0003 \end{bmatrix}$	Q	$\begin{bmatrix} 0.0001 & 0 \\ 0 & 0.0003 \end{bmatrix}$

The tuning parameters for the low-pass and Kalman filter was determined through trial and error in the lab and the results are presented in Table 6.1 and 6.2. A cutoff frequency of 10Hz, corresponding to $\tau = 0.1$ provided a satisfactory compromise between noise filtering and filter speed. However, it is close to the timestep of the Bachmann controller.

The output covariance of the accelerometers were initially measured to be 1.5, however, after tuning the filter by hand a value of 5 provided good filtering. The variance of the bias and orientation was assumed independent, meaning that $\boldsymbol{Q}$ has been chosen to be a diagonal matrix.



(a) Filtered vs. unfiltered lower arm orientation.

(b) Filtered vs. unfiltered upper arm orientation.

(c) Filtered vs. unfiltered lower arm velocity.

(d) Filtered vs. unfiltered upper arm velocity.

Figure 6.1: Comparisons of unfiltered and filtered joint orientation and velocity measurements.

Figure 6.1 shows a comparison between the filtered and unfiltered sensor measurements for the lower and upper crane arms. Figure 6.1a and 6.1b compares the orientation measurement, showing that the unfiltered signal oscillated between $51°$ to $43°$ for the lower arm and $-15°$ to $-29°$ for the upper. Meaning that the unfiltered orientation was corrupted by a significant amount of noise. However, the Kalman filter was able to remove a significant part of the noise, resulting in a smooth output signal.

The velocity measurements are compared in Figure 6.1c and 6.1d which shows that the unfiltered signal oscillated between $1.5°\,\mathrm{s}^{-1}$ and $-0.5°\,\mathrm{s}^{-1}$ for both arms. In addition, the signal bias causes an offset resulting in a mean velocity of approximately $0.5°\,\mathrm{s}^{-1}$. The low-pass filter was able to remove most of the noise, but some components remains. However, the original noise is significantly diminished. When removing the bias approximated by the Kalman filter, the measurement oscillates around zero mean.

In summary, the low-pass and Kalman filter was able to remove most of the gyroscope and accelerometer noise and estimate the gyro bias. However, some noise still remains in the gyroscope measurements, but it is expected that the filtering is good enough to be used in the control law. The filters could be improved by additional tuning, expanding the low-pass filter to higher order, using an extended Kalman filter based on the crane model or by adding additional sensors to the system.

6.2 Position Control Law

The low-pass and Kalman filters have now been tuned, meaning that joint orientation and velocity measurements can be used in a state-feedback control law. Therefore, the joint position controller that was designed in Section 4.1 can now be implemented and tested in the crane lab.

This section will present a series of tests where orientation of the crane will be made to follow a ramp and a sine shaped reference signal. The ramp reference will test the ability to lower and lift the crane arms and the sine shaped reference will be used to simulate heave compensation.

The control law has been tuned in preparation of the experiments and the final tuning parameters can be found in Table 6.3. The parameters of the wire can be found in Table 6.4.

Table 6.3: Final tuning parameters for the position control law.

Parameter	Value	Parameter	Value
P_1	$\begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.001 \end{bmatrix}$	P_2	$\begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}$
Q_1	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	Q_2	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
A_1	$\begin{bmatrix} -5 & 0 & 0 \\ 0 & -50 & 0 \\ 0 & 0 & -500 \end{bmatrix}$	A_2	$\begin{bmatrix} -5 & 0 & 0 \\ 0 & -5 & 0 \\ 0 & 0 & -5 \end{bmatrix}$
K_i	50		

Table 6.4: Parameters of the wire used in the lab.

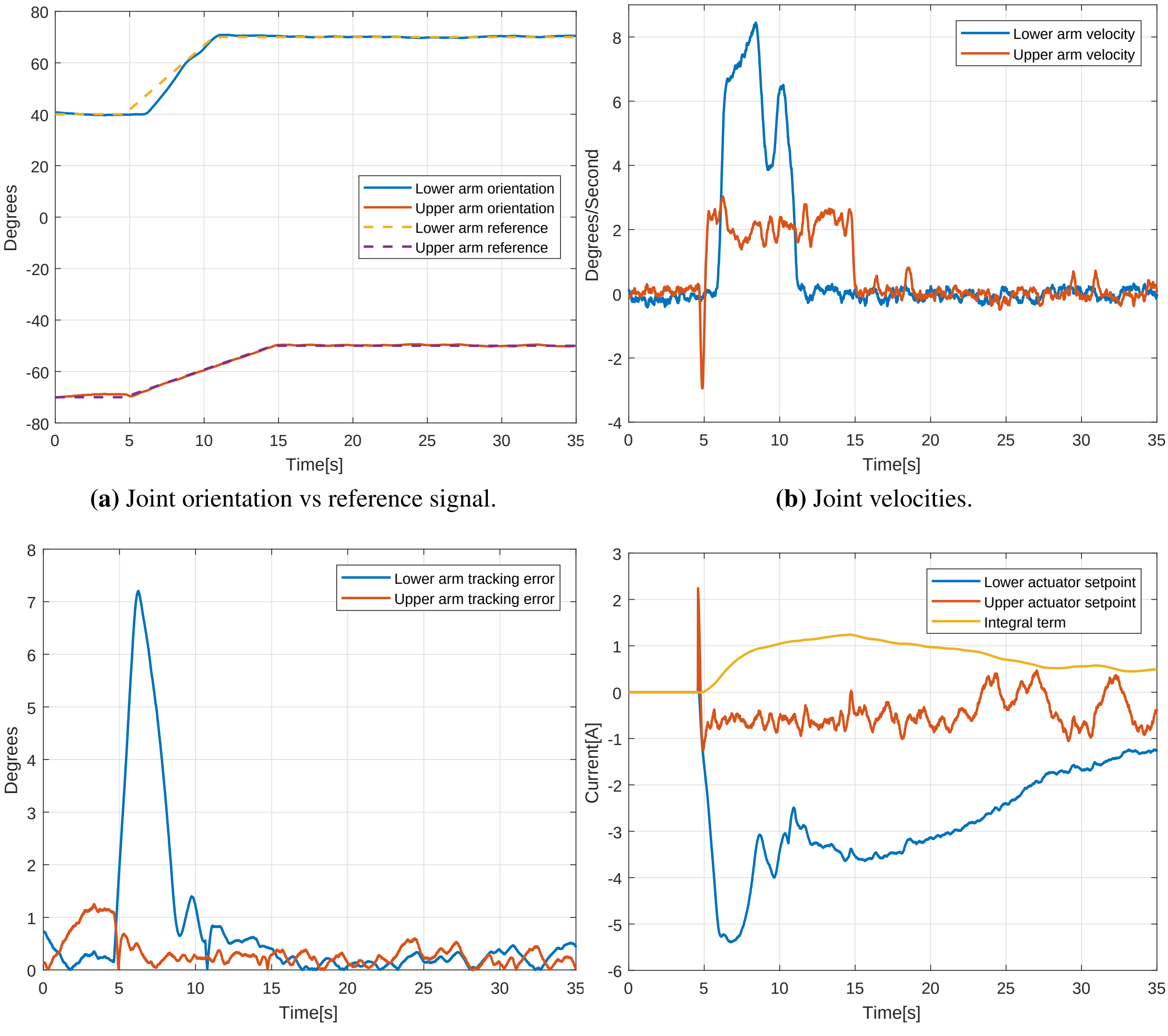
Parameter	Value	Unit
Wire Length	100	m
Wire Diameter	1.7	mm

6.2.1 No Load Tests

The first series of tests were run without an attached load. The objective was to verify the tuning parameters from Table 6.3 and the control law's ability to position the crane at a desired orientation. Three tests were run: lifting both the crane arms simultaneously, lowering the arms simultaneously and making the upper arm follow a sine shaped reference signal while holding the lower arm fixed.

Lifting

The purpose of the lifting test was to observe the controller's performance when positioning the crane arms at an orientation above their initial position. It was essential to observe that the integral term in the control law was able to remove the static deviations induced by gravity, within a reasonable period. A second motivation was to observe the response and behaviour of the actuators and compare their response to the simulations from Section 4.1.4.



(a) Joint orientation vs reference signal.

(b) Joint velocities.

(c) Absolute value of the tracking error.

(d) Motor current setpoints and integral state.

Figure 6.2: Lab results from lifting the crane arms with no load attached to the crane.

The results from the lifting test are presented in Figure 6.2.

The orientation of the lower arm and the reference signal are shown in Figure 6.2a and it can be observed that the lower arm managed to reach the final reference position, but was not accurately able to follow the desired path. The actuator motor needed to draw a current of about -5.5A, Figure 6.2d, in order to overcome static friction, which explains why the orientation lags behind the path at the five seconds mark. However, once static friction was overcome, the arm managed to catch up to the reference signal. The same result can also be observed from the tracking error, Figure 6.2c, that shows a large deviation at five seconds.

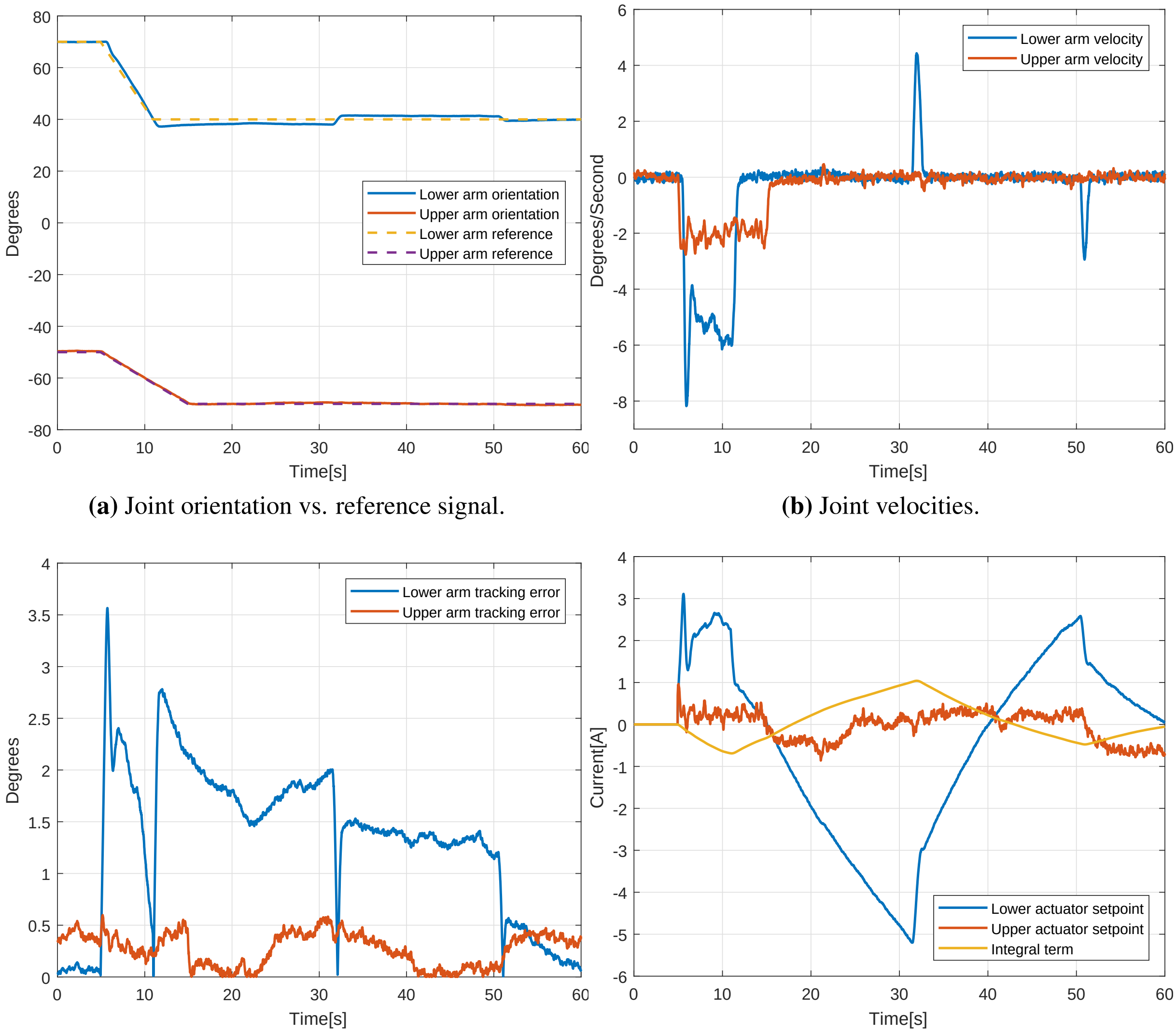
The upper arm was able to follow the reference signal without large errors. The largest tracking error was around $0.7°$ and can be observed in Figure 6.2c, right after the reference signal was initialized. The upper actuator did not need to draw the same amount of current as the lower, peaking at -1A in order to overcome static friction.

The noise characteristics of the sensors can be observed in the actuator current setpoint, Figure 6.2d, and the tracking error, Figure 6.2c. This is why the tracking error does not converge perfectly to zero.

In summary, the control law was able to lift both arms simultaneously, following a ramp reference signal. The lower arm lagged slightly behind due the amount of current needed to overcome static friction, but once moving it caught up with reference signal.

Lowering

The objective of the lowering test was to observe the control law's performance when positioning the crane arms at an orientation beneath their initial position. During this test the gravitational forces on the arms and actuators aided the motions of the system, meaning that the actuators should draw less current to overcome static friction. It was essential to observe that the orientation stabilized at the reference without significantly overshooting.



(a) Joint orientation vs. reference signal.

(b) Joint velocities.

(c) Absolute value of the tracking error.

(d) Motor current setpoints and integral term.

Figure 6.3: Lab results from lowering the crane arms with no load attached to the crane.

The results from the lowering test are presented in Figure 6.3.

When comparing the motion of the lower arm with the lifting test, it can be seen that static friction was overcome faster and by using lower current. This is because the gravitational force on the actuators aided the motion. However, it can also be observed that the lower actuator overshot the desired reference by $2.5°$ and that the integral term used about 20 seconds to correct the error, which again overshot the reference. The final correction can be seen at 50 seconds where lower arm finally reached the desired orientation

This result, together with the results from the lifting test, shows an aspect of the lower actuator that has not been included in the simulation model or control law. Namely, its friction profile, which is highly dependent upon which direction the actuator is moving. When being lowered a current of 3A is needed to overcome static friction, as opposed to around -5A when lifting.

This was most likely the reason why the integral term used 40 seconds to correct the deviation and the response can be made faster by increasing the integral gain. However, this could lead to larger overshoots. Another solution is to reset the integral term whenever it crosses the reference, thereby reducing the build-up time.

The upper actuator had no problems with following the reference signal and its tracking error is mostly caused by measurement noise.

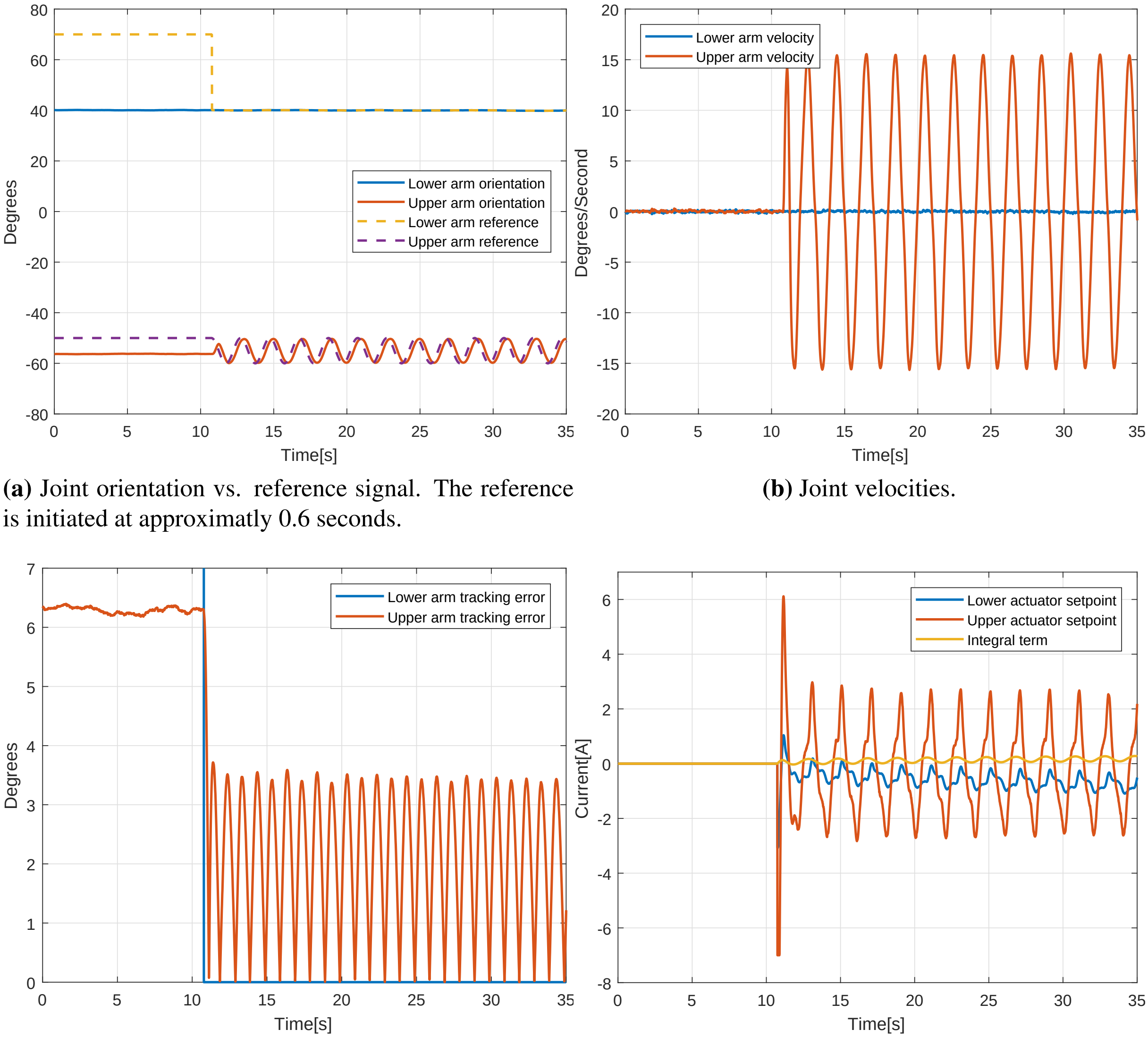
In summary, the control law was able to lower both arms simultaneously, following a ramp reference signal. The lower arm overshot the reference, but the integral term removed the deviation in about 50 seconds. The response can most likely by improving by retuning the control law.

Sine Tracking

In the last test, the upper arm was made to follow a sine shaped reference signal. The purpose was to test the controller with a wave-like reference signal that may be used when performing heave compensation. One of the design requirements of the crane, from (Gyberg, 2017), is that the crane tip should be able to follow a the motions induced by a scaled down sea state.

The reference signal was given by

$$\theta_{3,d} = -55 + 5\sin(\pi t) \tag{6.1}$$



(a) Joint orientation vs. reference signal. The reference is initiated at approximatly 0.6 seconds.

(b) Joint velocities.

(c) Error between joint orientation and reference.

(d) Motor current setpoints and integral term.

Figure 6.4: Lab results from performing heave compensation with the upper arm with no load attached to the crane.

The results of the sine tracking test is shown in Figure 6.4.

The lower arm was kept fixed for the duration of this test since the crane is design to perform heave compensation with the upper arm. The orientation of the upper arm is shown in Figure 6.2a and it can be observed that it managed to follow the reference signal with a small delay. Meaning that the response is slightly out of phase from the reference.

By comparing the actuator currents with the simulation of the same scenario, Figure 4.4d, it can be observed that larger currents were used in the physical system. This is because the self locking property of the actuators were not modelled in the simulation, meaning that the arm cannot be lowered by simply reducing the current. The current must be reversed in order to counter the self-locking property.

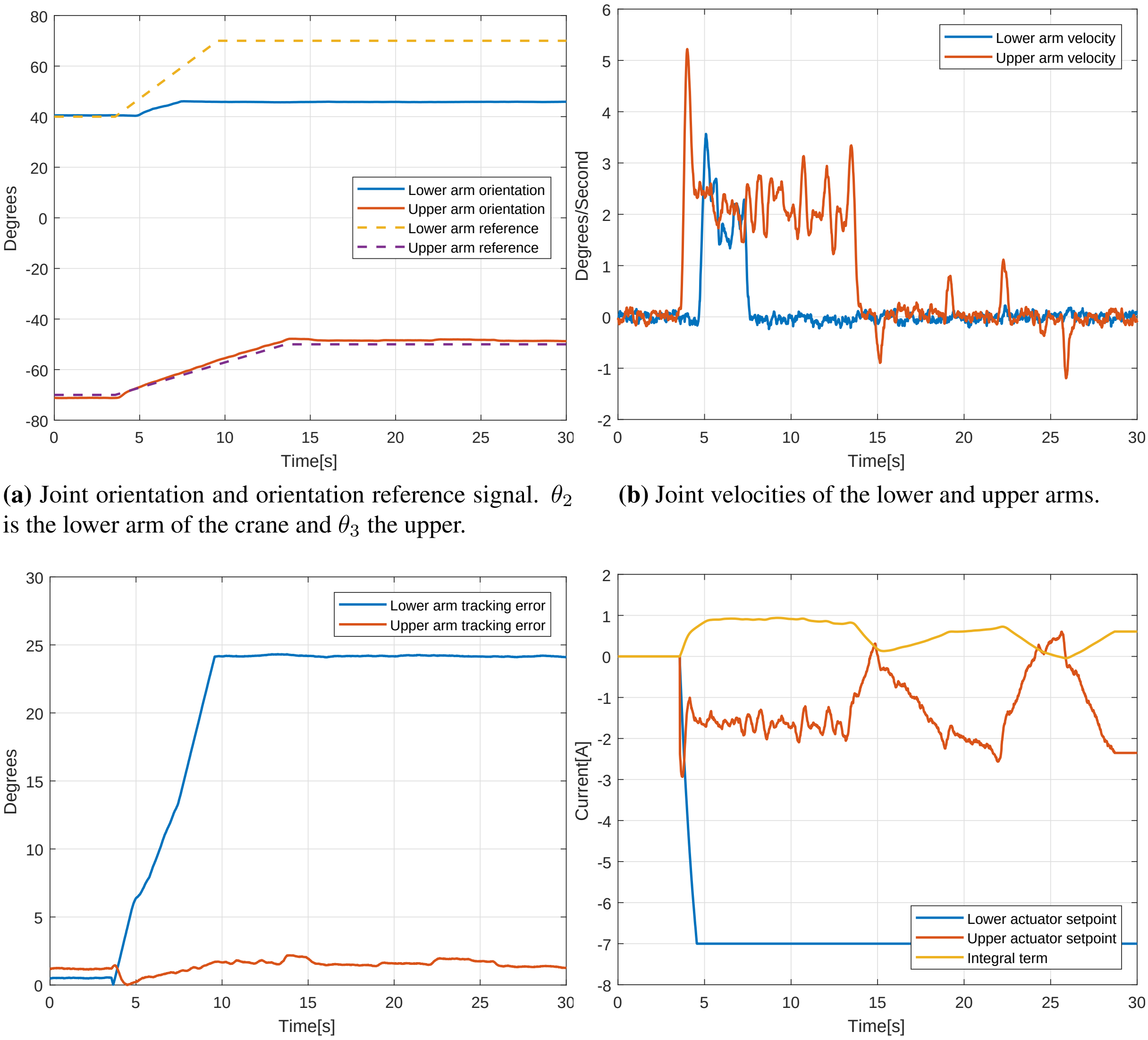
The tracking error, Figure 6.2c, is also larger than in the simulations, Figure 4.4d, because the arm lags behind the reference signal.

A a summary, the lowering, lifting and heave compensation tests without load has shown that the control law is able to position the crane at a desired reference, and to follow a shine shaped reference signal with some delay. It has been observed that the friction profile of the lower actuator is highly dependent upon its direction and that large currents are needed to lift the lower arm. In the next section, the same tests will be repeated with a load attached to the crane.

6.2.2 Tests With Load

The seconds series of tests were run with a load of 6kg attached to the crane and suspended 0.5m from the crane tip. The wire parameters are given in Table 6.4. The objective of these tests were to evaluate the control law's performance when the crane had to lift, lower or perform heave compensation with an attached load.

Lifting



(a) Joint orientation and orientation reference signal. θ_2 is the lower arm of the crane and θ_3 the upper.

(b) Joint velocities of the lower and upper arms.

(c) Error between joint orientation and reference.

(d) Motor current setpoints and integral term.

Figure 6.5: Lab results from lifting the crane arms with a load of 6kg attached to the crane.

Figure 6.5 shows the result of the lifting test with a 6kg attached to the crane.

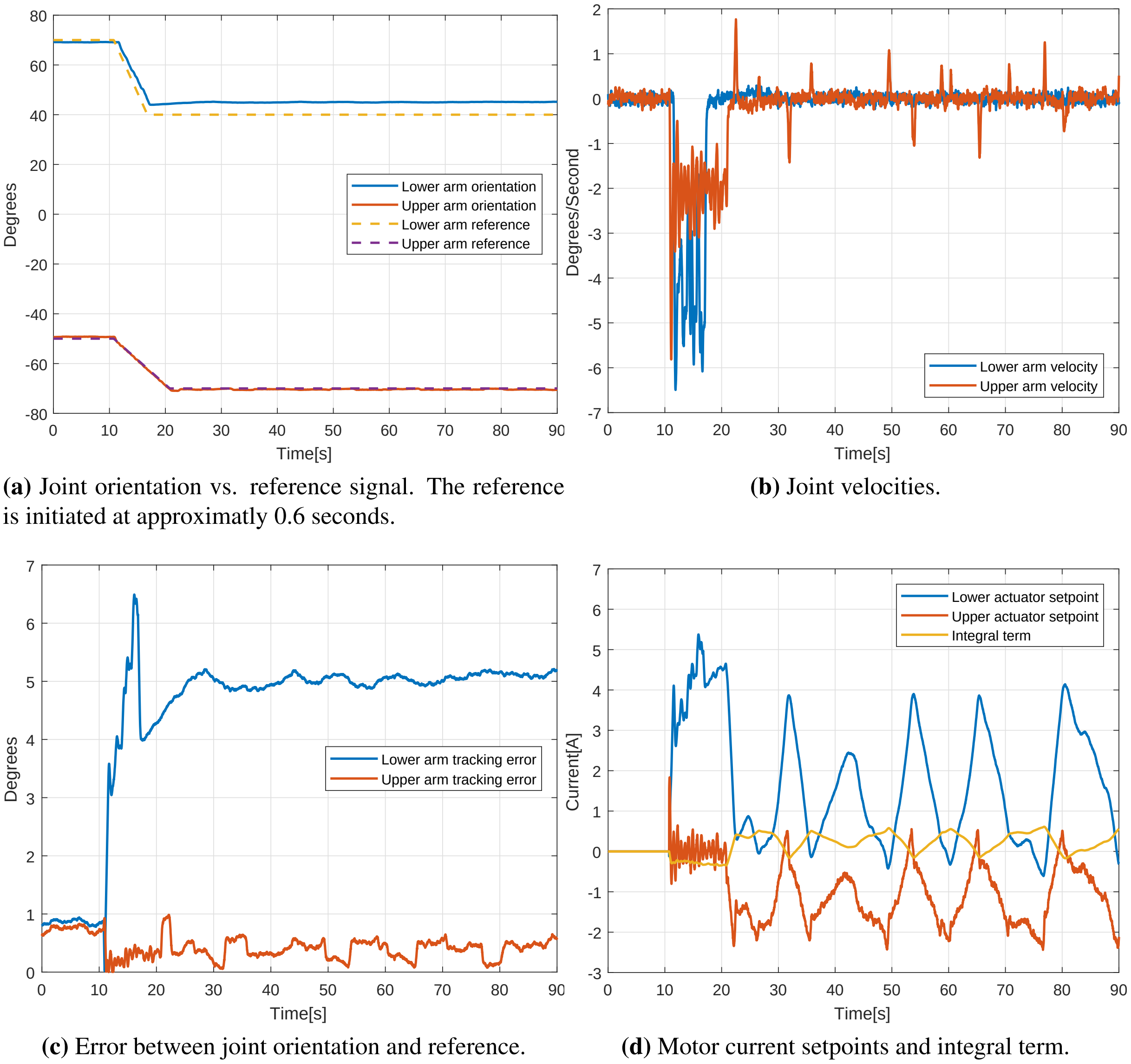
It can be observed from Figure 6.5a that the lower actuator was not able to follow the desired reference. The actuator was able to reach an orientation of 45° before stopping. Comparing with Figure 6.2d, showing motor currents, shows that the control law demanded maximum motor current -7A. This means that the electric motor was not able to provide enough force to overcome the static friction of the actuator.

The upper actuator had no problems with following the reference, but overshot the desired reference by approximately 2°. However, the integral term was able to remove the deviation on about 15 seconds.

As a summary, the lower actuator is not able to lift a load of 6kg when using a wire with a diameter of 1.7mm. The response can not be improved by tuning the control law since the current drawn by the lower motor reached its saturation value without moving the actuator. The cause may be either the friction profile of the actuator or the diameter of the wire and the tests should be run again with a thicker wire.

Lowering

The lifting test showed that the control law was not able to lift the crane arms with a load of 6kg. However, it is expected that it will be possible to lower the crane arms, since, in this case, the load will assist in overcoming static friction.



(a) Joint orientation vs. reference signal. The reference is initiated at approximatly 0.6 seconds.

(b) Joint velocities.

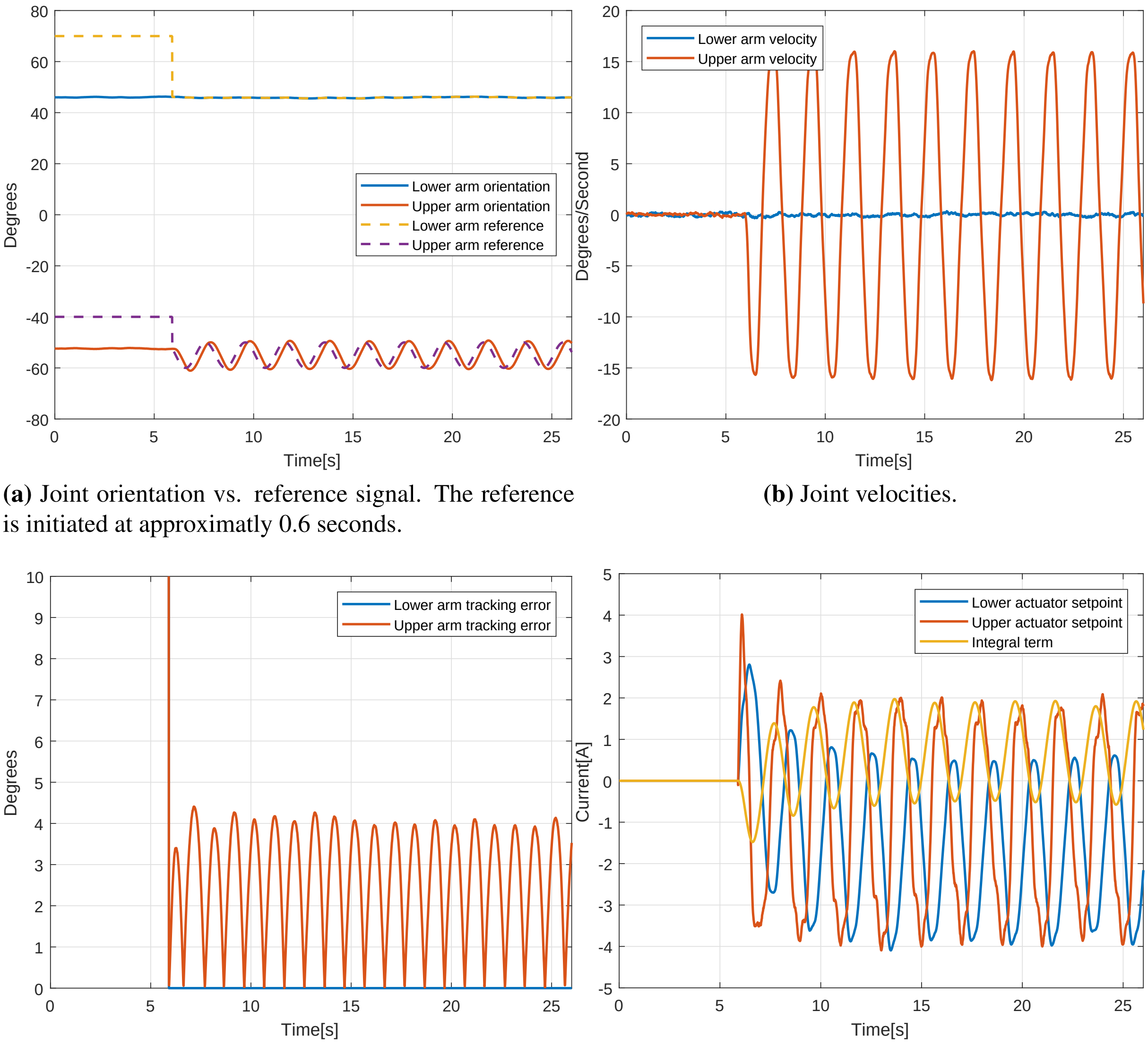
(c) Error between joint orientation and reference.

(d) Motor current setpoints and integral term.

Figure 6.6: Lab results from lowering the crane arms with a load of 6kg attached to the crane.

Sine Tracking

The last test evaluated to control law's performance when trying to perform heave compensation with the upper arm. The same reference signal as for the no load test (6.1) was used.



(a) Joint orientation vs. reference signal. The reference is initiated at approximatly 0.6 seconds.

(b) Joint velocities.

(c) Error between joint orientation and reference.

(d) Motor current setpoints and integral term.

Figure 6.7: Lab results from performing heave compensation with the upper arm with a load of 6kg attached to the crane.

Results from the sine tracking test are presented in Figure 6.7.

The results are similar to the sine tracking without the load, Figure 6.4. The orientation of the upper arm, Figure 6.7a, was able to follow the reference signal, slightly out of phase. The tracking error, Figure 6.5c, is slightly larger than the no load test, Figure 6.4c, peaking at $4 + ^\circ$.

It is unexpected that the control law outputs a sinusoidal current for the lower actuator, Figure 4.4d, since the lower arm should be fixed. However, the current do not exceed values necessary to overcome static friction, and the arm remains fixed. If the integral gain is increased, the lower arm will most likely start moving as well. This may hint to some errors the integral state, in the implementation of the control law or that the tuning could be improved. For instance, the gains for the lower arm may be increased to dampen the effect of the integral state.

In summary, the control law is able to perform heave compensation with the upper arm with a

6kg load. The response lags behind the reference signal, but the response can most likely be improved by better tuning.

6.3 Chapter Summary

This chapter has presented the results of testing the control law for positioning the crane joints, designed in Section 4.1, with the crane lab that was presented in Section 1.1 and Chapter 5. The tests made the crane follow a ramp signal for lifting and lowering the arms simultaneously and a sine shaped reference in order to test a heave compensation like scenario. The test were first run with no load attached to the crane and then with a weight of 6kg.

The no load tests revealed that the control law was able to lift and lower the crane arms and make the upper arm follow a sine shaped reference. The integration term was able to remove constant deviations caused by gravity, but used considerable time to reach a values able to lift the lower arm. This was most likely due to the friction profile of the lower actuator, it required a current of about -5.5A to overcome static friction during lifting operations.

The load tests revealed that the crane was not able to lift a load of 6kg with a wire with diameter 1.7mm. The control law saturated the current of the lower actuator without being able to move the arm. In addition, when lowering the arms, a significant increase in the gyrscope noise was observed. The noise was especially prominent for the lower arm gyroscope and may have been induced by the load's pendulum motions or by vibrations in the crane's foundation. However, since the amplified noise was only observed when lowering the arm with an attached load, it is most likely due to pendulum motions.

Chapter 7

Conclusion and Further Work

This thesis has presented the derivation of simulation models for a knuckleboom crane and its winch system. The simulation models have been used to design control laws for positioning the orientation of the crane joints and for softly landing its cargo. The control law for positioning the crane has been implemented and tested in a new crane lab at the department of marine technology. As part of this process the lab has been improved with a new set of sensors, filters and hands-on control systems. Due to delays in the lab, it has not been time to test the soft landing controller.

The crane simulation model has been based on the physical crane in the lab and was derived using Lagrangian mechanics and modelling procedures for robotic manipulators developed by (Spong et al., 2006). The model was derived by treating the crane as a system of linked rigid bodies and includes the weight of the actuators, but not their dynamics. In addition, friction in the crane joints have been assumed to be negligible. The load of the crane has been modelled as a weight acting on the tip of the crane and is assumed constant during crane operations.

A control law for positioning the base, lower and upper arms of the knuckleboom crane has been designed. It has been based upon the simulation model of the crane and designed by using the non-linear, recursive backstepping method. The method has produced a global asymptotically stable control law with integral action. Simulating the control law with the crane dynamics has shown that it was able to stabilize the model to a desired reference. The simulations were run with 16kg attached to the crane tip and consisted of following a ramp reference signal for lifting and lowering the arms and performing heave compensation by following a sine shaped reference.

The control law has been implemented on the PLC in the crane lab and tested with the physical knuckleboom crane. Two series of tests have been run, with and without a load attached to the crane tip. Both series consisted of a lowering, lifting and heave compensation like test. During the lowering and lifting tests, the crane was made to follow a ramp signal and for the heave compensation, a sine shaped reference signal.

The tests without load showed that the crane was able to follow the lowering and lifting ramp signals and the sine shaped reference. The integral term was able to remove deviations caused by gravity when lifting the crane arms. When lowering the arms, it caused the lower actuator to overshoot the desired reference by approximately 2°, and it used about 40 seconds to completely remove the deviation. During the heave compensation like test, the lower arm was kept fixed, and the upper managed to follow the sine reference, with a slight delay.

The response of the no-load tests can most likely be improved by finding better tuning parameters for the control law. However, the deviation that was experienced during lowering was most likely caused by the dynamics of the lower actuator. The lower motor must draw currents of about 5A in order to overcome static friction. This means that the integral term had to reverse after overshooting the reference. Of course, the integral tuning constant can be increased to decrease the recovery time, but this could lead to larger overshoots. An improvement, could be to reset the integral term whenever the lower arm orientation crosses the reference.

Lab tests with an attached load revealed that the lower actuator was not able to lift a 6kg weight, using a wire with a diameter of 1.7mm. The motor drew maximum current, 7A, but was not able to overcome the static friction of the actuator. It was possible to lift the arm slightly by manually controlling the crane and initialising a downwards motion that was quickly reversed. Thereby starting the lifting operation with dynamic friction, however, the motion quickly slowed down and the procedure had to be repeated to further lift the arm.

There may be many reasons for the lower actuator's incapability of lifting a load. The result was unexpected since the crane has been designed for loads up to 16kg and should therefore be able to easily lift 6kg. It could be that the small wire diameter caused extra load on the lower arm, thereby increasing its resistance. It could also be that the actuator dynamics are significant enough to be included in the simulation model. The upper actuator did not display any problems with either lifting or lowering the attached load. It managed to follow both the lifting and lowering reference signals with good accuracy.

In addition, gyroscopes and accelerometers have been added to the marine crane lab and the winch system has been completed. A simple hands on control for the winch has been implemented, where the winch speed is controlled by a joystick. A Kalman and low-pass filter has been designed and tuned in order to remove noise from the sensor measurements and estimate gyro bias. The filters provided good performance when being used with the control laws.

However, pendulum motions of the load were induced when lowering both arms simultaneously. The motions caused excessive noise in the gyroscope measurements, especially for the gyro placed at the lower arm. This means that the filters are vulnerable to vibrations caused by the load or other external sources. The crane is currently placed on a concrete block with rubber feet, which may also have contributed to vibrations during the operations. However, the excessive noise was only observed with an attached load.

The winch has been modelled as an electric DC motor, whose mechanical side is modelled at the winch reel. The equations developed by (Skjong. and Pedersen, 2014) has been used to estimate the diameter and inertia of the winch reel, since these parameters are functions of the amount of wire on the reel.

The winch model has been used to design a control law for softly landing the crane's cargo. Soft landing has been defined as a tracking problem, as developed by (Skjetne et al., 2004), where the control objective is to make the system output follow a defined path and velocity profile. The definition of the velocity profile is the essence of the soft landing problem as it will determine the velocity of the load. It has been chosen such that the load is quickly brought to full speed until a certain point in the path is reached, then a slow velocity phase is initiated.

Simulation results of the soft landing control law has shown that the winch model is able to follow the defined velocity profile with a maximum speed of $0.5 \mathrm{m\,s^{-1}}$, while following a straight line path. However, due to delays, it has not been time to implement and test the control law in the crane lab.

Delays have been caused by unforeseen problems with the lab. The problems have occurred since the lab is currently in a prototype phase and at the start of this thesis, there were no infrastructure for implementing and testing control laws. This meant that most functions for filtering and processing the PLC's I/O had to be written and that a lot of time was used to test whether the implemented functions behaved as intended.

Other delays were caused by already existing sensors and equipment. For instance, the stop sensors placed at the ends of both actuators were not reliable during the experiments. Several times the actuators ran past the stop position, crashing into the end piece of the actuator housing. Therefore, a lot of time was used to open the actuators and manually screw them back into position. Sometimes the sensors did not register the actuator screw and did therefore not provide a stop signal. The unreliability could be due to some programming errors in the controller code, but this only explains the cases when the actuators ran past the sensors even when the stop signal was registered.

However, the delays and challenges have provided valuable experience and provided new areas of improvement of the lab.

7.1 Further Work

It is recommended that a model study of the actuators are undertaken. Especially the friction profile of the lower actuator should be determined. This will make it possible to construct an accurate dynamic model of the actuators, which can be added to the simulation model of the crane and included in the position control law. Tests should be run with a thicker wire to determine whether the wire diameter caused the poor response of the lower actuator.

The soft landing controller should be implemented and tested in the lab. The simulation results are promising, however, a parameter study of the lab winch should be performed to more accurately determine its friction, transmission and weight properties. After the soft landing controller has been tested an effort should be made to combine the crane and winch dynamics to try and achieve soft landing by controlling the winch and crane arms simultaneously.

If the crane is to be placed on a barge and tested in the ocean basin, a simulation model estimating interconnect crane-vessel dynamics should be developed. An approach for developing such models can be found in (Rokseth et al., 2016).

When preparing the lab for the experiments, the Bachmann controller ran out of I/O channels. Therefore, additional modules should be procured in order to accommodate more sensors and electronics. As of now, the complete winch system cannot be run simultaneously with the crane position controller. Which does not include the gyroscope and accelerometer placed at the base. In addition, the gyroscope that was defective upon delivery should be replaced and fitted to the crane base.

It is suggested that optical, rotary encoders are installed in the crane joints in order to measure absolute angular position. This will provide an extension to the lab's sensors and allow for more accurate filtering and sensor fusions.

The stop sensors should be more permanently integrated in the actuator housing, as opposed to being glued as they were during the work of this thesis. It is also suggested that some mechanical damping mechanism is installed in the actuators ends, in order to act as an extra safeguard should the stop sensor fail. This will stop the actuators from jamming if they run past the stop sensors.

7.2 Conclusion

In conclusion, most of the goals established in Section 1.3 has been accomplished. Simulation models of the knuckleboom crane and winch in the marine crane lab has been derived. Control laws for positioning the joints orientation of the crane and softly landing its cargo has been designed and proven stable. The position control law has been tested in the lab and was able to position an unloaded crane. Tests with an attached load has shown that the actuator dynamics and parameters warrants further study. The proposed extensions to the lab has been done by adding additional sensors, designing sensor filter and finishing the implementation of the winch. The filters have shown to perform well for the experiments that were done. Due to delays, it has not been time to test the soft landing controller in the lab.

Bibliography

Alibeji, N., Sharma, N., 2 2017. A pid-type robust input delay compensation method for uncertain euler-lagrange systems. IEEE Transactions on Control Systems Technology.

Bø, T. I., Dhal, A. R., et al., 10 2015. Marine vessel and power plant system simulator. IEEE Access.

Bachmann, 2017. Bachmann products. `https://www.bachmann.info/en/products/controller-system/`, accessed: 2017-27-06.

Balchen, J. G., Andresen, T., Foss, B. A., 2003. Reguleringsteknikk. Institutt for teknisk kybernetikk, NTNU.

Commons, W., 2017. Capstan equation. `https://upload.wikimedia.org/wikipedia/commons/thumb/5/56/Capstan_equation_diagram.svg/220px-Capstan_equation_diagram.svg.png`.

Contrinex, 2017. `https://www.contrinex.com/product/dw-ad-623-04/`, accessed: 2017-27-06.

Devices, A., 2017. Adxrs624. `http://www.analog.com/media/en/technical-documentation/data-sheets/ADXRS624.pdf`, accessed: 2017-27-06.

Fang, Y., Wang, P., Sun, N., Zhang, Y., 1 2016. Dynamics analysis and nonlinear control of an offshore boom crane. IEEE Transactions on Industrial Electronics.

Ginsberg, J. H., 2010. Advanced Engineering Dyanmics. Cambridge University Press.

Gyberg, F., 1 2017. Design, modelling and control of a generic crane for marine applications. Master's thesis, Norwegian University of Science and Technology, Høgskoleringen 1, 7491 Trondheim.

Khalil, H. K., 2014. Nonlinear Control. Pearson.

Mathworks, 2017. Symbolic math toolbox. `https://se.mathworks.com/products/symbolic.html`, accessed: 2017-15-06.

Memsic, 2017. Mxr9500g/m. `https://www.memsic.com/userfiles/files/Datasheets/Accelerometer-Datasheets/MXR9500GM__RevD.pdf`, accessed: 2017-07-007.

Qian, Y., Fang, Y., 5 2016. Dynamics analysis of an offshore ship-mounted crane subject to sea wave disturbances. Intelligent Control and Automation,.

Rokseth, B., 2014. A bond graph approach for modelling systems of rigid bodies in spatial motion. Master's thesis, Norwegian University of Science and Technology.

Rokseth, B., Skjong, S., Pedersen, E., 2016. Modeling of generic offshore vessel in crane operations with focus on strong rigid body connections. IEEE Journal of Oceanic Engineering.

Scilab, 2017. Nonlinear control of 2-link manipulator. `http://www.scilab.ninja/images/scilab/nonlinear2link/fig1_1.jpg`, accessed: 2017-09-07.

Skjetne, R., Fossen, T. I., June 2004. On integral control in backstepping: Analysis of different techniques. In: Procc. of the American Control Conference. pp. 1899–1904.

Skjetne, R., Fossen, T. I., Kokotović, P. V., 3 2004. Robust output maneuvering for a class of nonlinear systems. Automatica.

Skjong., S., Pedersen, E., 2014. Modeling hydraulic winch system. simulation series. 2014 International Conference on Bond Graph Modeling and Simulation - ICBGM'2014.

Smith, A., Yang, C., Li, C., Ma, H., Zhao, L., 9 2016. Development of a dynamics model for the baxter robot. IEEE International Conference on Mechatronics and Automation,.

Spong, M. W., Hutchinson, S., Vidyasagar, M., 2006. Robot Modeling and Control. Wiley & Sons inc.

Appendix A

Matlab Source Code

This appendix presents the source code of the Matlab scripts that were used to generate and initialize the simulations models of the knuckleboom crane, winch, position control law and soft landing control law.

A.1 Crane Simulation Model and Control Law

```
clear
%% Defining Symbolic Variables
syms t g
syms theta1(t) theta2(t) theta3(t)
syms q1 q2 q3
syms dq1 dq2 dq3
syms ddq1 ddq2 ddq3

syms a b c h r s u w o p n k zcm1
syms L1 L2

syms I1x I1y I1z I2x I2y I2z I3x I3y I3z
syms I4x I4y I4z I5x I5y I5z

syms m1 m2 m3 m4 m5 mLoad

syms Fa1 Fa2

syms e1 e2 e3
syms q1_ref q2_ref q3_ref

q = [q1; q2; q3];
dq = [dq1; dq2; dq3];
```

```
I1 = [I1x 0 0; 0 I1y 0; 0 0 I1z];
I2 = [I2x 0 0; 0 I2y 0; 0 0 I2z];
I3 = [I3x 0 0; 0 I3y 0; 0 0 I3z];
I4 = [I4x 0 0; 0 I4y 0; 0 0 I4z];
I5 = [I5x 0 0; 0 I5y 0; 0 0 I5z];

%% Actuator angles
d1 = asin((h + sin(q2)*b -cos(q2)*u)/(sqrt((cos(q2)*b + sin(q2)
   *u -a)^2 +(h + sin(q2)*b - cos(q2)*u)^2)));


my2 = pi + q3 - asin(s/(sqrt(c^2 + s^2))) - asin(r/(sqrt(r^2 +
   (L1 - w)^2)));
v = sqrt(r^2 + (L1 - w)^2);
cm = sqrt(c^2 + s^2);

d2 = pi + q2 + q3 - asin(s/cm) - acos((cos(my2)*v - cm)/(sqrt(v
   ^2 + cm^2 - 2*v*cm*cos(my2))));

%% Transformation matrices
Rz = [cos(q1) -sin(q1) 0;...
      sin(q1) cos(q1) 0;
      0 0 1];

Ry1 = [cos(q2) 0 sin(q2);...
        0 1 0;...
        -sin(q2) 0 cos(q2)];

Ry2 = [cos(q3) 0 sin(q3);...
        0 1 0;...
        -sin(q3) 0 cos(q3)];

Ra1 = [cos(d1) 0 sin(d1);...
    0 1 0;...
    -sin(d1) 0 cos(d1)];

Ra2 = [cos(d2) 0 sin(d2);...
    0 1 0;...
    -sin(d2) 0 cos(d2)];

R1_0 = Rz;
R2_0 = Ry1*Rz;
R3_0 = Ry2*Ry1*Rz;
R3_0 = simplify(R3_0);
R4_0 = Ra1*Rz;
R5_0 = Ra2*Rz;
```

```
R0_1 = transpose(R1_0);
R0_2 = transpose(R2_0);
R0_3 = transpose(R3_0);
R0_4 = transpose(R4_0);
R0_5 = transpose(R5_0);

%% Inertia and Mass Matrices
I1 = R0_1*I1*transpose(R0_1);
I2 = R0_2*I2*transpose(R0_2);
I3 = R0_3*I3*transpose(R0_3);
I4 = R0_4*I4*transpose(R0_4);
I5 = R0_5*I5*transpose(R0_5);

M1 = blkdiag(eye(3)*m1,I1);
M2 = blkdiag(eye(3)*m2,I2);
M3 = blkdiag(eye(3)*m3,I3);
M4 = blkdiag(eye(3)*m4,I4);
M5 = blkdiag(eye(3)*m5,I5);

%% Position of Reference Frames and Mass Centers
r1_0 = R0_1*[0;0;0];
r2_0 = r1_0 + R0_1*[0;0;h];
r3_0 = r2_0 + R0_2*[L1;0;0];
r4_0 = r1_0 + R0_1*[a;0;0];
r5_0 = r2_0 + R0_2*[w;0;-r];

rcg1_0 = r1_0 + R0_1*[0;0;zcm1];
rcg2_0 = r2_0 + R0_2*[k;0;0];
rcg3_0 = r3_0 + R0_3*[n;0;0];
rcg4_0 = r4_0 + R0_4*[p;0;0];
rcg5_0 = r5_0 + R0_5*[o;0;0];

rtp_0 = r3_0 + R0_3*[L2;0;0];


%% Angular Velocities
w1 = R0_1*[0;0;1];
w2 = R0_2*[0;1;0];
w3 = R0_3*[0;1;0];

%% Jacobians
J1 = [cross(w1,rcg1_0), zeros(3,2); w1 zeros(3,2)];
J2 = [cross(w1,rcg2_0), cross(w2, rcg2_0 - r2_0), zeros(3,1);
   w1 w2 zeros(3,1)];
J3 = [cross(w1,rcg3_0), cross(w2, rcg3_0 - r2_0), cross(w3,
   rcg3_0 - r3_0); w1 w2 w3];
```

```
J4 = [cross(w1,rcg4_0), cross(w2, rcg4_0 - r2_0), zeros(3,1);
   w1 w2 zeros(3,1)];
J5 = [cross(w1,rcg5_0), cross(w2, rcg5_0 - r2_0), cross(w3,
   rcg5_0 - r3_0); w1 w2 w3];

%% Generalized Load
F_load = [0;0;-g*mLoad];
r_load = r3_0 + R0_3*[L2;0;0];
Q_load = [transpose(F_load)*diff(r_load,q1); transpose(F_load)*
   diff(r_load,q2); transpose(F_load)*diff(r_load,q3)];


%% Actuator Forces
F_a1 = R0_1*Fa1*[cos(d1);0;sin(d1)];
F_a2 = R0_1*Fa2*[cos(d2);0;sin(d2)];


r_a11 = r4_0;
r_a12 = r2_0 + R0_2*[b;0;-u];

r_a21 = r2_0 + R0_2*[w;0;-r];
r_a22 = r3_0 + R0_3*[c;0;-s];

Q_a11 = [transpose(F_a1)*diff(r_a11,q1); transpose(F_a1)*diff(
   r_a11,q2); transpose(F_a1)*diff(r_a11,q3)];
Q_a12 = [transpose(F_a1)*diff(r_a12,q1); transpose(F_a1)*diff(
   r_a12,q2); transpose(F_a1)*diff(r_a12,q3)];

Q_a21 = [transpose(F_a2)*diff(r_a21,q1);transpose(F_a2)*diff(
   r_a21,q2);transpose(F_a2)*diff(r_a21,q3)];
Q_a22 = [transpose(F_a2)*diff(r_a22,q1);transpose(F_a2)*diff(
   r_a22,q2);transpose(F_a2)*diff(r_a22,q3)];

%% Generalized Inertia, Coriolis and Gravity Matrices
D = transpose(J1)*M1*J1 + transpose(J2)*M2*J2 + transpose(J3)*
   M3*J3 + transpose(J4)*M4*J4 + transpose(J5)*M5*J5;
D = simplify(D);

C = coriolisMatrix(D,q,dq);

V = -g*m1*rcg1_0(3) - g*m2*rcg2_0(3) - g*m3*rcg3_0(3) - g*m4*
   rcg4_0(3) - g*m5*rcg5_0(3);
V = simplify(V);

G = -[diff(V,q1); diff(V,q2); diff(V,q3)];

%% Numerical Parameter Values
```

```
m_num = [67, 23.3, 5.5, 8.5, 6.8]; %m1 m2 m3 m4 m5
I1_num = [0, 0, 1.8];
I2_num = [0 9.2 9.4];
I3_num = [0, 1.6, 1.5];
I4_num = [0, 2.3, 2.3];
I5_num = [0, 1.1, 1.1];

g_num = 9.81;

arm_length = [1.251, 0.96]; %L1 L2
dimensions = [0.461, 0.1, 0.495, 0.461, 0.19, 0.140, 0.140,
   0.118, 0.533, 0.420, 0.324, 0.440, −0.023]; % b,a,h,w,c,u,r,
   s,k,n,o,p, zcm1

%% Inserting Numerical Values
D = subs(D, [m1 m2 m3 m4 m5],m_num);
D = subs(D, [I1x,I1y,I1z,I2x,I2y,I2z,I3x,I3y,I3z,I4x,I4y,I4z,
   I5x,I5y,I5z],[I1_num,I2_num,I3_num,I4_num,I5_num]);
D = subs(D,[b,a,h,w,c,u,r,s,k,n,o,p,zcm1],dimensions);
D = subs(D,[L1,L2],arm_length);


C = subs(C, [m1 m2 m3 m4 m5],m_num);
C = subs(C, [I1x,I1y,I1z,I2x,I2y,I2z,I3x,I3y,I3z,I4x,I4y,I4z,
   I5x,I5y,I5z],[I1_num,I2_num,I3_num,I4_num,I5_num]);
C = subs(C,[b,a,h,w,c,u,r,s,k,n,o,p,zcm1],dimensions);
C = subs(C,[L1,L2],arm_length);

G = subs(G, [m1 m2 m3 m4 m5],m_num);
G = subs(G, [I1x,I1y,I1z,I2x,I2y,I2z,I3x,I3y,I3z,I4x,I4y,I4z,
   I5x,I5y,I5z],[I1_num,I2_num,I3_num,I4_num,I5_num]);
G = subs(G,[b,a,h,w,c,u,r,s,k,n,o,p,zcm1],dimensions);
G = subs(G,[L1,L2],arm_length);


V = subs(V, [m1 m2 m3 m4 m5],m_num);
V = subs(V, [I1x,I1y,I1z,I2x,I2y,I2z,I3x,I3y,I3z,I4x,I4y,I4z,
   I5x,I5y,I5z],[I1_num,I2_num,I3_num,I4_num,I5_num]);
V = subs(V,[b,a,h,w,c,u,r,s,k,n,o,p,zcm1],dimensions);
V = subs(V,[L1,L2],arm_length);
V = subs(V, g, g_num);

rtp_0 = subs(rtp_0,[L1,L2,h],[arm_length 0.495]);

Q_load = subs(Q_load,[L1,L2,h], [arm_length,0.495]);

Q_a11 = subs(Q_a11,[L1,L2],arm_length);
```

```
Q_a11 = subs(Q_a11, [b,a,h,w,c,u,r,s,k,n,o,p,zcm1],dimensions);

Q_a12 = subs(Q_a12,[L1,L2],arm_length);
Q_a12 = subs(Q_a12, [b,a,h,w,c,u,r,s,k,n,o,p,zcm1],dimensions);

Q_a21 = subs(Q_a21,[L1,L2],arm_length);
Q_a21 = subs(Q_a21, [b,a,h,w,c,u,r,s,k,n,o,p,zcm1],dimensions);

Q_a22 = subs(Q_a22,[L1,L2],arm_length);
Q_a22 = subs(Q_a22, [b,a,h,w,c,u,r,s,k,n,o,p,zcm1],dimensions);

Q_a1 = -Q_a11 + Q_a12;
Q_a2 = -Q_a21 + Q_a22;

Q = [[1; zeros(2,1)] diff(Q_a1,Fa1), diff(Q_a2,Fa2)];

%% Backstepping Controller
syms qd_1 qd_2 qd_3
syms p11 p12 p13
syms p21 p22 p23
syms a11 a12 a13
syms a21 a22 a23
syms xi
syms K_i
P_1 = diag([p11; p12; p13]);
P_2 = diag([p21; p22; p23]);
A_1 = diag([a11; a12; a13]);
A_2 = diag([a21; a22; a23]);

q_d = [qd_1; qd_2; qd_3];

z_1 = q - q_d;

alpha_1 = A_1*z_1;

z_2 = dq - alpha_1;
alpha_1_dot = A_1*dq;

D_inv = D\eye(3);

u_0 = (D_inv*C*(alpha_1 +z_2) + alpha_1_dot - inv(P_2)*P_1*z_1
    + A_2*z_2);

xi_dot = transpose(D_inv*diff((Q_load - G) ,g))*transpose(
    transpose(z_2)*P_2);
u = (Q\eye(3))*(D*u_0 - diff((Q_load - G),g)*K_i*xi);
```

```
G = subs(G, g, g_num);
Q_load = subs(Q_load,g,g_num);

%% Simulink Function Blocks
matlabFunctionBlock('Crane/Backstepping_Controller/u_1',u);
matlabFunctionBlock('Crane/Backstepping_Controller/xi_dot',
    xi_dot);

% System Dynamics
matlabFunctionBlock('Crane/Crane_Dynamics/M',D);
matlabFunctionBlock('Crane/Crane_Dynamics/C',C);
matlabFunctionBlock('Crane/Crane_Dynamics/G',G);
matlabFunctionBlock('Crane/Tip_motion/tip', rtp_0);

% Actuator Connections
matlabFunctionBlock('Crane/Load/Q_load',Q_load);
matlabFunctionBlock('Crane/Actuator_1/Q_a1',Q_a1);
matlabFunctionBlock('Crane/Actuator_2/Q_a2',Q_a2);
```

A.2 Crane Simulation Model Initialization Script

```
clear

%% Joint Angle Limitations and Initial Conditions
q_lower_lim = [-inf; 30.2; -90.5].*pi/180;
q_upper_lim = [inf; 86; -24.4].*pi/180;

q_0 = [0;45*pi/180; -45*pi/180];


%% Tuning Parameters
Ki = 400;
mLoad = 16;

P_1 = eye(3)*0.01;
P_2 = eye(3)*0.01;
A_1 = lyap(P_1,eye(3));
A_2 = lyap(P_2,eye(3));

p11 = P_1(1,1);
p12 = P_1(2,2);
p13 = P_1(3,3);

p21 = P_2(1,1);
p22 = P_2(2,2);
p23 = P_2(3,3);


a11 = A_1(1,1);
a12 = A_1(2,2);
a13 = A_1(3,3);

a21 = A_2(1,1);
a22 = A_2(2,2);
a23 = A_2(3,3);

%% Parameters for estimating motor current
eta_1 = (13/74)*1/(186*0.86);
eta_2 = 0.01/(2*pi)*1/(26*0.83);
eta_3 = 1/(15.7*0.83)*0.01/(2*pi);

Kt_1 = 60.4e-3;
Kt_2 = 60.4e-3;
Kt_3 = 60.4e-3;
```

A.3 Winch Simulation Model and Control Law

```
clear

syms L L_dot i V
syms Dw wr fw Dr0
syms mreel mflange mL Drfull rhow
syms Kt etat g phi1 phi2 phi3 etaw Kv La Ra

%%Defining Winch Model
D = sqrt(L*Dw^2/(wr*fw) + Dr0^2);
DL = diff(D,L);

I = 0.5*mreel*Dr0^2/4 + mflange*Drfull^2/4 + 0.5*rhow*L*(D^2 -
   Dr0^2)/4;

L_ddot = (DL/D)*L_dot^2 + (D/(2*I))*Kt*etat*i - D^2/(4*I)*g*mL/
   exp(phi1 + phi2 + phi3);
i_dot = -etaw*Kv*D*L_dot/(La*2) - Ra*i/La + V/La;

%% Soft Landing Controller
syms zs Lt Larms
syms s s_dot(s) bp Ld(s) Lds Ldss
syms a1 a2 a3
syms p1 p2 p3
syms Lds

z1 = L - Ld;


alpha1 = diff(Ld,s)*s_dot + a1*z1;
alpha1_dot = diff(alpha1,L)*L_dot + diff(alpha1,s)*s_dot;


z2 = L_dot - alpha1;
alpha2 = (2*I/(D))*(a2*z2 - z2*p1/p2 - DL/(D*Kt*etat)*(z2 +
   alpha1)^2 + alpha1_dot);

alpha2_dot = diff(alpha2,L)*L_dot + diff(alpha2,L_dot)*L_ddot +
     diff(alpha2,s)*s_dot;

z3 = i - alpha2;

V = La*((Ra/La)*(z3 + alpha2) + etaw*Kv*D/(2*La)*(z2 + alpha1)
   - z2*p2*(D/(2*I*p3))*Kt*etat + a3*z3 + alpha2_dot);
```

```
%% Nunmerical Values
Kt_num = 53.4e-3;
Kv_num = 2*pi/(179*60);
Ra_num = 0.307;
La_num = 0.188e-3;

etat_num = 2*43;
etaw_num = 1/(2*43);
Dr0_num = 0.04;
Drfull_num = 0.1;

mreel_num = 1;
mflange_num = 1;

fw_num = 0.9096;
wr_num = 0.13;
Dw_num = 0.0017;
rhow_num = 100;

g_num = 9.81;
phi1_num = pi/4;
phi2_num = pi/4;
phi3_num = pi/4;

%% Substituting Numerical Values
D = subs(D, [Dr0, Drfull, etat, etaw], [Dr0_num, Drfull_num,
   etat_num, etaw_num]);
D = subs(D, [fw wr, Dw], [fw_num, wr_num, Dw_num]);

I = subs(I, [Dr0, Drfull, etat, etaw], [Dr0_num, Drfull_num,
   etat_num, etaw_num]);
I = subs(I, [fw wr, Dw], [fw_num, wr_num, Dw_num]);
I = subs(I, [mreel, mflange, rhow], [mreel_num, mflange_num,
   rhow_num]);

L_ddot = subs(L_ddot, [Dr0, Drfull, etat, etaw], [Dr0_num,
   Drfull_num, etat_num, etaw_num]);
L_ddot = subs(L_ddot, [fw wr, Dw], [fw_num, wr_num, Dw_num]);
L_ddot = subs(L_ddot, [mreel, mflange, rhow], [mreel_num,
   mflange_num, rhow_num]);
L_ddot = subs(L_ddot, [Kt, Kv, Ra, La], [Kt_num, Kv_num, Ra_num
   , La_num]);
L_ddot = subs(L_ddot, [g, phi1, phi2, phi3],[g_num, phi1_num,
   phi2_num, phi3_num]);
L_ddot = simplify(L_ddot);
```

```
i_dot = subs(i_dot, [Dr0, Drfull, etat, etaw], [Dr0_num,
    Drfull_num, etat_num, etaw_num]);
i_dot = subs(i_dot, [fw wr, Dw], [fw_num, wr_num, Dw_num]);
i_dot = subs(i_dot, [mreel, mflange, rhow], [mreel_num,
    mflange_num, rhow_num]);
i_dot = subs(i_dot, [Kt, Kv, Ra, La], [Kt_num, Kv_num, Ra_num,
    La_num]);
i_dot = subs(i_dot, [g, phi1, phi2, phi3],[g_num, phi1_num,
    phi2_num, phi3_num]);
i_dot = simplify(i_dot);


V = subs(V, [Dr0, Drfull, etat, etaw], [Dr0_num, Drfull_num,
    etat_num, etaw_num]);
V = subs(V, [fw wr, Dw], [fw_num, wr_num, Dw_num]);
V = subs(V, [mreel, mflange, rhow], [mreel_num, mflange_num,
    rhow_num]);
V = subs(V, [Kt, Kv, Ra, La], [Kt_num, Kv_num, Ra_num, La_num])
    ;
V = subs(V, [g, phi1, phi2, phi3],[g_num, phi1_num, phi2_num,
    phi3_num]);
V = simplify(V);

syms v vds vdss
syms yd yds ydss

V = subs(V, [s_dot, diff(s_dot,s), diff(s_dot,s,s)],[v,vds,vdss
    ]);
V = subs(V, [Ld, diff(Ld,s), diff(Ld,s,s)],[yd,yds,ydss]);
V = V(s);

%% Generating Model
matlabFunctionBlock('winch/I',I);
matlabFunctionBlock('winch/D',D);
matlabFunctionBlock('winch/Winch/L_ddot',L_ddot);
matlabFunctionBlock('winch/Winch/i_dot',i_dot);
matlabFunctionBlock('winch/Control_Law/Soft_Landing',V);
```

A.4 Winch Simulation Model Initialization Script

```
p1 = 0.01;
p2 = 0.1;
p3 = 0.0001;
q1 = 1;
q2 = 1;
q3 = 1;
mL = 16;
L_0 = 100;

p = [p1;p2;p3];
mu = 0;
yds = -1;
uMax = 0.5;
Lw0 = 10;

s1 = 0;
s2 = 3;
a1 = 0.01;
a2 = 0.01;


a = [lyap(p1,q1)  ;  lyap(p2,q2);lyap(p3,q3)];
```

A.5 Soft Landing Speed Profile

```
function [v, vds, vdss] = speed_profile(s,uMax, yds, s1,s2,a1,
   a2, a3)


if (s >= s1) && s < (s1 + 4*(s2 - s1)/5)
    v = uMax/(pi*abs(yds))*atan((s - s1 - a1)/a2) + uMax/(2*abs
       (yds));
    vds = uMax/(pi*abs(yds))*1/(1 + ((s - s1 -a1)/a2)^2);
    vdss = uMax/(pi*abs(yds))*(2*a1 - 2*s + 2*s1)/(a2^2*((a1 -
       s + s1)^2/a2^2 + 1)^2);
elseif (s >=  (s1 + 4*(s2 - s1)/5)) && s <= s2

    v = uMax/(pi*abs(yds))*atan((s2 - a3 - s)/a2) + uMax/(2*abs
       (yds));
    vds = uMax/(pi*abs(yds))*-1/(1+ ((s2 - a3 -s)/a2)^2);
    vdss = uMax/(pi*abs(yds))*(a3 + 2*s - 2*s2)/(a2^2*((10*a1 +
        s - s2)^2/a2^2 + 1)^2);
else
  v = uMax/(pi*abs(yds))*atan((s2 - a3 - s)/a2) + uMax/(2*abs(
     yds));
    vds = uMax/(pi*abs(yds))*-1/(1+ ((s2 - a3 -s)/a2)^2);
    vdss = uMax/(pi*abs(yds))*(a3 + 2*s - 2*s2)/(a2^2*((10*a1 +
        s - s2)^2/a2^2 + 1)^2);
end
```