NTNU
Norwegian University of
Science and Technology

DESIGN, MODELING AND CONTROL OF A GENERIC CRANE FOR MARINE APPLICATIONS

Fredrik Andre Gyberg

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

Norwegian University of Science and Technology
Department of Marine Technology

NTNU – Trondheim
Norwegian University of
Science and Technology

DESIGN, MODELING AND CONTROL OF A GENERIC CRANE FOR MARINE APPLICATIONS

Fredrik Gyberg

January 2017

TMR4930 - Marine Technology Master Thesis
Department of Marine Technology
Norwegian University of Science and Technology

Supervisor 1: Associate Professor Eilif Pedersen

Supervisor 2: PhD Candidate Stian Skjong

MASTER THESIS IN MARINE ENGINEERING

FALL 2016

FOR

STUD.TECHN. FREDRIK GYBERG

DESIGN, MODELLING AND CONTROL OF A GENERIC CRANE FOR MARINE APPLICATIONS

Work description

Marine cranes comes in different designs and for different operations and often the foundation is a ship or a moving object and the requirement is to put the load safely on shore or on a different moving floater. Crane dynamics and crane heave compensation therefore needs to be designed and analyzed thoroughly. In this project a scaled crane for laboratory testing should be developed, proper mathematical models and control system should be developed and the capabilities of the crane should be demonstrated both mathematically and using tests in the lab.

Scope of work:

1. Perform a literature review on maritime cranes, crane dynamics and modelling of such systems.
2. Describe and document the improved design developed based on you initial pre-project design.
3. Develop a high fidelity mathematical model of the crane design based on Lagrange-Hamiltonian theory and bond graphs and carry out simplified simulations to demonstrate accuracy and capability.
4. Develop and implement necessary controllers for basic operation of the crane. Document the implementation and write a simple operations manual for the crane.
5. Implement the control structure into you mathematical model and perform a systematic investigation demonstrating the capability of the design.

The report shall be written in English and edited as a research report including literature survey, description of mathematical models, description of control algorithms, simulations results, discussion and conclusion including a proposal for further work. Source code developed shall be provided on a CD or equivalent with code listing enclosed in appendix.

The Department of Marine Technology, NTNU, can use the results freely in its research work by referring to the students work.

The thesis should be submitted in three copies within 5 months after the registered start-up of this thesis.

Trondheim September, 2016

Eilif Pedersen
Associate Professor (Advisor)

Stian Skjong
PhD-student (Co-Advisor)

Preface

This masters thesis is submitted to the Norwegian University of Science and Technology (NTNU) in fulfillment of the requirements for the degree Master of Science in Marine Engineering. The thesis is a continuation of the work presented in Gyberg (2015), and documents the process of designing, modeling and controlling a scaled down crane model intended for lab use. The scaled crane model is at the time of writing this thesis partly produced and installed in the lab. Test results from the crane lab is included in the thesis. The project have been conducted at the Department of Marine Technology, under the supervision of Associate Professor Eilif Pedersen, to whom I would like to extend a special thanks for good advice and makeing this project possible. I would also like to thank Phd-Candidate Stian Skjong for always being available for guidance and discussions during the project. In addition to the official advisors assigned to this project several persons deserve a huge thanks for their involvement in the project, these include; Senior Consultant Peter Sandvik, Engineer Gunnar Bremset, Senior Engineer Frode Gran, and all the other guys at the workshop that always step up to help. This project would never been a reality without you.

Trondheim, 2017-01-16

Fredrik Gyberg

Abstract

English Version

This thesis is presenting the results and processes of designing a scaled down crane model that is intended for studying of crane behavior and control systems. The design process is performed as an iteration process with a system design approach. The objective is to develop a crane that will provide a solid basis for future research and testing.

One of the most central parts of the work done and presented in this thesis is the development of a simulation model that is able to predict the behavior of several different configurations of the scaled down crane model. The usefulness of the simulation model during the design process of the crane is well documented throughout this thesis.

Tests comparing the simulation model and the real crane is performed and documented, and the accuracy of the simulation model is demonstrated. The results found in the comparison is that the behavior predicted by the simulation model is more or less identical to the behavior of the real crane. This concludes that the methods and models used to create the simulation model presented in this thesis can be considered verry effective.

The design of the crane is not only based on the results form the dynamic simulations. The structure have been analyzed with the use of final element analysis to make sure that the structure is in accordance with the design requirements. It has also been done several conceptual evaluations of the crane design regarding modularity and future expansion and modifications. This is done to making the crane a solid basis for further development and modification.

A control system is developed and tested in both the simulation model and in the crane lab. The control systems is fairly basic and only consisting of controllers that control the speed of the motors powering the crane and the initiation of the winch brake. Even though the controllers are basic and only controlling one separate motor each the level of accuracy that is possible to achieve with this type of control has proven to be impressive. The controllers will therefore provide a good basis for further development of more sophisticated control of the crane.

Sammendrag

Norsk Versjon

Denne rapporten presenterer resultatet og prosessen som har blitt gjenomført for å designe en nedskalert kranmodel. Kranmodellen er tiltenkt å brukes som et utgangspunkt for å studere kraners dynamiske oppførsel og utvikling av kontrollsystemer for å kontrollere denne oppførselen. Designprosessen er utført med utgangspunkt i en iterativ prosess og med et system-design perspektiv. Målet for prosjektet er å utvikle en nedskalert kran som vil gi et solid fundament for fremtidig forskning og testing.

Et av de mest sentrale aspektene som er presentert i denne rapporten er utviklingen av en simuleringsmodell som er kapabel til å forutse oppførselen til flere forskjellige konfigurasjoner av kranen. De ulike anvendelsesområdene for simuleringsmodellen i designprosessen er dokumentert gjennom hele denne rapporten.

Tester som sammenligner resultatene fra simuleringsmodellen og den faktiske kranen er utført, og nøyaktigheten til simuleringsmodellen er evaluert. Resultatet er at nøyaktigheten til simuleringsmodellen er tilnærmet perfekt. Det konkluderes derfor med at metodene og modellene som er brukt for å bygge opp simuleringsmodellen er svært effektive.

Designet av kranen er ikke bare basert på resultatene fra simuleringsmodellen. Strukturen har også blitt vurdert ved bruk av "final element" analyse. Dette for å være sikker på at strukturen til kranen er kapabel til å motstå alle krefter den vill bli utsatt for. Strukturens styrke er evaluert med utgangspunkt i designkriteriene satt i denne rapporten. Flere konseptuelle vurderinger har også blitt utført for å sikre god modularitet og mulighet for fremtidige endringer og forbedringer. Dette er gjort for å sikre at kranen har et optimalt utgangspunkt for testing av fremtidige problemstillinger.

Et kontrollsystem er også utviklet og testet i både simuleringsmodellen og i laben hvor kranen er installert. Kontrollsystemet som er utviklet er et ganske elementært kontrollsystem som separat tar for seg de forskjellige motorene som gir kranen den ønskede bevegelse. På tross av at kontrollsystemet er grunnleggende og lite avansert så er graden av nøyaktighet som er mulig å oppnå funnet til å være svært høy. Kontrollsystemet som er utviklet er også ansett for å gi et godt utgangspunkt for å utvikle mer avansere kontrollere.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

To be able to move heavy cargo in a safe and efficient way is often a huge challenge, especially in a marine environment where these systems often are affected by severe environmental conditions. Lifting operations is often done by some sort of crane, and it is critical that the crane is functioning as intended. As an example the crane can be placed on a floating object, e.g. a vessel, and are to place a wind turbsingtop of an offshore windmill structure, illustrated in figure 1.1. In an operation like this the positioning accuracy and wave motion compensation will be critical aspects. This means that the crane designer will have to have a thorough understanding of the crane structure, dynamics and control.

Figure 1.1: Offshore Windmill Construction

Crane operations is traditionally performed by manual input from a crane operator. The safety, accuracy and efficiency of the crane operation is therefore greatly influenced by the skill of the operator. In order to improve the crane operations advanced control systems are developed and implemented.

This thesis will document the work done when designing a scaled crane model and the installment of the model in a lab environment. The intension for the crane lab is to extend the understanding of crane behavior and be a basis for future research of this subject. Such research fields can be: Automation, positioning accuracy, wave motion compensation, collision avoidance, etc.

1.2 Exposition of the Problem

This thesis documents several aspects regarding crane design. One of the most central aspects is the development of a generic and accurate mathematical simulation model that will be used as a tool to determine and verify the design of the crane. This model will be developed using bond-graphs based around a kinematic model for the rigid bodies of the crane which is derived with the use of Lagrange-Hamiltonian theory.

Based on the design developed by the author in Gyberg (2015) input from the project advisors and results form the simulation model, a complete 3D CAD model of the crane is developed. The 3D model is used to calculate accurate parameters for the simulation model, such as masses and inertias. The model is also basis for the production drawings for the physical crane installed in the lab.

A basic control system for the finished crane is implemented both in the simulation model and in the physical crane. The control system will provide basic control of the crane and serve as a basis for further development of advanced control-systems.

The physical crane and the installation in the lab will be documented, including an operations manual for the crane.

1.3 Related Work

Several references have been used during the work that this thesis presents, this section will provide a short presentation of the most important references.

One of the main challenges in this thesis is to design a well functioning crane that are to be the basis in the crane lab that is under construction. The two main references that have been used regarding systems engineering/design is Kossiakoff et al. (2011) and Jackson (2010). Both references make a systems approach to design and present various methods and technics on how a complicated design process can be structured.

The simulation model presented in this thesis is based on theory from several sources. Rokseth (2014) provides a framework for modeling of the kinematics of the rigid parts of the cranes. Karnopp et al. (2012) and Pedersen and Engja (2010) provides the basic modeling theory, and theory on bond graph modeling, and Skjong (2014) provides the theory for the wire and winch model for the crane.

The structural strength of the crane is an important measure, and the different parts of the crane will therefore be analyzed with the use of the final element method. Madenci and Guven (2006) is consulted to get an understanding of the theory of the final element analysis, and Redell (2008) and Autodesk (2016) is used as a practical reference on how to use the final element analysis tool in the engineering software Autodesk Inventor.

1.4 Structure of the Thesis

This section is providing a short introduction to the contents of the different chapters in the thesis, thus giving an overview of the structure of the thesis.

Chapter 2 provides an introduction to the theoretical background of the work done in this thesis. This includes dynamic modeling with the use of Langrange-Hamiltonian theory and bond graphs. A short introduction to system design, marine cranes, and final element analysis is also provided.

Chapter 3 provides the framework which is the basis for the crane design. A presentation of the crane design on a conceptual level is also provided, and a parameterized concept which is used as a basis for the simulation model is presented.

Chapter 4 presents the simulation model developed with basis in the parameterized concept developed in chapter 3, and the theory presented in chapter 2. The modeling procedure is well documented with all relevant equations and figures showing the bond graph layout. This chapter is also presenting the control system developed for the simulation model, and the capability of the control system is documented.

Chapter 5 presents the final design of the crane, and the most important processes and considerations that have been made during the design of the crane. The simulation model from chapter 4 is used continuously through this chapter both as a tool to determine design parameters and as a tool to verify the capability of different components selected for the crane.

Chapter 6 presents the lab setup with the partially produced crane. All equipment installed in the lab is presented, this includes a presentation on how the control system is implemented in the lab and which parts the control system is built upon. A qualitative review of the crane performance in the lab environment is also presented.

Chapter 7 presents final test results from both the simulation model and the real crane installed in the lab. The results are compared and a reviewed of the accuracy and capability of the simulation model and the final crane is presented.

Chapter 8 Presents the final conclusions and recommendations for further work.

Chapter 2

Background Material

2.1 Marine Cranes and Operations

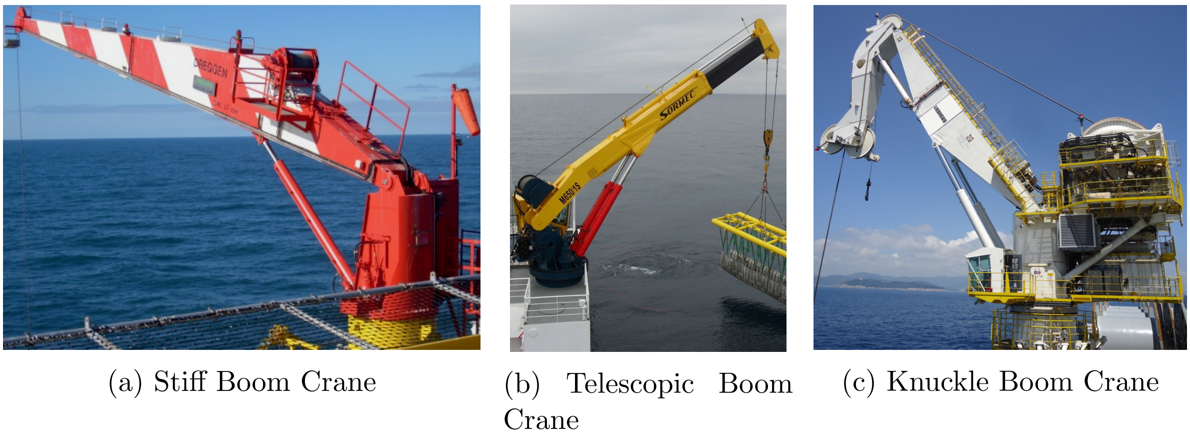
Cranes are central in several marine operations today. These operations can be standardized operations where the cranes can be optimized with regard to efficiency, e.g. transfer of cargo between a vessel and shore in merchant vessels. The operations can also be almost unique complicated operations, e.g. offshore and subsea construction. It is natural with different operations that there are various crane configurations, each specialized for the intended operation.

A review of some vendors that supply marine cranes have been performed to provide an insight of what types of cranes that are supplied today. A summary of the result is presented in table 2.1. Se also Palfinger (2016), Heila (2016) and Appleton (2016) for reference.

From table 2.1 its clear that the same crane types are available from different vendors, and it also suggests that some configuration of these crane types are relevant for most marine applications. There are three cranes that all the vendors supply, these cranes are also the basis for the crane types that are unique to the three vendors. The common crane types are; the stiff/straight bom crane, the telescopic/extending boom crane, and the knuckle bom crane. These three types are illustrated in figure 2.1

Table 2.1: Crane Vendors and their Crane Types

Vendor	Vendor Description	Crane types
Palfinger	*"Palfinger marine is the global leading manufacturer of highly reliable, innovative and customized deck equipment and handling solutions for the maritime industries."*	• Foldable Knuckle Bom Cranes. • Stiff Bom Cranes. • Telescopic Bom Cranes. • Knuckle Bom Cranes. • Traveling Cranes.
Heila Cranes	*"Heila Cranes SpA, manufacture sector-specific heavy-duty cranes expressly designed and manufactured for marine applications, both shipboard and off-shore, where design, safety and reliability standards are extremely demanding and severe hostile environmental conditions prevail."*	• Stiff Bom Cranes. • Telescopic Bom Cranes. • Fully Foldable Telescopic Cranes. • Knuckle Bom Cranes.
Appleton Marine	*"Appleton Marine has continued to supply custom marine deck machinery including cranes, winches, windlasses, capstans, hose reels and special systems for military and commercial shipbuilding since 1971."*	• Straight Boom Crane. • Extending Boom Crane. • Knuckle Boom Crane. • Folding Boom Crane.



(a) Stiff Boom Crane

(b) Telescopic Boom Crane

(c) Knuckle Boom Crane

Figure 2.1: Common Marine Crane Types

The three crane types in figure 2.1 has a lot in common. They are all mounted to the vessel through a rotating platform. The arm/arms of the cranes are all actuated by hydraulic linear actuators. The lifting action is accomplished by a combination of actuators mounted on the crane arm/arms and the use of a winch. The telescopic boom crane and the knuckle boom crane differ from the stiff boom crane by having one more degree of freedom, and the telescopic boom crane and the knuckle boom crane differ from each other in how this last degree of freedom is implemented. The telescopic boom cane can change the angle of the arm and also the length of the arm. The knuckle boom crane consists of two arms which can both change their angles.

2.2 System Design/Engineering

When designing a system such as a crane it is important to do this as a structured process with a clearly defined goal. The design process used in this thesis is based on Kossiakoff et al. (2011) and Jackson (2010). The design process will be executed with a system approach. The system approach differ from traditional engineering by analyzing the system and its interaction with the environment as a whole, compared to analyzing each part of the system separately. This is often a challenge as many systems often are quite complex and difficult to understand as a whole. It is therefore in many cases useful to divide the complex main system into several smaller sub systems and then studying and analyzing these smaller systems individually and their interaction with each other without loosing the focus of the performance of the complete system. This is a difficult task, and the design process should therefore be structured prior to the design work.

One way to structure the design process is an iteration process illustrated by the design spiral in figure 2.2. This involves defining the requirements of the system and using this to define one or more concepts. The concept(s) will then be an initial guess(es) on the solution. The concept(s) is then analyzed over several iterations where all aspects of the concept is evaluated with regard to the requirements of the system. The focus of the iteration process is to optimize the design and verify that the design will fulfill the system requirements. The reader is referred to Kossiakoff et al. (2011) and Jackson (2010) for further details on the system approach to design/engineering.

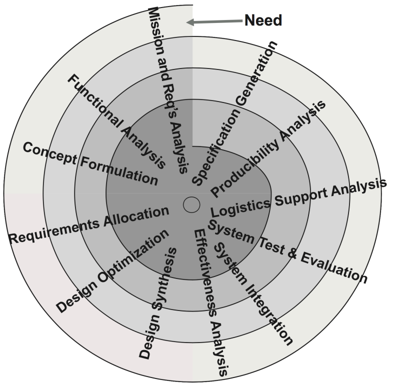


Figure 2.2: Design Spiral

2.3 Crane Dynamics and Modeling

As explained in section 2.1 marine cranes exist in several different shapes and modes of operation. Common for most of the different crane types are however that they can be characterized as a set of rigid bodies in spatial motion. This means that the dynamics of the crane can be modeled with the framework developed by Rokseth (2014). This section will give a brief introduction to the theory used to create a high fidelity mathematical model of the crane. The theory presented in this section is based on Karnopp et al. (2012), Pedersen and Engja (2010) and Rokseth (2014).

2.3.1 Bond-Graph modeling

The bond graph method is an energy based graphical approach to model dynamic systems. The method uses several basic ideal elements, each with properties of either storing, transforming, dissipating or supplying energy. In order to make the bond graph it is necessary to identify the flow, storage and interchange of energy in the real system and then use the basic elements, connected by power bonds, to simulate these characteristics. The power bond connecting the different elements carries the power from one element to another without any loss. The power is defined as the product of the effort, $e(t)$, and the flow, $f(t)$, ref equation 2.1.

$$P(t) = e(t)f(t) \tag{2.1}$$

The effort and flow represents physical quantities in different systems. The effort can represent a force or a torque in mechanical systems, a pressure in hydraulic systems, or the voltage in electrical systems. The flow can represent a velocity in mechanical systems, a volume flow in hydraulic systems and the current in electrical systems.

Figure 2.3 shows an example of two systems connected by a power bond. The positive direction of the flow of energy is defined by the direction of the half arrow, the vertical line (causality stroke) defines for which system the effort is input.

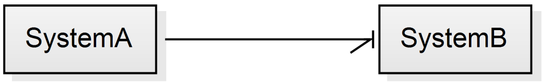


Figure 2.3: Energy Interchange between two Systems over Power Bond

In addition to the power variables, effort, $e(t)$, and the flow, $f(t)$, there are tow variables that that qualifies as states.Theese are the momentum, $p(t)$ and the displacement, $q(t)$, defined as follows in equation 2.2 and 2.3:

$$p(t) = \int_0^t e(t)dt + p(0) \tag{2.2}$$

$$q(t) = \int_0^t f(t)dt + q(0) \tag{2.3}$$

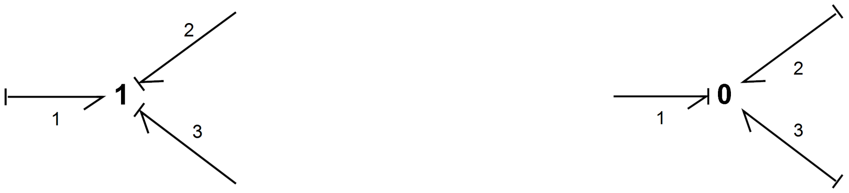
Where $p(0)$ and $q(0)$ are initial conditions.

The state variables are used in the state equations describing the behavior of the different bond-graph elements.

To build a bond graph model of a physical system several different ideal elements will have to be used. A short presentation of the different elements used in this thesis will follow.

Junction Elements

The junction elements are used to connect the other elements in such a way that the complete system is represented by the interconnected elements. There are two types of junctions, the 1-junction and the 0-junction. Figure 2.4 show an example of a 1-junction and a 0-junction.



(a) 1-Junction (b) 0-Junction

Figure 2.4: Junction Elements

The properties of the 1-junction is that all power bonds connected to the junction must have the same flow. This means that only one port on the 1-junction can have a flow as input. The other ports will send out the flow received by the first port. The constitutive relation for the 1-junction in figure 2.4a can then be written as follows in equation 2.4

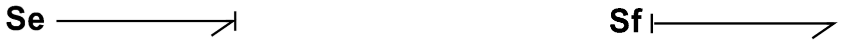
$$f_1(t) = f_2(t) = f_3(t) \qquad e_1(t) = e_2(t) + e_3(t) \tag{2.4}$$

The properties of the 0-junction are that all power-bonds connected to the junction must have the same effort. This means that only one port on the 0-junction can have an effort as input. The other ports will send out the effort received by the first port. The constitutive relation for the 0-junction in figure 2.4b can then be written as follows in equation 2.5

$$e_1(t) = e_2(t) = e_3(t) \qquad f_1(t) = f_2(t) + f_3(t) \tag{2.5}$$

Sources

Sources model energy supply to the system. There are two different types of sources, the flow source, S_f and the effort source, S_e. They each supply either a flow or an effort as a function of time. It is important to note that the effort source supplies an effort regardless of what the flow is, and the flow source supplies a flow regardless of what the effort is. Figure 2.5 illustrates the two source elements.



(a) Effort Source

(b) Flow Source

Figure 2.5: Source Elements

Resistor Element

The resistor element models power dissipation out of the system. The element will either take a flow as an input and calculate an output effort, or take an effort as an input and calculate a flow as an output. The positive flow direction is usually into the resistor element so that the energy is flowing out of the system which the resistor element is a part of. Figure 2.6 illustrates an example of a resistor element.

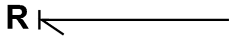


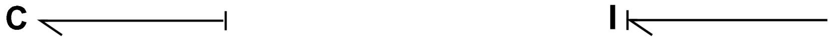
Figure 2.6: Resistor Element

The constitutive relation for the resistor element will be on the form as follows in equation 2.6

$$e(t) = \phi_R(f(t)) \tag{2.6}$$

Storage Elements

There are two types of storage elements in the bond-graph environment. One that store potential energy: The capacitor element, C. And one that store kinetic energy: The inertia element, I. These are illustrated in figure 2.7



(a) Capacitor Element

(b) Inertia Element

Figure 2.7: Storage Elements

The constitutive relation for the capacitor and inertia elements will be on the form as follows in equation 2.7, and 2.8

$$e(t) = \phi_c^{-1}(q(t)) = \phi_c^{-1}(\int_0^t f(t)dt) \tag{2.7}$$

$$f(t) = \phi_I^{-1}(p(t)) = \phi_I^{-1}(\int_0^t e(t)dt) \tag{2.8}$$

As seen from equations 2.7 and 2.8 the constitutive relations does not directly define effort as a function of flow, or flow as a function of effort. This is not problematic when the flow is input to the capacitor element or the effort is input to the inertia element. This is because it is usually no problems related to integrating the flow to get the displacement or integrating the effort to get momentum. This causality assignment is often called integral causality. The problems can appear when the effort is input to the capacitor or the flow in input to the inertia element, ref. equation 2.9 and 2.10.

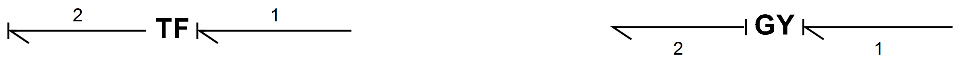
$$f(t) = \frac{d}{dt}(\phi_c(e(t))) \tag{2.9}$$

$$e(t) = \frac{d}{dt}(\phi_I(f(t))) \tag{2.10}$$

It is now required to calculate the derivative of the effort and the flow, which sometimes will be problematic and can lead to increased simulation time. This causality assignment is often called differential causality.

Transformer and Gyrator element

The transformer and gyrator element model energy transformations. The elements are illustrated in figure 2.8.



(a) Transformer Element

(b) Gyrator Element

Figure 2.8: Transformation Elements

The transformer element can for instance be used to transform between angular velocity and linear velocity, or to model the reduction in a gearbox. Its constitutive relation is on the form as shown in equation 2.11

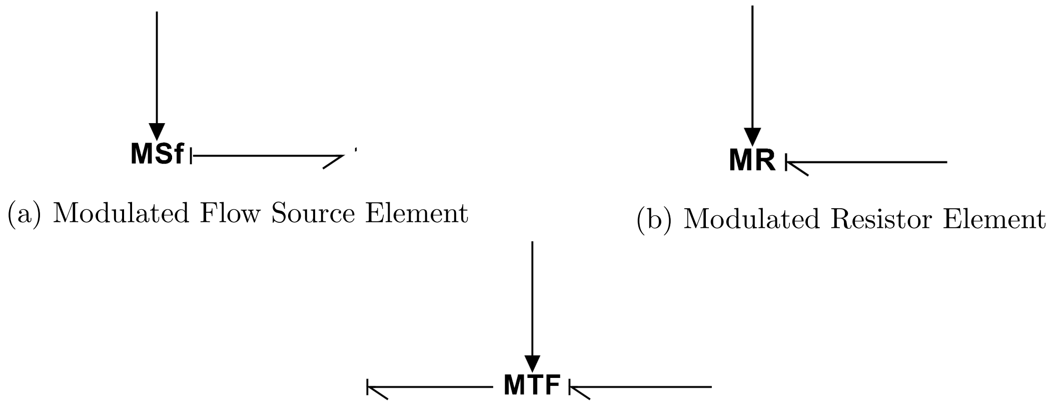
$$\begin{aligned} e_1 &= me_2 \\ f_2 &= mf_1 \end{aligned} \tag{2.11}$$

The gyrator element can for instance be used to transform between the electrical domain and the mechanical rotation domain in an electric motor. Its constitutive relation is on the form as shown in equation 2.12

$$\begin{aligned} e_1 &= rf_2 \\ e_2 &= rf_1 \end{aligned} \tag{2.12}$$

Modeulated Elememt

The constitutive relation in the different bond-graph elements will sometimes consist of parameters which the elements does not have direct access to. This can be solved by sending the parameters into the element over a signal bond. The element will then be called a modulated element, this is illustrated in figure 2.9



(a) Modulated Flow Source Element

(b) Modulated Resistor Element

(c) Modulated Transformer Element

Figure 2.9: Modulated Elements

Combined Inertia Capacitor Element

Sometimes it is more conveient to model a part of the system with the use of differential equations directly, this can for instance be done with the Lagrange equation on state space form. The result will then be a set of first order differential equations which correspond to the effort and flow of the system. This set of equations can then be implemented in a combined inertia capacitor element, also called IC-element. This will be further discussed in section 2.3.2 and 4.1. Figure 2.10 shows an example of an IC-element that models a system with three generalized coordinates.

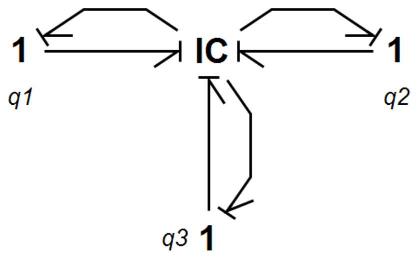


Figure 2.10: Example of an IC-Element for a system of 3 DOF.

2.3.2 Lagrange-Hamiltonian Theory

Lagrange method is an energy based method which consider the kinetic and potential energy of the system, described by a set of generalized coordinates, and generalized forces. In this section it will be shown how to use the Lagrange equation of motion on state space form can be used to derive expressions for both effort and flow for a system of rigid bodies in spatial motion which then can be implemented in the bond graph IC-element.

Lagrange Equation of motion states the following:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\boldsymbol{q}}}\right) - \left(\frac{\partial T}{\partial \boldsymbol{q}} - \frac{\partial V}{\partial \boldsymbol{q}}\right) = \boldsymbol{\tau} \tag{2.13}$$

Where T is the kinetic energy and V is the potential energy both as functions of the generalized coordinates. The vectors $\mathbf{q}$ and $\dot{\mathbf{q}}$ is the generalized coordinates, and their time derivatives respectively. The right hand side of the expression represents the generalized forces acting on the system and is in this case denoted $\boldsymbol{\tau}$.

When the Lagrange equation is solved the result will be a set of second order differential equations which describes the system dynamics. Because the differential equation is of second order the process for solving them can be challenging. It is however possible to modify the Lagrange equation to a form that can be used directly in the bond graph environment.

As shown in Rokseth (2014) the kinetic energy can be expressed as:

$$T(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \frac{1}{2}\dot{\boldsymbol{q}}^T \boldsymbol{B}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{2.14}$$

Where

$$\boldsymbol{B}(\boldsymbol{q}) = \sum_{i=1}^{n} \boldsymbol{J}_i^T(\boldsymbol{q})\boldsymbol{M}_i\boldsymbol{J}_i(\boldsymbol{q}) \tag{2.15}$$

The matrix $\boldsymbol{J}_i$ in equation 2.15 is a matrix satisfies the following relation:

$$\boldsymbol{J}_i(\boldsymbol{q})\dot{\boldsymbol{q}} = \begin{bmatrix} \boldsymbol{v}^0_{cg_i/0} \\ \boldsymbol{w}^0_{cg_i/0} \end{bmatrix} = \boldsymbol{v}_i \tag{2.16}$$

Where $\boldsymbol{v}^0_{cg_i/0}$ is the linear velocities at the center of gravity og the i-th body, and $\boldsymbol{w}^0_{cg_i/0}$ is the angular velocities of the i-th body. Both the linear and angular velocities has to be expressed relative to and in terms of a common reference frame.

The matrix $\boldsymbol{M}_i$ in equation 2.15 is a matrix which includes both the inertia and mass of the i-th body. Mathematically it will take the following form:

$$\boldsymbol{M}_i = \begin{bmatrix} m_i\boldsymbol{I}_{3x3} & \boldsymbol{0}_{3x3} \\ \boldsymbol{0}_{3x3} & \boldsymbol{I}_{bi} \end{bmatrix} \tag{2.17}$$

Where m_i is the mass of the i-th body, $\boldsymbol{I}_{3x3}$ is the 3 x 3 identity matrix, and $\boldsymbol{I}_{bi}$ is the inertia tensor for the i-th body.

It is often the kinematics of the system which is most beneficial to implement in the bond graph IC-element. The potential energy of the system can therefore be set equal zero in equation 2.13. The forces resulting from potential energy can easily be added later as generalized external forces acting on the system. This means that the Lagrange equation will take the following form:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\boldsymbol{q}}}\right) - \frac{\partial T}{\partial \boldsymbol{q}} = \boldsymbol{\tau} \tag{2.18}$$

By combining equation 2.14 with equation 2.18 the following expression for the system dynamics can be obtained:

$$\frac{d}{dt}(\boldsymbol{B}(\boldsymbol{q})\dot{\boldsymbol{q}}) - \left(\frac{1}{2}\dot{\boldsymbol{q}}^T\frac{\partial \boldsymbol{B}(\boldsymbol{q})}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}}\right) = \boldsymbol{\tau} \tag{2.19}$$

Recall from section 2.3.1 that the flow is the generalized velocity, $\dot{\boldsymbol{q}}$, and *effort* is defined as the time derivative of the generalized momentum, $\boldsymbol{e} = \frac{d}{dt}\boldsymbol{p}$. Where the momentum is the derivative of kinetic energy with the respect to velocity, $\boldsymbol{p} = \frac{\partial T}{\partial \dot{\boldsymbol{q}}}$. The Lagrange equation can then be formulated as a state space model as shown equation 2.20 and 2.21.

$$\boldsymbol{f} = \boldsymbol{B}^{-1}\boldsymbol{p} \tag{2.20}$$

$$\boldsymbol{e} = \frac{1}{2}\dot{\boldsymbol{q}}^T\frac{\partial \boldsymbol{B}(\boldsymbol{q})}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}} \tag{2.21}$$

Equation 2.20 and 2.21 can then be implemented in the bond graph IC-element.

2.4 Final Element Analysis

Today the final Element Analysis, or FEA, is more or less mandatory when a new physical system is to be designed. FEA is a numerical method that can easily, for very complex geometries, compute accurate results in various engineering disciplines, these include: Stress/strain calculation, heat transfer and flow calculations. It is the analysis of the structure of the crane with regard to stresses and strains that is applicable for the application of the FEA in this thesis. A short background to the FEA used in this thesis is presented in this section, for a more in depth presentation of FEA the reader is referred to Madenci and Guven (2006) which this theory is based on.

The FEA will in general consist of decompositioning the domain which represents the system that is to be analyzed into a finite number of smaller subdomains or elements with a simple geometry, like rectangles or triangles. The discretization of the domain is called for mashing. Each element, and their boundaries is then analyzed by solving equation 2.22.

$$\mathbf{K}\mathbf{u} = \mathbf{F} \tag{2.22}$$

Where **K** is the system stiffness matrix which represents the system resistance to change. **u** is the vector of unknowns, for instance the variables related to deformation of an element. **F** is the force vector which represents the load that the system is subjected to.

The accuracy of the solution is greatly influenced by the properties of the mash and the size of the mash elements. To make sure that the mash is yielding accurate results a convergence analysis should be performed. This is done by performing multiple simulations where the mash is refined for each step with focus on areas where the stresses/strains are largest. The result for each simulation is then compared and if the refinement of the mash does not yield any change in the stresses/strains for several refinements the result converges and the mash could be considered correct.

The process of mashing the system, establishing the system equations and solving these equations is automated by several engineering software tools. The the analysis in this thesis is done with the built in ANSYS software in Autodesk Inventor following the procedure in Redell (2008) and Autodesk (2016).

The workflow when using Autodesk Inventor to perform FEA analysis is:

1. Design of the part or assembly that is to be analyzed.
2. Assign the materials the part or assembly consist of.
3. Apply the constraints.
4. Apply Loading Condition.

5. Define the contact between the different parts if the domain that is to be analyzed is an assembly.
6. Solve the system by using the mash refinement and convergence plot
7. Evaluate the solution

The evaluation of the solution is the most important step. This could be done partly by making sure the solution converges, and partly by evaluating the constraints and loading condition that is applied to the system in question. It is also important to have a perception on how the part/assembly will be deformed and where the highest stresses will appear. Redell (2008) states that the most common source of errors in FEA is incorrect application of constraints and loading conditions. It is therefore important to understand how the system that is to be analyzed interferes with its environment.

2.5 Software

This section provides short introduction to the different software used to accomplish the results presented in this thesis.

2.5.1 Maplesoft's Maple

Maple is a program that is used to perform symbolic mathematical calculations, it can also be used to generate C code for different algebraic expressions. The program have been useful to perform algebraic operations on relatively large mathematical expressions, thus ensure that the calculations are performed correctly throughout this thesis. The program has also been used to generate the C-code implemented in a dll. file that communicate with the simulation software 20-sim.

2.5.2 MathWork's Matlab

Matlab is a program that is used to perform numerical computing and plotting throughout this thesis. The strength of Matlab is that it is easy to write programs, analyze data and illustrate the results through graphs.

2.5.3 Autodesk's Inventor

Autodesk Inventor is a CAD (computer aided design) program that is used for drawing 3-dimensional solid objects. In this thesis this program is used to create a 3-dimensional model of the crane that is developed. It is also used to calculate masses, inertias and the location of center of gravity for different parts, and other values that depend on the shape of the parts. Inventor has a built in ANSYS FEA solver that is used to perform the FEA presented in this thesis. Inventor is also used to create figures that illustrate this thesis.

2.5.4 Autodesk's AutoCad

AutoCad is a CAD program that is used for drawing 2-dimensional drawings. The program is used throughout this thesis to create different schematics that illustrate the thesis.

2.5.5 Controllab's 20-Sim

20-Sim is a simulation software used to perform simulations with the simulation model presented in this thesis. The program supports the bond-graph language so that the complete bond-graph is implemented in the program. The program supports 3D animation of the simulation. This is used to display a moving 3D animation of the complete crane during simulation.

2.5.6 Microsoft's Visual Studio

Visual studio is used in this thesis to modify C-code generated by Maple and to create a dll. file as an extension for 20-Sim. This is done because some of the expressions that will be implemented in the bond graph model are quite large and computational intensive to calculate for each simulation step by the built in 20-Sim solver. The dll. file is able to evaluate the large expressions at a much higher speed.

Chapter 3

Crane Concept and Framework

3.1 Framework

A framework is needed in order to know what the goal and requirements of the crane is. The framework will be the baseline for the design, and also constitute the requirements that the design will be evaluated up against. A well defined framework is necessary in order for the crane lab to be a flexible and universal testing facility for several different projects involving crane operations, control and construction.

The framework presented in this section is a result of several meetings between the author, Associate Professor Eilif Pedersen, PhD Candidate Stian Skjong, Senior Consultant Peter Sandvik, and Engineer Gunnar Bremset.

The framework is still quite similar as presented in Gyberg (2015), some alterations and additions have however been implemented. The complete framework is presented in the list below.

- The crane model should have a modular design. The modular design should ensure that the crane is easy to take apart and put together, give possibility to replace different parts, and add additional parts and functionality. This can for instance be to change the arms or actuators of the crane with different types, or add functionality to the winch.
- The crane should be developed and produced in a cost effective way. It is considered most cost effective for this project to produce the needed parts with the use of basic production tools, such as lathing, milling and welding, compared to purchasing the needed parts. Some parts will have to be purchased, these are motors, bearings, etc.
- The crane should be able to, with both the arms and the winch, lift and maneuver the design load specified in table 3.1.

- The structure of the crane should be able to withstand a much greater load than the design load, giving the possibility to mount more powerful hydraulic actuators without changing the structure in the future.
- The crane should be able to simulate a real crane in a scale of 1:25. For the scaling conditions se table 3.1.
- The crane design should facilitate testing and research of wave compensating control systems. The wave condition in table 3.1 is set as a design requirement.
- A smaller version than the 1:25 scale version of the crane should be constructed. This should be done by fabricating a shorter crane arm than for the correct scale version. This version of the crane should be of a size where the different parts are not too heavy to lift and move around without the need for support from a forklift or other lifting equipment.
- The crane should have a mounting arrangement which facilitates mounting on different fundaments. In the first stage the crane is to be mounted on a heavy concrete table, the crane can later be mounted on a floating platform/vessel.
- The actuators should be driven directly with electric DC motors. Hydraulics should be avoided at this stage, but it should be possible to implement later, ref the point on modularity.

Table 3.1: Crane Scaling Conditions

	Full Scale	**Model Scale**
Scale	1:1	1:25
	Crane Dimensions	
Capasity	250 metric ton	16 kg
Lifting lenght	25 m	1 m
Lifting height	100 m	4 m
	Sea conditions	
Hs	5 m	0.2 m
Hmax	10 m	0.4 m
Tp	14 s	2.8 s

3.2 Conceptual Design

Several crane concepts has been developed in the process of the design and analysis of the crane. Most of these concepts and the early conceptual evaluations that have been done in this project is presented in Gyberg (2015). The concept is as explained in section 2.2 an integral part of the iteration process that will result in the final design of the crane. The concept should end up in a parameterized model that can form the baseline for the simulations and analysis that will be performed to evaluate the design. This means that it is important to have in mind that the concept allows for changes and optimization when developing the design on a conceptual level.

With basis in section 2.1 and Gyberg (2015) it is clear that the most versatile and universal basis for the crane lab will be to design and build a knuckle bom crane. The knuckle bom crane can easily simulate the stiff/straight bom crane by removing the upper arm, and the extending/telescopic bom crane by either replacing one of the arms with an extending arm, or adding an extending part to one of the arms.

Figure 3.1 illustrates the design of the crane in Gyberg (2015) which is used as a stating point for the conceptual design presented in the sections to come.

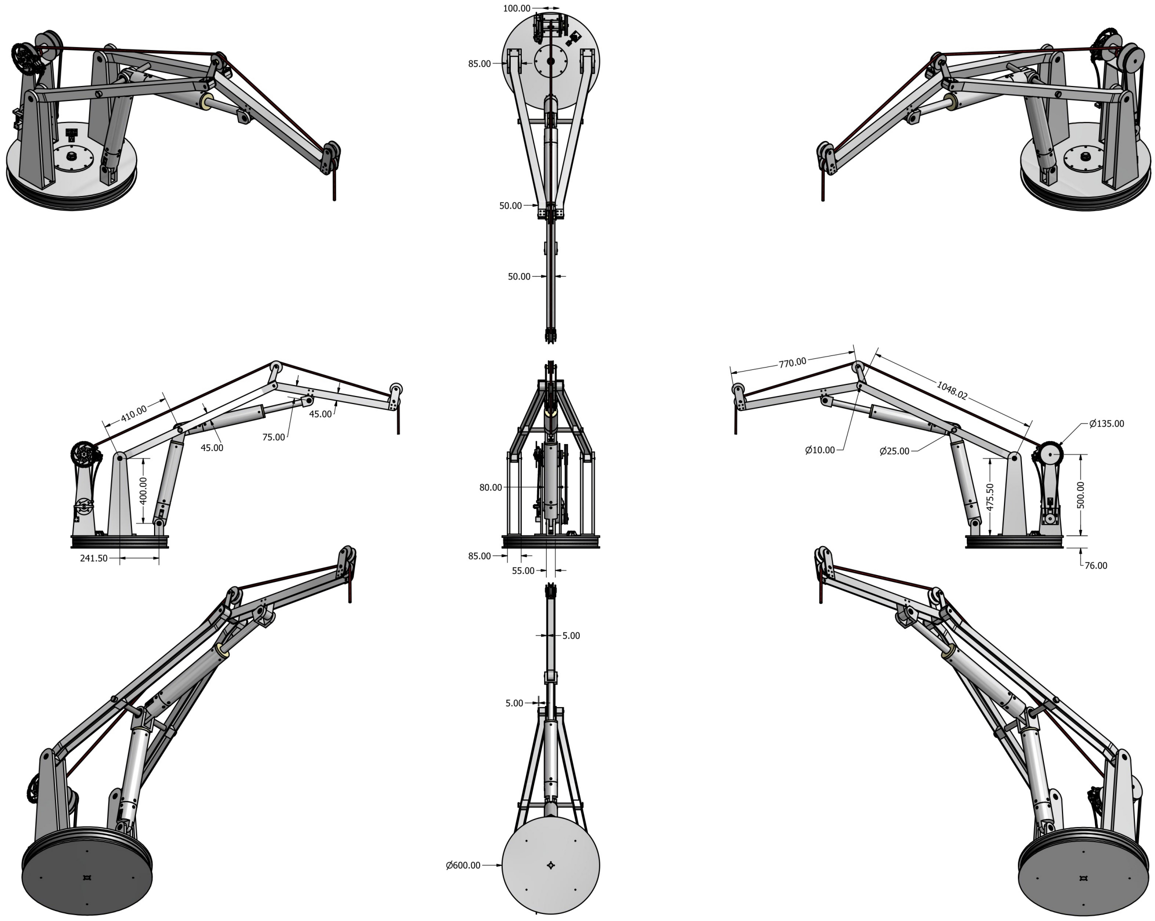


Figure 3.1: Crane Design as Presented in Project Thesis

3.2.1 Base and Winch assembly

The concept for the base and winch assembly have seen several major updates during the design iteration process. One of the most major updates is related to the winch assembly. The winch assembly is moved down and made more compact compared to Gyberg (2015). This is done to improve the structural strength of the winch and brake assembly. The winch will still be driven with a belt connection between the electrical DC-motor and winch, as this was found in Gyberg (2015) to be an effective solution. The solution presented in Gyberg (2015) regarding the actuation of the brake has been revised to use a servo from a radio control vehicle, this is a simpler and more compact solution. The solution with the brake servo will also make room for the motor that is to power the base rotation. Figure 3.2 shows an illustration of the concept for the winch assembly.

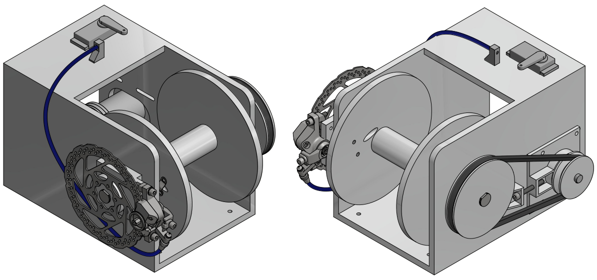


Figure 3.2: Conceptual Design of the Winch Assembly

The lower arm supports have also been revised with fully closed cross-sections to improve the strength and stability. The lower plate of the base will now be a square plate with enough space to accommodate for several different mounting arrangements. Figure 3.3 illustrates the conceptual design of the base.

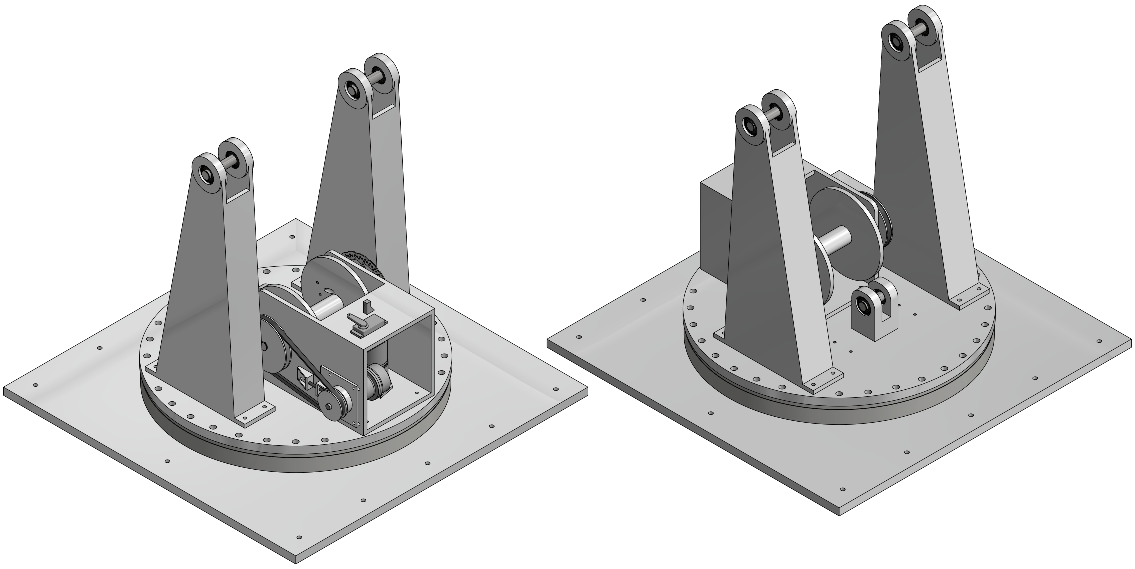


Figure 3.3: Conceptual Design of the Crane Base

The roller bearing presented in Gyberg (2015) will be changed out with a slewing ring with an internal gear, illustrated in 3.4. This is because it is considered a more stable and long lasting solution.

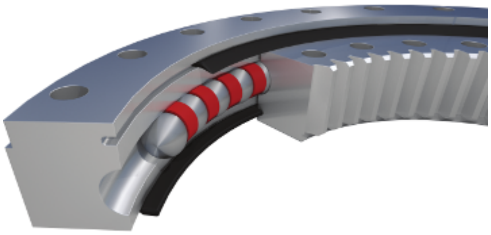


Figure 3.4: Slewing Ring with Internal Gear

The rotation of the base will be enabled by the same principal as in Gyberg (2015), but the friction connection will be changed out with a gear connection between the electrical DC-motor and the slewing ring, as illustrated in figure 3.5

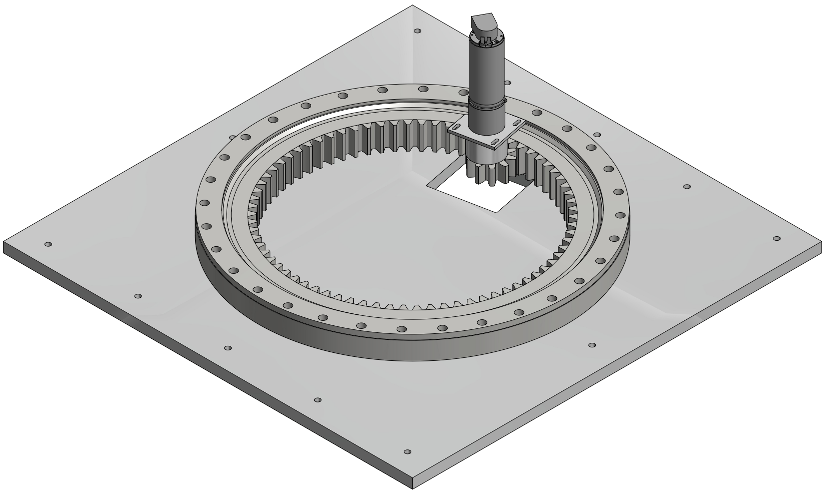


Figure 3.5: Connection between Base Motor and Slewing Ring

3.2.2 Crane Arms

The crane arms will roughly have the same concept as in Gyberg (2015), but some vital changes have been made. The crane arms will now have a design which in greater extent represents how the crane arms on real cranes look. This is accomplished by adding stiffeners at the underside of the crane arms. The stiffeners will in addition to the choice of a square section instead of a U-section of the arm beams and the addition of a truss vigorously increase the strength of the crane arms. Two different sizes of the lower arm will be designed, as pr. the framework presented in section 3.1. The two crane arms will have the same dimensions in both ends, and the same type of actuator connection. This ensures that the arms will be interchangeable without replacing other components. The joints, and the wire guidance rollers will have the same concept as in Gyberg (2015).

Figure 3.6 illustrates the conceptual design of the arms.

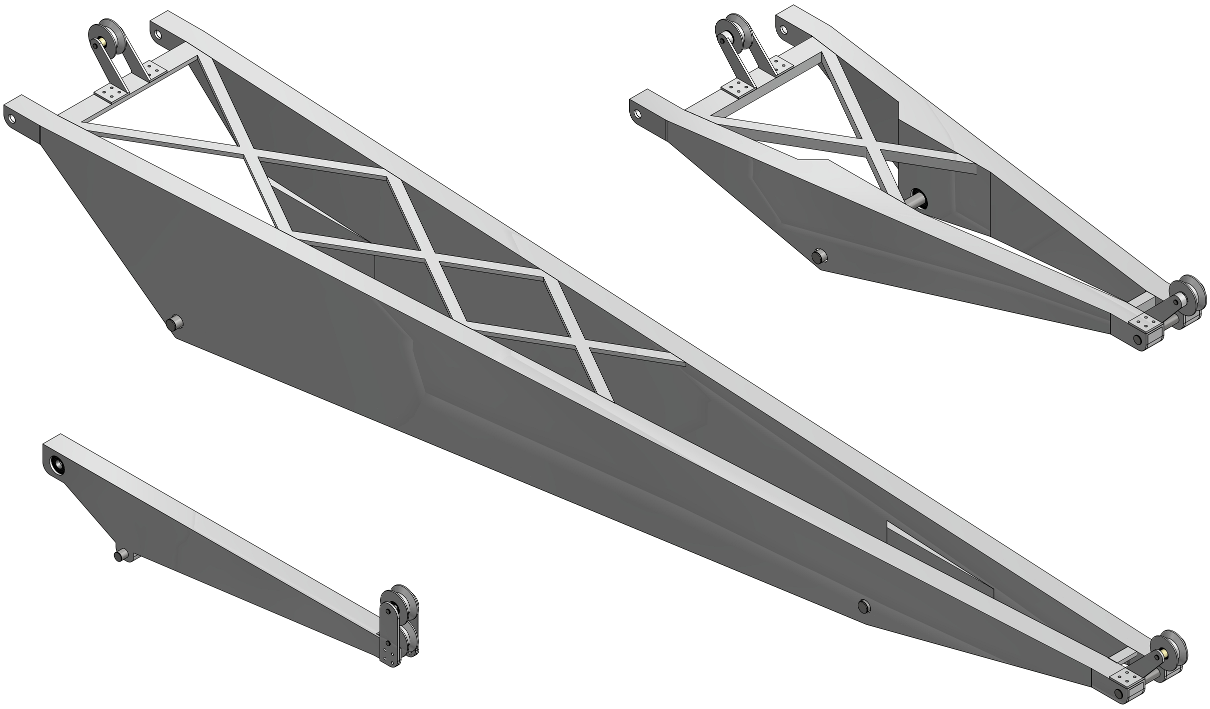


Figure 3.6: Conceptual Design of the Crane Arms

3.2.3 Actuators

The concept for the actuators will more or less be identical with the actuator concept presented in Gyberg (2015). Figure 3.7 illustrates the actuator concept and all the parts that build up the actuator.

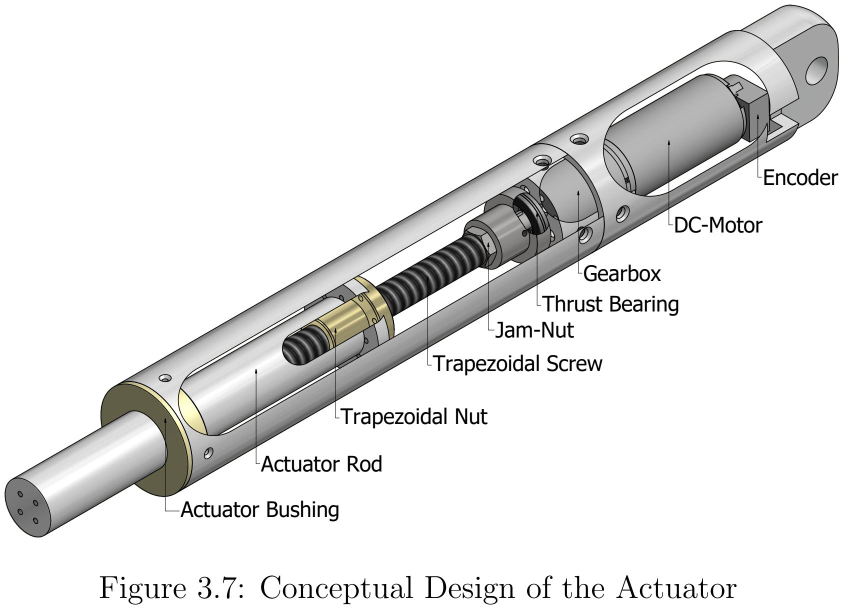
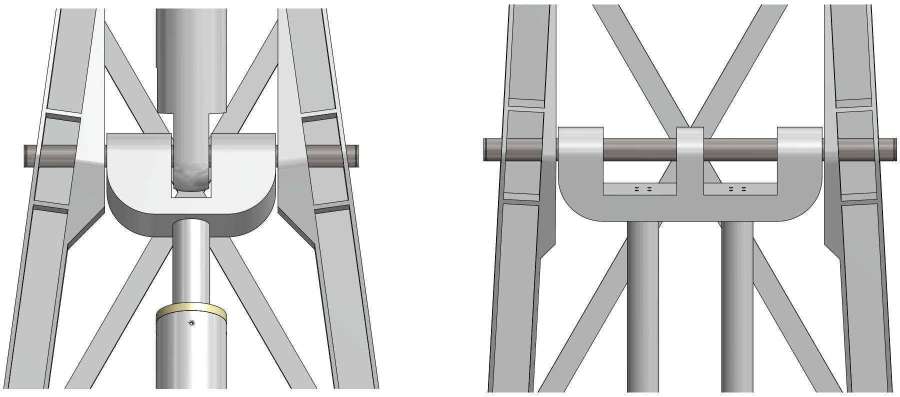


Figure 3.7: Conceptual Design of the Actuator

Conceptual changes have been made in the actuator joints compared to Gyberg (2015). The top bracket of the lower actuator will now be wider so that it supports the connecting rod in the lower arm. Se figure 3.8 for illustration. It is desired that all the actuators for the crane share as many parts as possible, to make this viable a concept is proposed where two or more of the actuator made for the short lower arm is used in parallel to make sure that there is enough force to operate the crane with the the long lower arm.



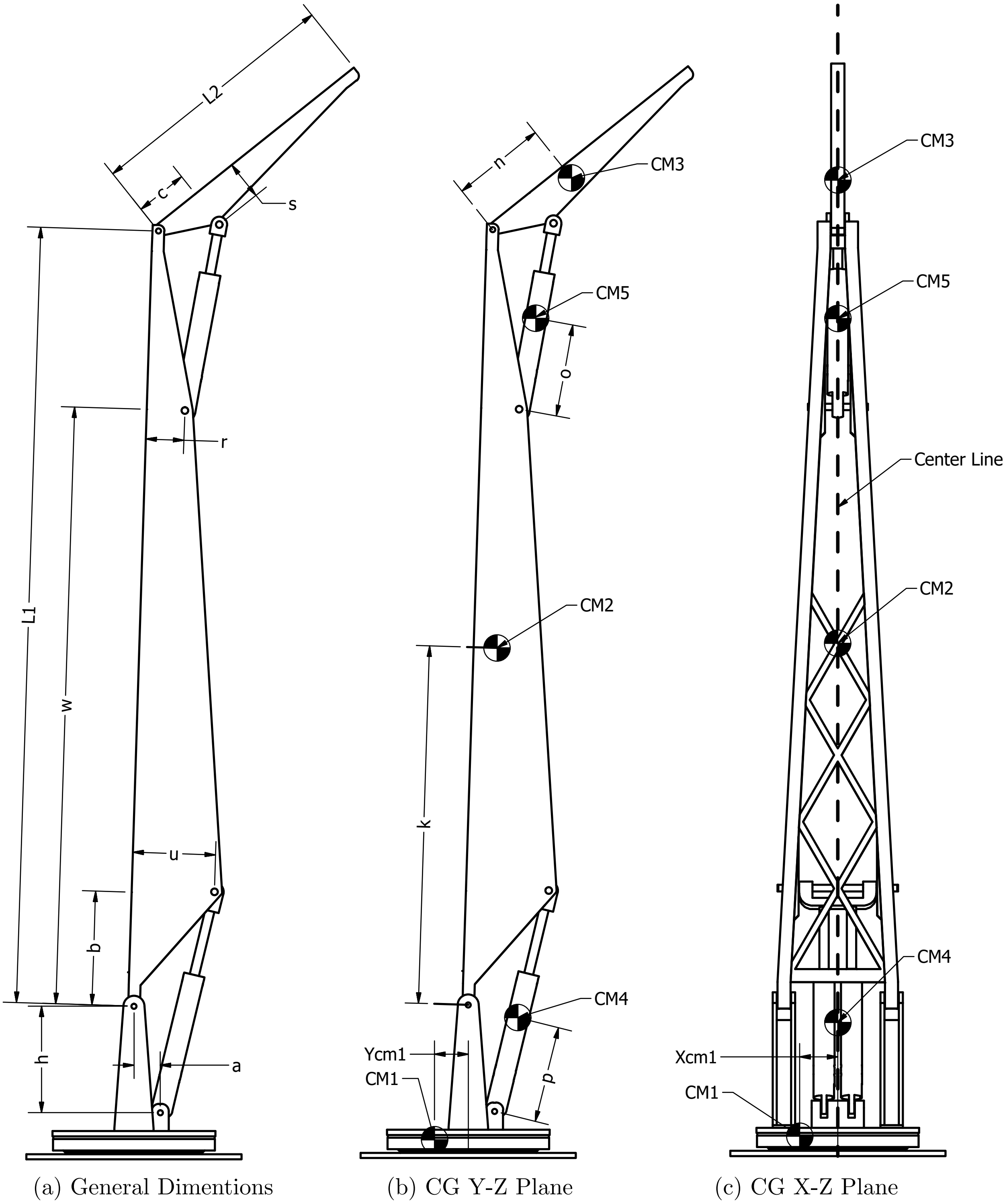
(a) Singel Actuator Joint

(b) Dual Actuator Joint

Figure 3.8: Conceptual Design of the Actuator Joints

3.3 Parameterized Concept

To make the analysis and iteration process of the design as simple as possible the conceptual design presented in the previous sections have been used to define a parameterized concept. The parameters will be input to the simulation model, and should be of such character that they represent any knuckle boom crane. Figure 3.9 displays these parameters. Note that the figure show the crane with the long lower arm, but by setting $w = b$ and $r = u$ the geometry of the shorter arm will also be described, ref figure 3.6.



(a) General Dimentions (b) CG Y-Z Plane (c) CG X-Z Plane

Figure 3.9: Crane General Geometric Parameters

Chapter 4

Simulation Model

A high fidelity mathematical simulation model is developed with basis in the parameterized concept and conceptual design presented in figure 3.9 and section 3.2. The simulation model will be used to predict the response of the crane, evaluate the crane design and determine the parameters of the crane. Development of controllers and demonstrations of the capability of the crane is also done with the use of this simulation model. During the process of creating the model it has been important that the model is developed as a general model that can easily be changed to represent any three degrees of freedom knuckle-boom crane. The framework presented in Rokseth (2014) is the theoretical basis for the development of this model.

4.1 Crane Kinematics

As discussed in Rokseth (2014) bond graph modeling of systems of rigid bodies in spatial motion will often yield I-elements with differential causality. This can give a model that is too computational intensive for any practical usage. One way to avoid this problem is to express the kinematics of the problematic part of the system with use of Lagrange method on state space form as shown in section 2.3.2. The formulation can then be implemented in a IC-element as shown in Rokseth (2014). The IC-element will then calculate both the effort and flow related to the kinematics for the part of the system that is implemented in the IC-element.

4.1.1 Generalized coordinates

It is necessary to express the kinetic energy of the system as functions of a set of generalised coordinates in order to take advantage of the power of Lagrange method on state space form implemented in the bond graph IC-element. The generalised coordinates should be the smallest number of independent coordinates that fully describes the configuration of the system at all times, Pedersen and Engja (2010). The set of generalised coordinates is however not a unique set, often several different sets of generalised coordinates can be chosen for any given system. It is therefore important to note that it is the smallest number of coordinates that is desired. The number of generalised coordinates will therefore often equal the number of degrees of freedom of the system, and can never be less than the degrees of freedom of the system.

The part of the crane that can be considered as rigid bodies in spatial motion has a total of three degrees of freedom. The smallest number of generalized coordinates is therefore equal to three. The coordinates chosen for the crane are as follows in equation 4.1 and illustrated in figure 4.1

$$\boldsymbol{q} = \begin{bmatrix}\theta_1 & \theta_2 & \theta_3\end{bmatrix} \tag{4.1}$$

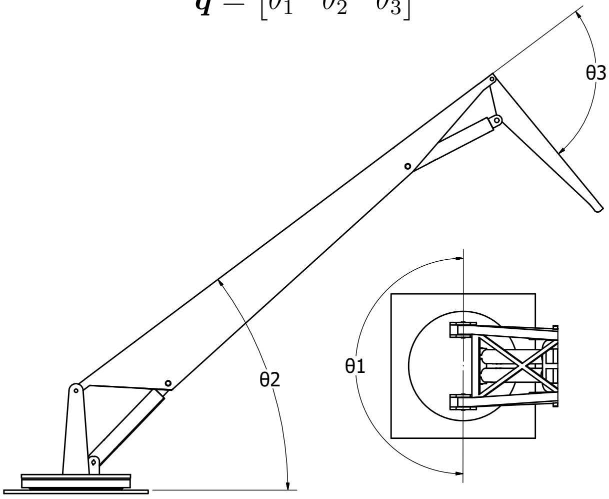
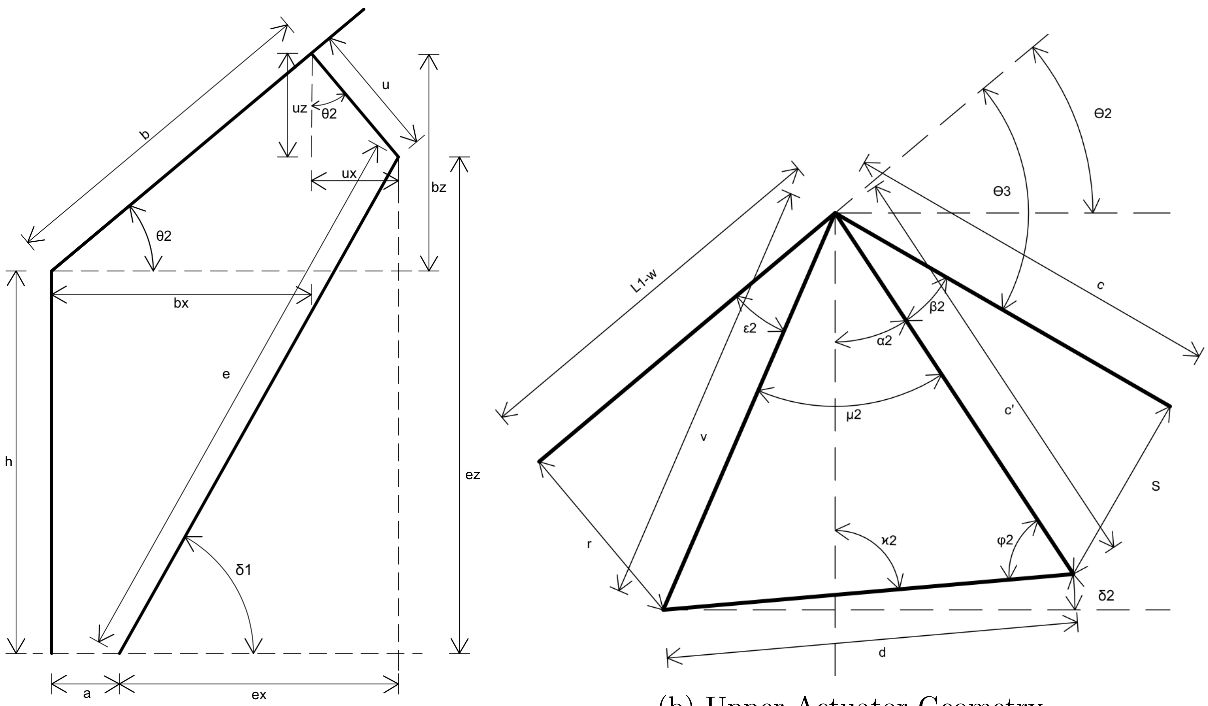


Figure 4.1: Generalized Coordinates

4.1.2 Actuator Rotation

As the mass of the actuators are expected to be quite large, ref. Gyberg (2015), the kinematics of the actuators will have to be included in the dynamic model. The rotation angle of the actuators relative to the horizontal plane are defined as δ_1 for the lower actuator and δ_2 for the upper actuator, ref. figure 4.2. These angles will then need to be expressed as functions of the generalized coordinates.



(a) Lower Actuator Geometry

(b) Upper Actuator Geometry

Figure 4.2: Actuator Geometry

With reference to figure 4.2a and appendix D the angle $\delta_1(\theta_2)$ can be expressed as shown in equation 4.2.

$$\delta_1(\theta_2) = \sin^{-1}\left(\frac{h + \sin(\theta_2)\, b - \cos(\theta_2)\, u}{\sqrt{(\cos(\theta_2)\, b + \sin(\theta_2)\, u - a)^2 + (h + \sin(\theta_2)\, b - \cos(\theta_2)\, u)^2}}\right) \tag{4.2}$$

With reference to figure 4.2b and appendix D the angle $\delta_2(\theta_2, \theta_3)$ can be expressed as shown i equation 4.3.

$$\delta 2(\theta_2, \theta_3) = \pi + \theta_2 + \theta_3 - \sin^{-1}\left(\frac{s}{c'}\right) - \cos^{-1}\left(\frac{\cos(\mu_2)\, v - c'}{\sqrt{v^2 + c'^2 - 2\, v c' \cos(\mu_2)}}\right) \tag{4.3}$$

Where

$$\mu_2 = \pi + \theta_3 - \sin^{-1}\left(\frac{s}{\sqrt{c^2+s^2}}\right) - \sin^{-1}\left(\frac{r}{\sqrt{r^2+(L_1-w)^2}}\right) \tag{4.4}$$

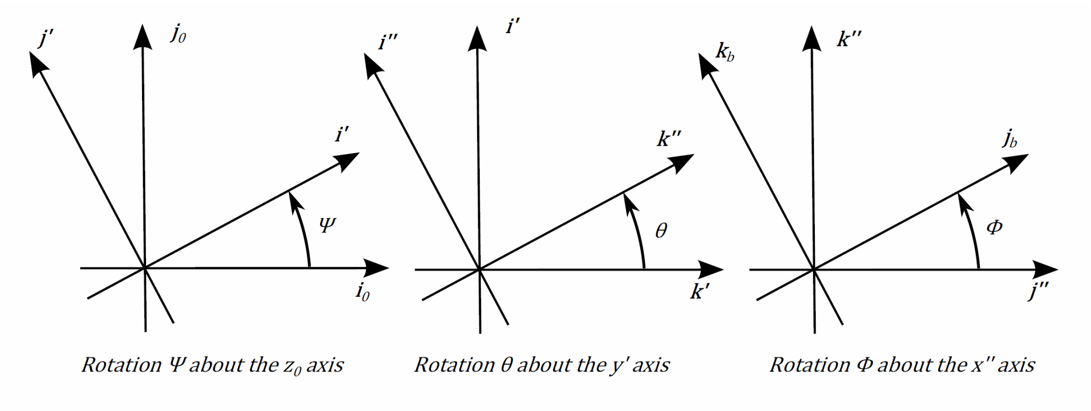
$$v = \sqrt{r^2+(L_1-w)^2} \tag{4.5}$$

$$c' = \sqrt{c^2+s^2} \tag{4.6}$$

4.1.3 Defining Relevant Postitions

It is necessary to define each center of gravity as a functions of the generalized coordinates and relative and in terms of a common reference frame in order to calculate the ***B***-matrix, ref. section 2.3.2 and Rokseth (2014). It can for large and complex systems often be difficult to define these positions directly. The positions are however often easily expressed in terms of local reference frames which can be transformed with the use of rotation transformations to positions expressed in terms of and relative to the common reference frame.

The rotation transformations used in this model is defined by the Euler angles and illustrated by figure 4.3. The positive direction of the rotation is defined throughout this thesis in the direction of the arrows in figure 4.3.



Rotation Ψ about the z_0 axis *Rotation θ about the y' axis* *Rotation Φ about the x'' axis*

Figure 4.3: Prinsipial Rotations Defined by the Euler Angles. Figure from Rokseth (2014)

The rotation transformations on matrix form, also referred to as rotation matrices, are then as defined in equation 4.7. Ref. Rokseth (2014).

$$\boldsymbol{R}_z(\psi) = \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{R}_y(\theta) = \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix}, \quad \boldsymbol{R}_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\psi & -s_\psi \\ 0 & s_\psi & c_\psi \end{bmatrix} \tag{4.7}$$

One local reference frame for each joint is defined as shown in figure 4.4. The 0-frame is the common reference frame and defines the global coordinate system for the entire model. The complete set of reference frames are only shown in the Y-Z plane in figure 4.4, this is because they all have the same orientation as the frame named ”1-frame” in the X-Y plane.

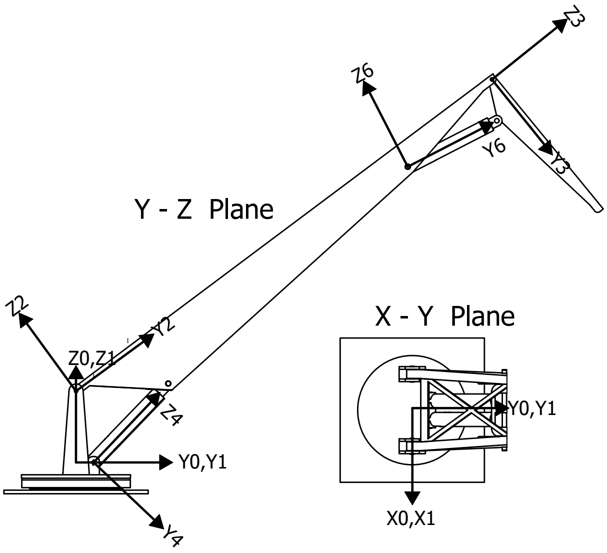


Figure 4.4: Reference Frames

The position of the reference frames is defined as follows in equation 4.8. The notation of reference frame a relative to frame b and in terms of frame c will be $\boldsymbol{r}^c_{a/b}$

$$\boldsymbol{r}^0_{0/0} = \begin{bmatrix}0\\0\\0\end{bmatrix} \quad \boldsymbol{r}^0_{1/0} = \begin{bmatrix}0\\0\\0\end{bmatrix} \quad \boldsymbol{r}^1_{2/1} = \begin{bmatrix}0\\0\\h\end{bmatrix} \boldsymbol{r}^2_{3/2} = \begin{bmatrix}0\\L1\\0\end{bmatrix} \quad \boldsymbol{r}^1_{4/1} = \begin{bmatrix}0\\a\\0\end{bmatrix} \quad \boldsymbol{r}^2_{6/2} = \begin{bmatrix}0\\0\\h\end{bmatrix} \tag{4.8}$$

Equation 4.9 defines one rotational matrix for each of the reference frames displayed in figure 4.4. See also figure 4.1, 4.2a and 4.2b for the relevant angles, and recall that the angels δ_1 and δ_2 is defined as functions of θ_2 and θ_3. The rotation of frame a relative to frame b is denoted R^b_a.

$$\begin{gathered}
\boldsymbol{R}^0_1 = \begin{bmatrix}\cos(\theta_1) & -\sin(\theta_1) & 0\\ \sin(\theta_1) & cos(\theta_1) & 0\\ 0 & 0 & 1\end{bmatrix}, \quad \boldsymbol{R}^1_2 = \begin{bmatrix}1 & 0 & 0\\ 0 & \cos(\theta_2) & -\sin(\theta_2)\\ 0 & \sin(\theta_2) & \cos(\theta_2)\end{bmatrix}, \\
\boldsymbol{R}^2_3 = \begin{bmatrix}1 & 0 & 0\\ 0 & \cos(\theta_3) & -\sin(\theta_3)\\ 0 & \sin(\theta_3) & \cos(\theta_3)\end{bmatrix}, \quad \boldsymbol{R}^1_4 = \begin{bmatrix}1 & 0 & 0\\ 0 & \cos(\delta_1) & -\sin(\delta_1)\\ 0 & \sin(\delta_1) & \cos(\delta_1)\end{bmatrix}, \\
\boldsymbol{R}^1_6 = \begin{bmatrix}1 & 0 & 0\\ 0 & \cos(\delta_2) & -\sin(\delta_2)\\ 0 & \sin(\delta_2) & \cos(\delta_2)\end{bmatrix}
\end{gathered} \tag{4.9}$$

The rotational matrices is combined to express any rotation relative to the 0-frame as follows in equation 4.10.

$$\boldsymbol{R}_2^0 = \boldsymbol{R}_1^0\boldsymbol{R}_2^1, \quad \boldsymbol{R}_3^0 = \boldsymbol{R}_1^0\boldsymbol{R}_2^1\boldsymbol{R}_3^2, \quad \boldsymbol{R}_4^0 = \boldsymbol{R}_1^0\boldsymbol{R}_4^1, \quad \boldsymbol{R}_6^0 = \boldsymbol{R}_1^0\boldsymbol{R}_6^1, \tag{4.10}$$

The frames in equation 4.8 which is not expressed in terms of and relative to the 0-frame, is now expressed in terms of and relative to the 0-frame as follows in equation 4.11

$$\begin{aligned} \boldsymbol{r}_{2/0}^0 &= \boldsymbol{r}_{1/0}^0 + \boldsymbol{R}_1^0\boldsymbol{r}_{1/2}^1, \quad \boldsymbol{r}_{3/0}^0 = \boldsymbol{r}_{2/0}^0 + \boldsymbol{R}_2^0\boldsymbol{r}_{3/2}^2, \\ \boldsymbol{r}_{4/0}^0 &= \boldsymbol{r}_{1/0}^0 + \boldsymbol{R}_1^0\boldsymbol{r}_{4/1}^1, \quad \boldsymbol{r}_{6/0}^0 = \boldsymbol{r}_{2/0}^0 + \boldsymbol{R}_2^0\boldsymbol{r}_{6/2}^2, \end{aligned} \tag{4.11}$$

The position of each center of mass is defined in terms of and relative to their local reference frames as follows in equation 4.12. See figure 4.4 and 3.9 for reference.

$$\boldsymbol{r}_{cm1/1}^1 = \begin{bmatrix} X_{cm1} \\ Y_{cm1} \\ 0 \end{bmatrix} \quad \boldsymbol{r}_{cm2/2}^2 = \begin{bmatrix} 0 \\ k \\ 0 \end{bmatrix} \quad \boldsymbol{r}_{cm3/3}^3 = \begin{bmatrix} 0 \\ n \\ 0 \end{bmatrix} \boldsymbol{r}_{cm4/4}^4 = \begin{bmatrix} 0 \\ 0 \\ p \end{bmatrix} \quad \boldsymbol{r}_{cm5/6}^6 = \begin{bmatrix} 0 \\ o \\ 0 \end{bmatrix} \tag{4.12}$$

The position of each center of mass in terms of and relative of the 0-frame is expressed as follows in equation 4.13

$$\begin{aligned} \boldsymbol{r}_{cm1/0}^0 &= \boldsymbol{r}_{1/0}^0 + \boldsymbol{R}_1^0\boldsymbol{r}_{cm1/1}^1, \quad \boldsymbol{r}_{cm2/0}^0 = \boldsymbol{r}_{2/0}^0 + \boldsymbol{R}_2^0\boldsymbol{r}_{cm2/2}^2, \\ \boldsymbol{r}_{cm3/0}^0 &= \boldsymbol{r}_{3/0}^0 + \boldsymbol{R}_3^0\boldsymbol{r}_{cm3/3}^3, \quad \boldsymbol{r}_{cm4/0}^0 = \boldsymbol{r}_{4/0}^0 + \boldsymbol{R}_4^0\boldsymbol{r}_{cm4/4}^4, \\ \boldsymbol{r}_{cm5/0}^0 &= \boldsymbol{r}_{6/0}^0 + \boldsymbol{R}_6^0\boldsymbol{r}_{cm5/6}^6, \end{aligned} \tag{4.13}$$

Each center of mass can then easily be expressed in terms of the 0-frame and relative to any other frame as follows in equation 4.14

$$\boldsymbol{r}_{cm_j/i}^0 = \boldsymbol{r}_{cm_j/0}^0 - \boldsymbol{r}_{i/0}^0 \tag{4.14}$$

4.1.4 J-matrix

One J-matrix is calculated for each center of mass. The matrix should be such satisfy the relation in equation 2.16.

The method used to calculate the J-matrix is from Rokseth (2014). The method involves defining a unit vector that each link revolve about. These unit vectors is then as follows in equation 4.15, see also figure 4.4 for reference.

$$\boldsymbol{e}_1 = \boldsymbol{R}_1^0\boldsymbol{k}, \quad \boldsymbol{e}_2 = \boldsymbol{R}_2^0\boldsymbol{i}, \quad \boldsymbol{e}_3 = \boldsymbol{R}_3^0\boldsymbol{i}, \tag{4.15}$$

Where the vectors $\boldsymbol{k}$ and $\boldsymbol{i}$ is the unit vector in z and x direction.

The J-matrices is then calculated as shown in equation 4.16.

$$\begin{gathered}
J_1 = \begin{bmatrix} [e_1 \times \boldsymbol{r}^0_{cm1/1}] & [0_{3\times2}] \\ [e_1] & [0_{3\times2}] \end{bmatrix}, \\
J_2 = \begin{bmatrix} [e_1 \times \boldsymbol{r}^0_{cm2/1}] & [e_2 \times \boldsymbol{r}^0_{cm2/2}] & [0_{3\times1}] \\ [e_1] & [e_2] & [0_{3\times1}] \end{bmatrix}, \\
J_3 = \begin{bmatrix} [e_1 \times \boldsymbol{r}^0_{cm3/1}] & [e_2 \times \boldsymbol{r}^0_{cm3/2}] & [e_3 \times \boldsymbol{r}^0_{cm3/3}] \\ [e_1] & [e_2] & [e_3] \end{bmatrix}, \\
J_4 = \begin{bmatrix} [e_1 \times \boldsymbol{r}^0_{cm4/1}] & [e_2 \times \boldsymbol{r}^0_{cm4/2}] & [0_{3\times1}] \\ [e_1] & [e_2] & [0_{3\times1}] \end{bmatrix}, \\
J_5 = \begin{bmatrix} [e_1 \times \boldsymbol{r}^0_{cm5/1}] & [e_2 \times \boldsymbol{r}^0_{cm5/2}] & [e_3 \times \boldsymbol{r}^0_{cm5/3}] \\ [e_1] & [e_2] & [e_3] \end{bmatrix}
\end{gathered} \tag{4.16}$$

4.1.5 Mass-Inertia matrices

The M-matrix is calculated for each center of mass. The matrix will be calculated as shown in equation 2.17 and explained in Rokseth (2014).

The mass of body i is denoted m_i and the inertia tensor for body i expressed in terms of reference frame j is denoted $\boldsymbol{I}_i^j$. The rotational matrices is used to transform the inertia tensor from the local reference frame to the 0-frame as shown in equation 4.17.

$$\boldsymbol{I}_1^0 = \boldsymbol{R}_1^0\boldsymbol{I}_1^1\boldsymbol{R}_0^1,\ \boldsymbol{I}_2^0 = \boldsymbol{R}_2^0\boldsymbol{I}_2^2\boldsymbol{R}_0^2,\ \boldsymbol{I}_3^0 = \boldsymbol{R}_3^0\boldsymbol{I}_3^3\boldsymbol{R}_0^3,\ \boldsymbol{I}_4^0 = \boldsymbol{R}_4^0\boldsymbol{I}_4^4\boldsymbol{R}_0^4,\ \boldsymbol{I}_5^0 = \boldsymbol{R}_6^0\boldsymbol{I}_5^5\boldsymbol{R}_0^6, \tag{4.17}$$

Where $\boldsymbol{R}_j^i$ is the transposed matrix of $\boldsymbol{R}_i^j$.

The five mass-inertia matrices is then expressed as follows in 4.18.

$$M_i = \begin{bmatrix} [m_i\boldsymbol{I}_{3\times3}] & [0_{3\times3}] \\ [0_{3\times3}] & [\boldsymbol{I}_i^0] \end{bmatrix} \tag{4.18}$$

Where M_i is the mass inertia matrix for the i-th center of mass.

4.1.6 Bond Graph Imprementation

Recall from section 2.3.2 that the Lagrange equation on state space form can be directly implemented in the IC-element as the effort and flow corresponding to each generalized coordinate. The equation to implement in the IC-element are derived as shown in equations 2.20 and 2.21. The $\boldsymbol{B}(\boldsymbol{q})$-matrix is with the results from section 4.1.4 and 4.1.5 calculated as shown in 2.19, thus all calculations for the IC-filed are complete, see appendix D for details on how the calculations is performed with the use of Maple.

As explained in Rokseth (2014) the $\boldsymbol{B}$-matrix and the $\frac{\partial \boldsymbol{B}(\boldsymbol{q})}{\partial \boldsymbol{q}}$-matrices will often be a large expressions. This is also the case for this model. Implementing this expressions directly in the 20-Sim bond graph model will then prove difficult. One problem is that the editor in 20-Sim does not fully support expressions of the size that are the case of this model. An other problem is the impact on the simulation time large expressions as these will have. This is because it is required by the simulation model to evaluate these expresions for each simulation step. To avoid these problems a dll. file containing both the $\boldsymbol{B}$-matrix and the $\frac{\partial \boldsymbol{B}(\boldsymbol{q})}{\partial \boldsymbol{q}}$-matrices are created as an extension for the 20-Sim bond graph model. This is done by exporting the results of the Maple calculations included in appendix D to C-code as explained in Rokseth (2014). The dll. file is then assembled and compiled as explained in Controllab (2016). The Dll file will run the time consuming calculations much faster than the solver in 20-Sim is capable of, and the problem with the 20-Sim editor not supporting expressions of this extent is also avoided.

The 20-sim model will send a vector containing all relevant parameters and variables as an input to the dll-file, and receive a vector containing the values of the $\boldsymbol{B}$-matrix and the $\frac{\partial \boldsymbol{B}(\boldsymbol{q})}{\partial \boldsymbol{q}}$-matrices evaluated at the parameters and variabels in the input vector. As the dll. file extension in 20-sim only supports sending vectors and not matreices the $\boldsymbol{B}$-matrix and the $\frac{\partial \boldsymbol{B}(\boldsymbol{q})}{\partial \boldsymbol{q}}$-matrices will have to be reassembled in 20-sim. Refer to appendix E.2 for the exact syntax of the bond graph model IC-element.

The IC-element will have to be connected with three 1-junctions, each related to one of the generalised coordinates, see figure 4.5. This IC-element will now represent the kinematics of the crane base, both arms, and both actuators.

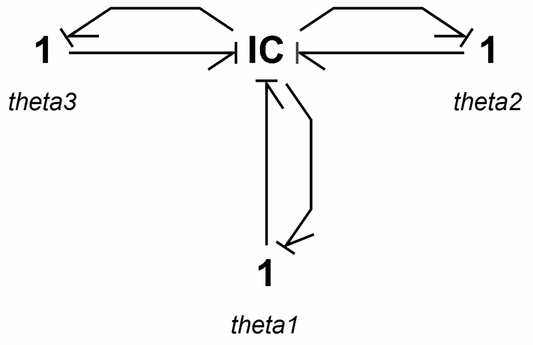


Figure 4.5: IC-element Bond Graph

4.2 System Interfacing

The crane dynamics can not alone be described by the kinematic model of the rigid bodies. Gravity will need to be added, and the actuators, base rotation and winch/wire interactions with the crane will have to be included. As shown in Rokseth (2014) any subsystem acting on the kinematic model (IC-element) can be interfaced as generalized forces, recall equation 2.13. This is done with a MTF-element between the 1-junction representing the generalised coordinates and a 1-junction representing a point p in the system. The system interacting with the kinematic model in point p can now be interfaced directly to the 1-junction representing point p.

The constitutive relations to implement in the MTF-element will then be as follows in equation 4.19

$$\begin{aligned} \boldsymbol{e}_q &= \boldsymbol{J}_p^T \boldsymbol{e}_p \\ \boldsymbol{f}_p &= \boldsymbol{J}_p \boldsymbol{f}_q \end{aligned} \tag{4.19}$$

Where $\boldsymbol{J}_p$ is the J-matrix calculated for the point p and $\boldsymbol{J}_p^T$ is the same J-matrix transposed. Note that if the velosities needed at point p is only the linear velocities then only the three first rows in the J-matrix should be used. If it is only the rotational velocities at point p that are of interest then only the three lower rows of the J-matrix should be used. $\boldsymbol{e}_q$ and $\boldsymbol{f}_q$ is the effort and flow at the generalised coordinates 1-junction, $\boldsymbol{e}_p$ and $\boldsymbol{f}_p$ is the effort and flow at the 1-junction representing the point p.

4.2.1 Gravity

The crane has a total of 5 moving bodies, each with its own center of gravity, see figure 3.9. The J-matrixes for each center of gravity is already calculated in section 4.1.4, see also appendix D for more details.

Figure 4.6 shows how the gravity submodel will be interfaced to the IC-element. The submodel *Calculation_of_Angles* takes the angular velocities $\dot{\theta}_1$, $\dot{\theta}_2$ and $\dot{\theta}_3$ and integrates them to calculate the angels θ_1 , θ_2 and θ_3, it also calculates the angles δ_1 and δ_2 with use of the equations 4.2 and 4.3 derived in subsection 4.1.2. Refer to appendix E.3 for the exact syntax of the *Calculation_of_Angles* submodel. These angles are sent as a signal as an input to MTF-elements. Note that only the three upper rows of the J-matrix will be used, as it is only the linear velocities at the center of gravity that are of interest.

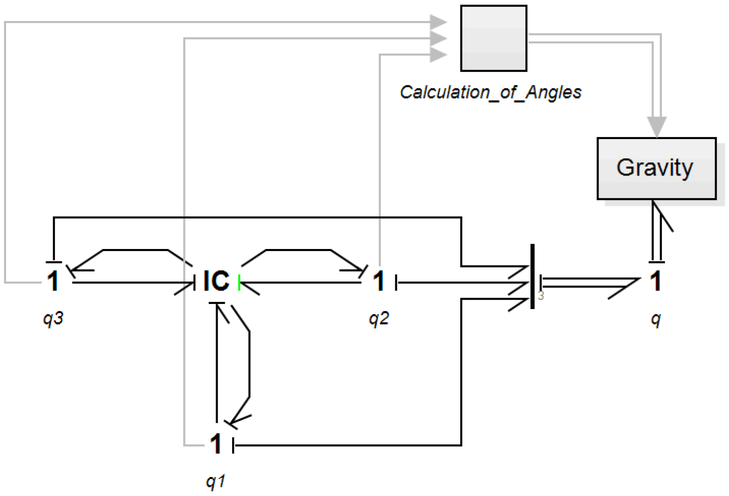


Figure 4.6: Gravity Submodel Interfaced with IC-element

Figure 4.7 show how the gravity submodel is built up. The MTF elements calculates the velocities for each center of gravity, CM1, CM2, etc, and the Se elements applies the gravitational force. Refer to appendix E.4 for the exact syntax of the MTF elements.

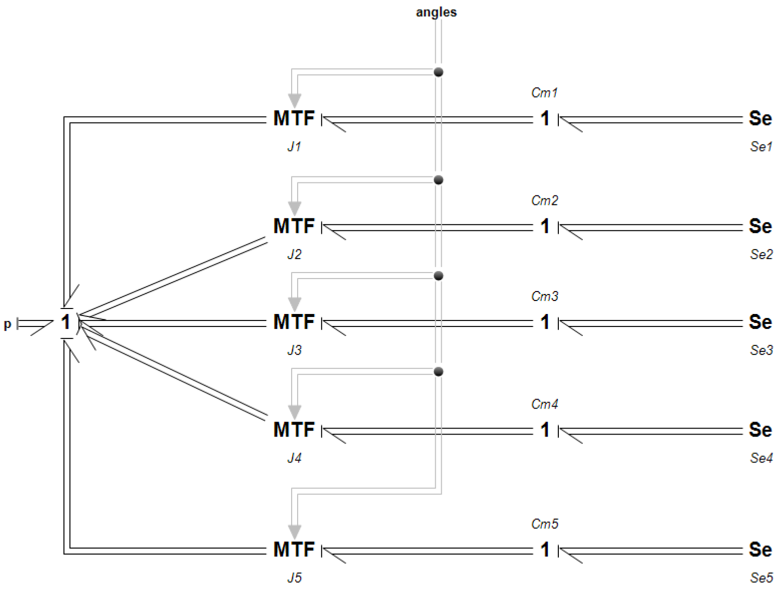
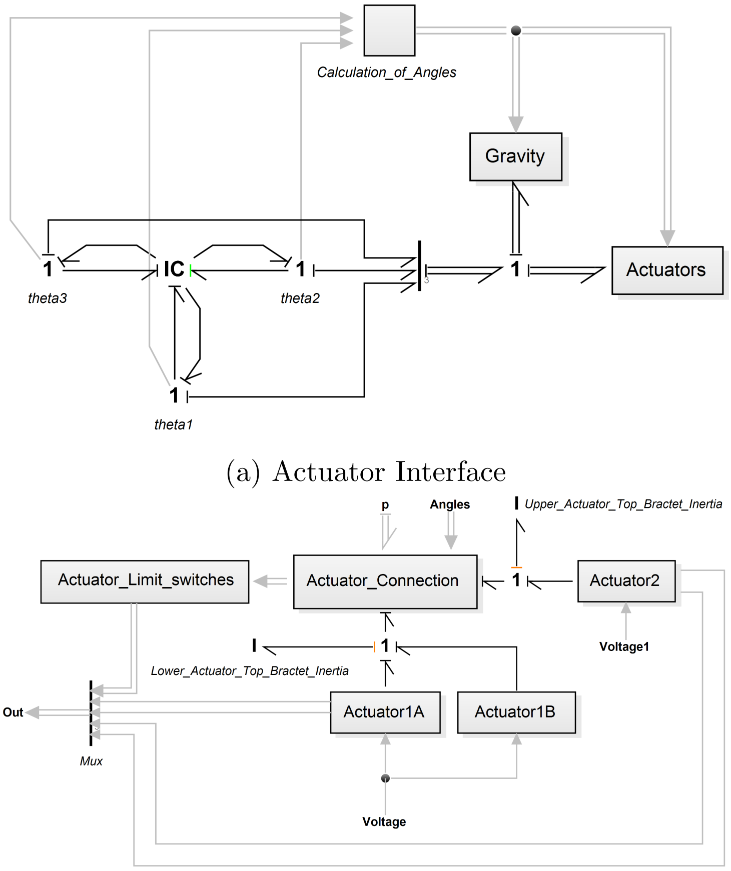


Figure 4.7: Gravity Submodel

4.2.2 Actuators

The actuators will be included in a submodel interfaced to the IC-element in the same manner as with the gravitational forces. Figure 4.8 illustrates the interfacing of the actuators.



(a) Actuator Interface

(b) Inside Actuator submodel

Figure 4.8: Actuators Submodel Interface

Actuator1A and Actuator1B represents the lower actuator assembly consisting of two actuators for the configuration with the long crane arm. More actuators can be implemented similarly as shown in figure 4.8b. When the crane configuration with the short lower arm is to be simulated Actuator1B has to be disconnected for the 1 junction joining the two lower actuators. Both Actuator1A and Actuator2 is sending out two signals that is needed for the control system. These signals are the rotational velocity and the current consumption of the actuator motors.

The actuator elongation will have to be limited to a fixed range dependent on the actuator design. The submodel Actuator_Limit_swhiches is used to calculate and keep track of the actuator elongation, and send out a signal if the actuator is at its boundaries. See appendix E.6 for the exact syntax of the Actuator_Limit_swhiches submodel.

The Actuator_Coection submodel is built up as shown in figure 4.9. The force from each actuator will result in an equal force at both the upper and lower connection point of the actuator. These connection points is defined as Actuator11 for the lower connection point of the lower actuator, and Actuator22 for the upper connection point of the upper actuator, etc. The integrator Actuator_pos_calculations calculates the position of each actuator joint used by the submodel Actuator_Limit_swhiches. See appendix E.5.7 for the exact syntax of the Actuator_pos_calculations submodel

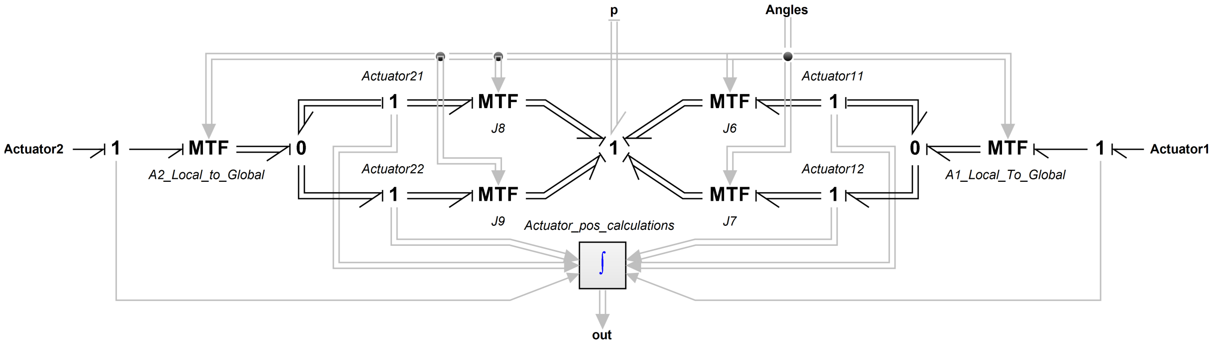


Figure 4.9: Actuator Conection Submodel

One J-matrix have to be calculated for each of the connection points and implemented in the MTF-elements. The lower connection point of both the lower and upper actuator is already defined as reference frame 4 and 6, ref. figure 4.4 and equation 4.8. The upper connection point for the lower and upper actuator is named frame 5 and 7 and defined as follows in equation 4.20

$$\boldsymbol{r}^2_{5/2} = \begin{bmatrix} 0 \\ b \\ -u \end{bmatrix} \quad \boldsymbol{r}^3_{7/3} = \begin{bmatrix} 0 \\ c \\ -s \end{bmatrix} \tag{4.20}$$

The position at the lower connection point of the actuators are already defined relative too, and in terms of the 0-frame as $\boldsymbol{r}^0_{4/0}$ and $\boldsymbol{r}^0_{6/0}$ in equation 4.11. The positions defined in equation 4.20 is expressed in terms of and relative to the 0-frame as follows in equation 4.21

$$\boldsymbol{r}^0_{5/0} = \boldsymbol{r}^0_{2/0} + \boldsymbol{R}^0_1\boldsymbol{R}^1_2\boldsymbol{r}^2_{5/2}, \quad \boldsymbol{r}^0_{7/0} = \boldsymbol{r}^0_{3/0} + \boldsymbol{R}^0_1\boldsymbol{R}^1_2\boldsymbol{R}^2_3\boldsymbol{r}^3_{7/3} \tag{4.21}$$

The frames can then be expressed in terms of the 0-frame and relative to any other frame as shown in equation 4.14

The four J-matrices are calculated in the same manner as in subsection 4.1.4. The unit vectors defined in equation 4.15 are reused. The J-matrices are expressed as follows in equation 4.22. See also appendix D for more details.

$$
\begin{aligned}
J_6 &= \begin{bmatrix} \begin{bmatrix} e_1 \times \boldsymbol{r}^0_{4/1} \\ e_1 \end{bmatrix} & \begin{bmatrix} 0_{3\times 2} \\ 0_{3\times 2} \end{bmatrix} \end{bmatrix}, \\
J_7 &= \begin{bmatrix} \begin{bmatrix} e_1 \times \boldsymbol{r}^0_{5/1} \\ e_1 \end{bmatrix} & \begin{bmatrix} e_2 \times \boldsymbol{r}^0_{5/2} \\ e_2 \end{bmatrix} & \begin{bmatrix} 0_{3\times 1} \\ 0_{3\times 1} \end{bmatrix} \end{bmatrix}, \\
J_8 &= \begin{bmatrix} \begin{bmatrix} e_1 \times \boldsymbol{r}^0_{6/1} \\ e_1 \end{bmatrix} & \begin{bmatrix} e_2 \times \boldsymbol{r}^0_{6/2} \\ e_2 \end{bmatrix} & \begin{bmatrix} 0_{3\times 1} \\ 0_{3\times 1} \end{bmatrix} \end{bmatrix}, \\
J_9 &= \begin{bmatrix} \begin{bmatrix} e_1 \times \boldsymbol{r}^0_{7/1} \\ e_1 \end{bmatrix} & \begin{bmatrix} e_2 \times \boldsymbol{r}^0_{7/2} \\ e_2 \end{bmatrix} & \begin{bmatrix} e_3 \times \boldsymbol{r}^0_{7/3} \\ e_3 \end{bmatrix} \end{bmatrix}
\end{aligned}
\tag{4.22}
$$

The three upper rows in the J-matrices in equation 4.22 are used as constitutive relations in the MTF-elements used to interface the points for the actuator connection to the crane. See appendix E.5.1 too E.5.4 for the exact syntax of these MTF-elements.

Since the force is equal at both the upper and lower connection point of the actuator a 0-junction is used to make the connection between the actuator connection points and the actuator itself. The force and velocity of the actuator will have to be decomposed in terms of the 0-frame in order to interface with the 0-junction. This is done with the MTF elements *"A2 Local to Global"* and *"A1 Local to Global"*, see figure 4.2, and appendix E.5.5 and E.5.6 for the exact syntax of these MTF-elements.

The submodels Actuator1A, Actuator1B and Actuator2 is modeled identical. This is done with reference to the conceptual design which shows that the internal components of the actuators will consist of the same type of parts, see figure 3.7. The actuator submodel is created as a regular bond-graph model as shown in figure 4.10.

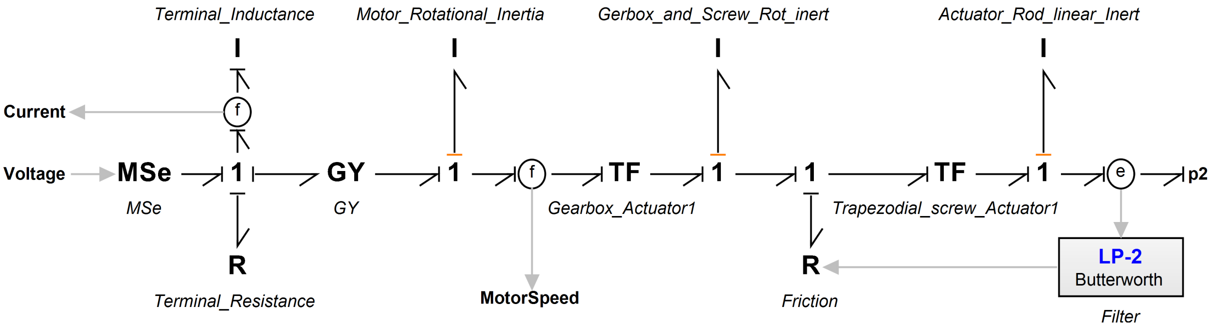


Figure 4.10: Actuator Bond-Graph

The Gyrator element represents the properties of the electric motor. This means that the 1-junction to the left side of the gyrator represents the electrical side of the motor, and the 1 junction to the right of the gyrator represents the mechanical rotation of the electric motor. The model is set up to take in a voltage and send out a signal corresponding to the rotational velocity of the motor and a signal corresponding to the current consumption of the motor.

To the right of the 1-junction representing the mechanical side of the electrical motor is a TF-element. The TF-element represents the gearbox connected to the motor. The energy loss in the gearbox will not be modeled with the use of a separate bond graph element with a friction model. This is because it is found to complicate the model unnecessarily. Instead the efficiency, η, of the gearbox, which will be provided by the gearbox vendor, is added to the TF-element so that the constitutive relation of the gearbox TF-element is as shown in equation 4.23

$$\begin{aligned} e_1\eta &= me_2 \\ f_2 &= mf_1 \end{aligned} \tag{4.23}$$

The 1-junction to the right of the gearbox TF-element represents the trapezoidal screw. The friction in the trapezoidal screw is modeled as a combination of static, μ_s, coulomb, μ_c, and viscous, μ_v, friction where the friction force/torque is given as shown in equation 4.24. The friction model in equation 4.24 is from Pedersen and Engja (2010).

$$F_f = F_N\left(\mu_s \tanh(sv) - \frac{1}{2}(\mu_s - \mu_c)\tanh\left(s\left(v - \frac{6.5}{s}\right)\right) + \mu_v v\right) \tag{4.24}$$

The parameters μ_s, μ_c and μ_v is the static, coulomb, and viscous friction parameters respecivly. F_N is the normal force, v is the velocity, and s is the slope. The slope determines how fast the screw must rotate in order to overcome the ”static” friction, larger s yields lower velocity to overcome the static friction.

The friction model is implemented in the MR-element which takes the normal force, F_N, as input.

The next TF-element transforms the rotation of the trapezoidal screw to linear translation of the actuator rod.

4.2.3 Rotation of the Base

The rotation of the base is implemented as a submodel connected to the IC-element as shown in figure 4.11. It is not necessary to define a point where the base rotation interacts with the IC-field as the rotation of the base is the generalized coordinate θ_1.

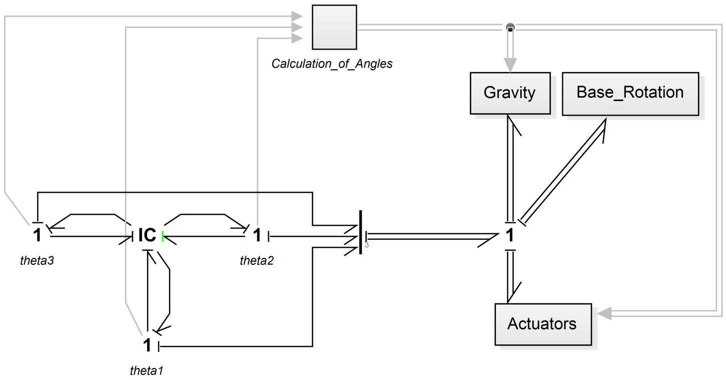


Figure 4.11: Base Rotation Interfacing

The base rotation submodel is created as a regular bond-graph model very similarly to the actuator submodel, the structure of the submodel is shown in figure 4.12. The only difference from the actuator sub model is to the right of the 1-junction representing the rotation out of the gearbox. The TF element "Slewing_Ring" is representing the gear connection between the motor and the slewing ring, and the friction element models the friction in the slewing ring. The friction of the slewing ring is modeled by the same model as the trapezoidal screw friction, see equation 4.24. The only difference in the friction models is that the normal force in the trapezoidal screw is considered to be constant. This is a simplification, but it is assumed that the friction can be modeled quite accurately with the normal force as a static parameter. The tree 1-junctions at the left side of the model is only a split of the velocities at the generalized coordinates from a vector bond to single power bonds.

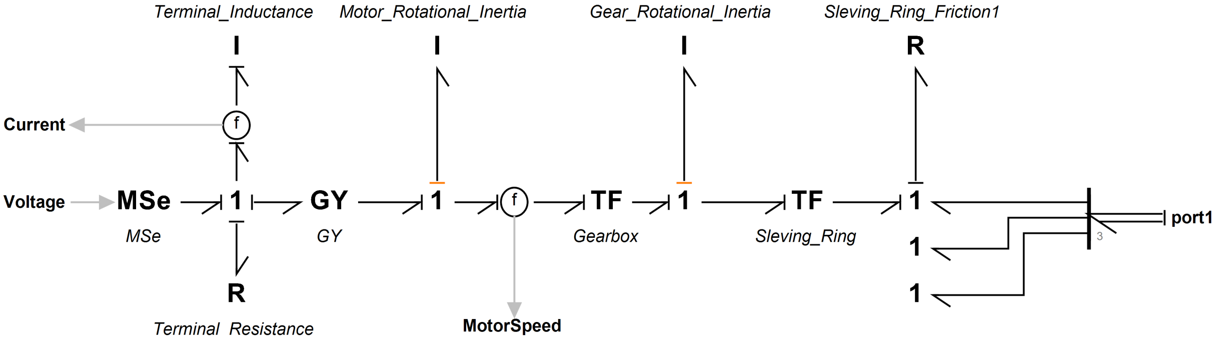


Figure 4.12: Base Rotation Submodel

4.2.4 Wire and Winch

A model for the wire and winch will be interfaced to the rest of the simulation model following the same principle as the actuators and the gravity submodel. This means that it is necessary to identify where the force from the wire interfaces with the crane. Figure 4.13 show the wire geometry and the geometry of the forces related to the wire. The force F_l is the force from the lower arm on the wire, and the force F_u is the force from the upper arm on the wire. This means that the forces from the wire on the crane will be equal in magnitude but opposite in direction of F_l and F_u. The force F_w is the tension force in the wire.

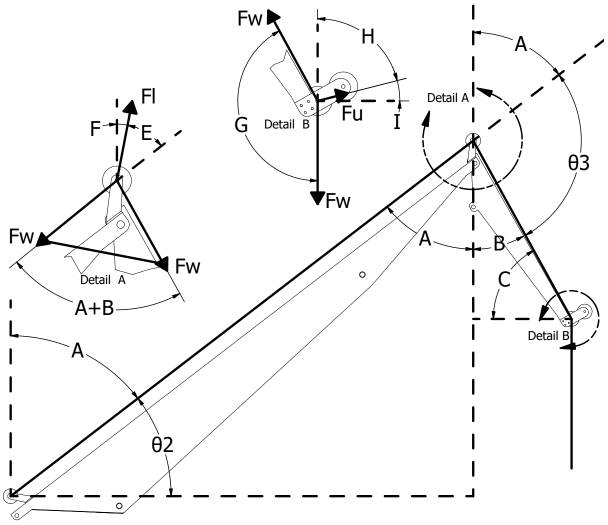


Figure 4.13: Wire Geometry

By studying figure 4.13 it is clear that it will be sufficient to interface the force from the wire at the tip of the upper crane arm. This can easiest be seen if one imagine how the geometry is if $\theta_2 = 0°$ and $\theta_3 = 0°$. The force F_l will then be zero. The same is the case if $\theta_2 = 0°$ and $\theta_3 = -90°$, the force F_u will then be zero. This means that the torque from the wire at θ_2 is given by the force in the wire multiplied by the horizontal distance from the lower connection of the lower arm to the crane tip. The torque from the wire at θ_3 is given by the force in the wire multiplied by the horizontal distance from the lower connection of the upper arm to the crane tip. The force from the wire will therfore be interfaced as a generalized force acting on the crane tip. The coordinates of the crane tip is then defined as reference frame 8, and has the coordinates shown in equation 4.25.

$$\boldsymbol{r}^3_{8/3} = \begin{bmatrix} 0 \\ L_2 \\ 0 \end{bmatrix} \tag{4.25}$$

The coordinates of the crane tip are then expressed relative to and in terms of the 0-frame as follows in equation 4.26, and relative to any other frame as shown in equation 4.14

$$\boldsymbol{r}^0_{8/0} = \boldsymbol{r}^0_{3/0} + \boldsymbol{R}^0_1\boldsymbol{R}^1_2\boldsymbol{R}^2_3\boldsymbol{r}^3_{8/3} \tag{4.26}$$

The J-matrix for the tip of the crane is then calculated as shown in equation 4.27

$$J_{10} = \begin{bmatrix} [e_1 \times \boldsymbol{r}^0_{8/1}] & [e_2 \times \boldsymbol{r}^0_{8/2}] & [e_3 \times \boldsymbol{r}^0_{8/3}] \\ [e_1] & [e_2] & [e_3] \end{bmatrix} \tag{4.27}$$

The theory used to create the wire and winch model is based on Skjong (2014). The basis of this theory is that the simulation model of the wire will be build up by a number of mass-spring-damper models linked together as pendulums. The chain of pendulums, or wire elements, will then quite accurately simulate the wire. A total of 5 wire elements have been implemented in the simulation model at this stage, the model can easily be expanded to include more elements if a higher accuracy is required. The complete model and the interfacing of the wire and winch model to the the rest of the simulation model is illustrated in figure 4.14 and 4.15

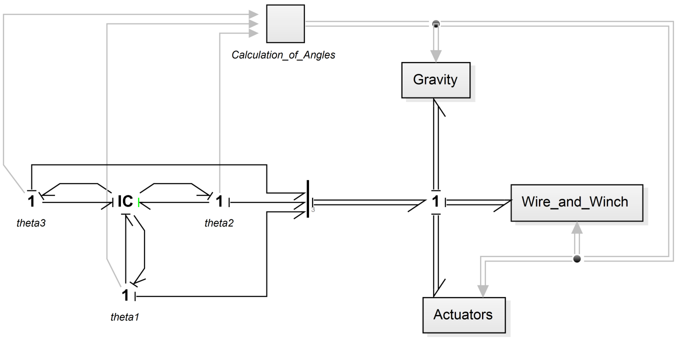


Figure 4.14: Wire and Winch Interface

The J-matrix calculated in equation 4.27 is implemented in the MTF-element at the top right of figure 4.15. See appendix E.7.1 for the exact syntax of the MTF-element. This makes the 1-junction to the left of the MTF-element a representation of the crane tip. The 1-junction at the bottom of figure 4.15 represents the velocities of the load hanging from the wire. Both the inertia and the gravitational force at the load is included. The position and acceleration of both the crane tip and load is calculated by the integrating and derivating block diagram elements respectively. The positions is sent out of the submodel to be used in a potential control system controlling the crane tip and load position. The winch is connected to the wire elements by signal bond connections as explained in Skjong (2014). The winch model sends out a total of three signals, current, velocity and amount of wire on the reel. These signals will be used by the control system for winch. The wire length is also sent out of the model as a signal to be used in the control system.

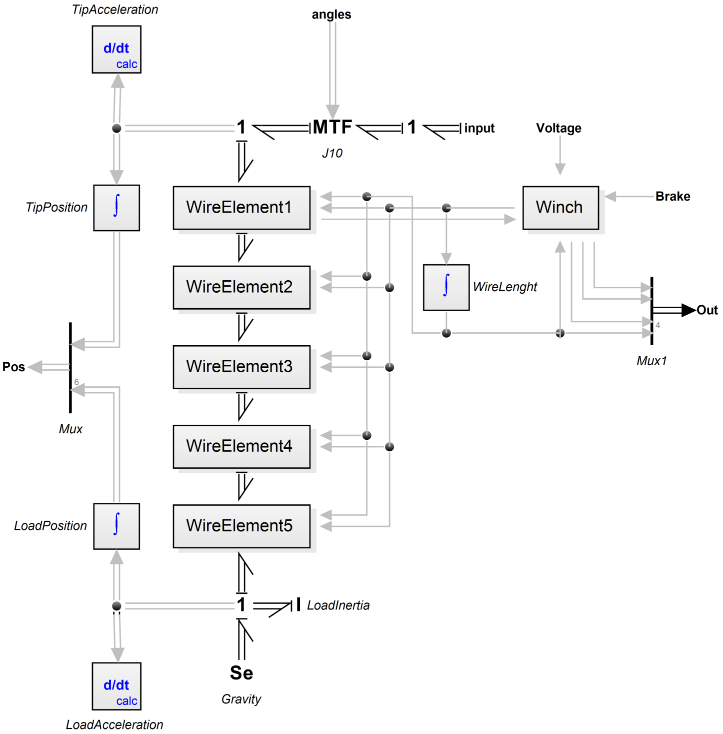


Figure 4.15: Wire and Winch Submodel

Each wire element will be built up as shown in figure 4.16, this is done similarly to the model explained in Skjong (2014).

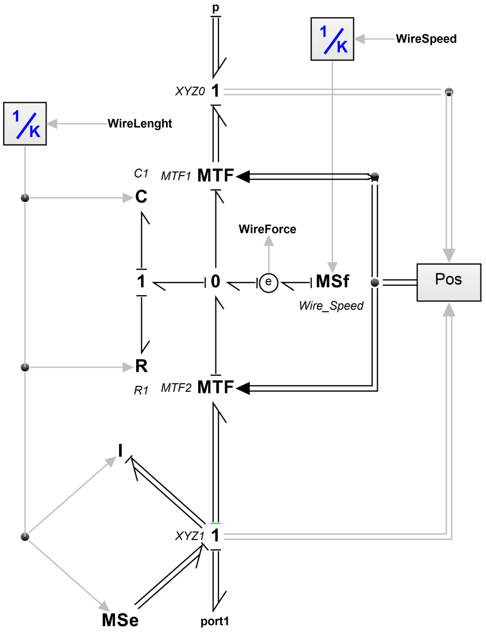


Figure 4.16: Wire Element Bond Graph

The 1-junction XYZ0 represents the top part of the wire element, and the 1-junction XYZ1 represents the bottom part of the wire element. At each of these one junctions the effort and flow is decomposed in terms of the common reference frame, e.g. global XYZ coordinates. The effort and flow at the 0-junction connecting the top and bottom of the wire element 1-junctions is expressed in local wire coordinates. This means that the flow at the 0 junction is related to speed of the wire elongation, and the effort at the 0-junction is related to the tension in the wire. The two MTF-elements transforms from the global XYZ coordinates to the local coordinates, this is done as shown in Skjong (2014). See also appendix E.8.1 and E.8.2 for the exact syntax of the MTF-elements, and appendix E.8.3 for the exact syntax of the Pos submodel. The wire element is modeled as a spring-damper system by the C-element, C1, and the R-element, R1 on order to include the elastic properties of the wire. The spring stiffness and the damping coefficient is implemented as shown in Skjong (2014). The MSf-element is setting the speed of the hoisting or lowering of the wire according to the speed of the wire in the winch model, the tension force in the wire is sent back to the winch model. The inertia and weight of the wire element is included as the MSe-element, and I element connected to the 1-junction representing the bottom of the wire element.

The structure of the winch submodel is illustrated in figure 4.17.

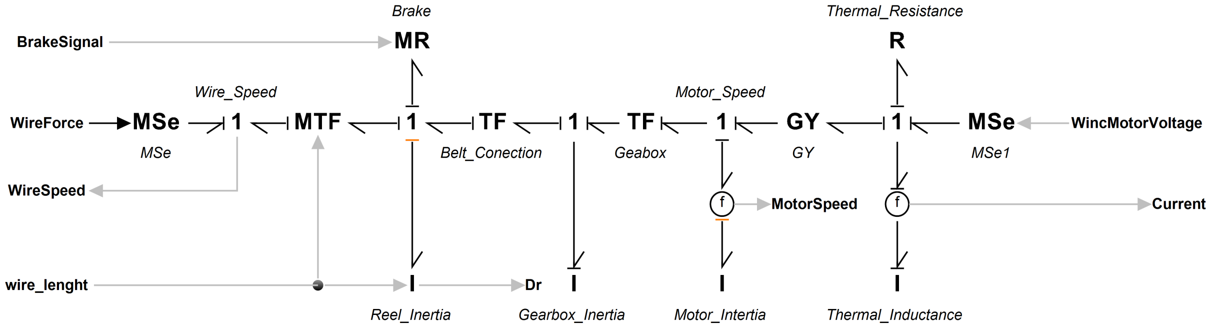


Figure 4.17: Winch Submodel

The first 1-junction from the left represents the speed of the wire out of the reel. The tension force given by the wire element submodel is implemented here as a MSe-element, and the wire speed is sent back to the wire element submodel. The transformation from the rotational velocity of the reel to the linear velocity of the wire is done by the MTF element. The amount of wire stored on the reel will have to be accounted for as it will change the effective radius of the reel, and thus the transformer modulus of the MTF-element. The effective diameter of the reel as a function of the amount of wire stored on the reel is given by equation 4.28, from Skjong (2014).

$$D(L_{wr}) = \sqrt{\frac{L_{wr} D_w^2}{w_r f_w} + D_{r0}^2} \tag{4.28}$$

The 1-junction to the right of the MTF element represents the rotational speed of the reel. The I-element connected will then represent the rotational inertia of the reel with all conected parts and amount of wire stored on the reel. The inertia of the wire stored on the reel will depend on the amount of wire stored on the reel and is given by equation 4.29, from Skjong (2014).

$$I_w(L_{wr}) = \frac{1}{2}\rho_w L_{wr}\left(\frac{D(L_{wr})^2 - D_{r0}^2}{4}\right) \tag{4.29}$$

The MR-element connected to the 1-junction representing the reel rotational velocity represents the barking torque supplied by the attached disk brake. The braking torque is modeled as a frictional model consisting of static, coulomb and viscous friction as with the trapezoidal screw and the slewing ring, see equation 4.24. The friction modeled for the brake is however modified to take in a parameter varying from 0 to 1 which represents the amount of force supplied to the brake.

The TF element named Belt_Conection represents the reduction in the belt transmission between the gearbox and the reel. The friction in this transmission will be included as an efficiency in the transmission. The rest of the model is modeled the same way as the motor and gearbox model in the actuator and base rotation bond-graph models.

4.3 Controll System

A basic control system is developed in order for the simulation model to produce any simulation results. The control system is set up as PID speed controllers with current limitation functionality connected to the four electric motors powering the movement of the crane.

A PID position controller intended to control the position of the crane tip relative to a reference signal has also been implemented. This controller is an unaltered version of the PID controller included in the 20-Sim library, see Controllab (2016) for reference.

The structure of the complete simulation model with the control system implemented is illustrated in figure 4.18. Note that a submodel named *Parameters* is also included. This submodel stores all the parameters for the complete simulation model, see appendix E.1 for the exact syntax.

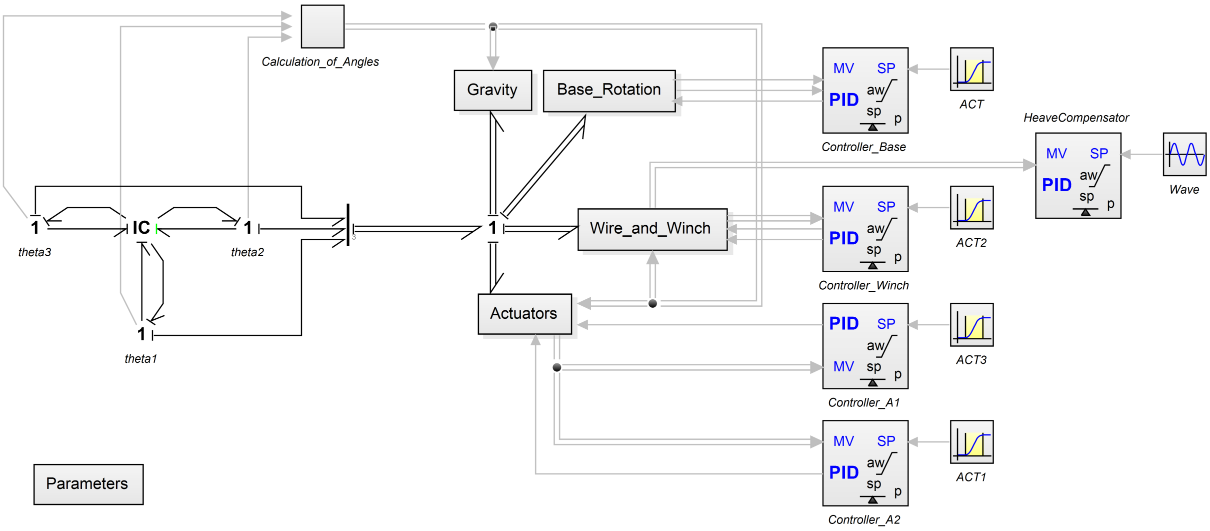


Figure 4.18: Complete Simulation Model with Control System

4.3.1 Motor Controllers

The four controllers for the electric motors are more or less implemented identically. They all receive signals corresponding the current consumption and rotational velocity of the relevant motor, and a set point signal corresponding the to the desired speed of the relevant motor. They are also all sending out a signal corresponding the motor voltage. The controllers that are running the actuators are in addition taking in stop signals from the Actuator_Limit_swhiches submodel. The controller running the winch motor and brake is in addition receiving signals related to the wire length and amount of wire on the reel, and is also sending out a signal corresponding to the force at the brake pads of the bake attached to the reel. A signal generator from the 20-Sim library or the position controller is used to give the desired motor speed.

The speed controllers are built up with basis in the PID controllers provided in the 20-Sim library, see Controllab (2016) for reference.

A current limiting function is implemented in all the motor controllers. The current limiter will if the current consumption of the relevant motor is to large reduce the error in the PID controller, thus reducing the gain of the controller. In order for the current limiter to have a continues but responsive limiting ability a slightly modified version of the logistic function from Weisstein (2016) is used. The current limiter is thus modeled by equation 4.30

$$CL = \frac{CL_{Gain}}{1 + \mathrm{e}^{-CL_{Incline}(C - C_{Limit})}} \tag{4.30}$$

The values for the current limiter is altered during the iterations of the crane design ending up with the values shown in table 4.1 and illustrated in figure 4.19

Table 4.1: Current Limiter Parameters

Parameter	Value
CL_{Gain}	1
$CL_{Incline}$	20
C_{Limit}	Current Limit of the Motor

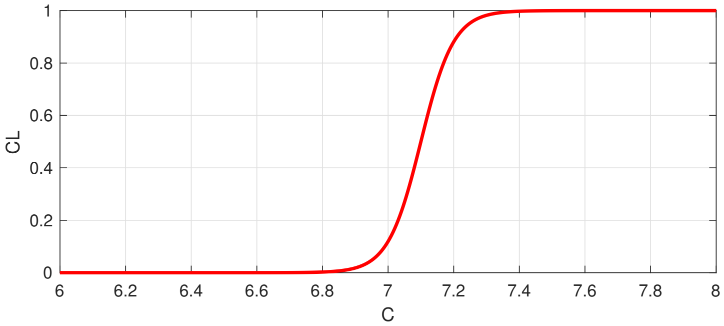


Figure 4.19: Current Limiter Behvour, $C_{Limit} = 7.1$

The main control algorithm common for all motor controllers is implemented as shown below.

```
if Current>=0 then
    CurrentLimiter=CLGain/(1+exp(-CLIncline*(limit(abs(Current),0,10)-CL)));
  else
    if Current<0 then
      CurrentLimiter=-CLGain/(1+exp(-CLIncline*(limit(abs(Current),0,10)-CL)));
    end;
  end;
  MVCorr=((MV/(2*pi))*60)/MaxRpm;
  if time < 0.99 then
    error=0;
  else
    error = SPCorr - MVCorr-CurrentLimiter;
  end;
  uP = K * error ;
  uI =limint ((K * error / Ti), minimum, maximum);
  uDstate = int ( uD * N / Td );
  uD = K * error * N - uDstate;
  ideal_output = (uP + uI + uD);
  output = (limit (ideal_output, minimum, maximum));
```

The controllers that run the actuators have in addition to the main control algorithm an algorithm that corrects the set point signal with reference to the stop signal. This algorithm is implemented as shown below.

```
  if (stopA11==0 and stopA12==0) then
    SPCorr=SP;
```

```
end;
if (stopA11==1) then
   SPCorr = limit(SP,0,1);
end;
if (stopA12==1) then
   SPCorr = limit(SP,-1,0);
end;
```

The controller that run the winch motor and brake have in addition to the main control algorithm an algorithm that corrects the set point signal with reference to the wire length and amount of wire on the reel. This algorithm is implemented as shown below.

```
if (WireLenght<=0.1) then
   SPCorr = limit(SPCorr,0,1);
end;
if (Dr<=Dr0) then
   SPCorr = limit(SPCorr,-1,0);
end;
```

A controller for the actuation of the brake is also included in the winch motor and brake controller. This is basically the same PID controller that is used for the motor, but with one significant difference. The error, or ”BrakeCondition” in this case, is calculated so that the ”BrakeCondition” is positive when the speed of the winch is decreasing. This and by making the controller only output a positive signal will make the controller only apply the brake when the winch is trying to decrease the speed.

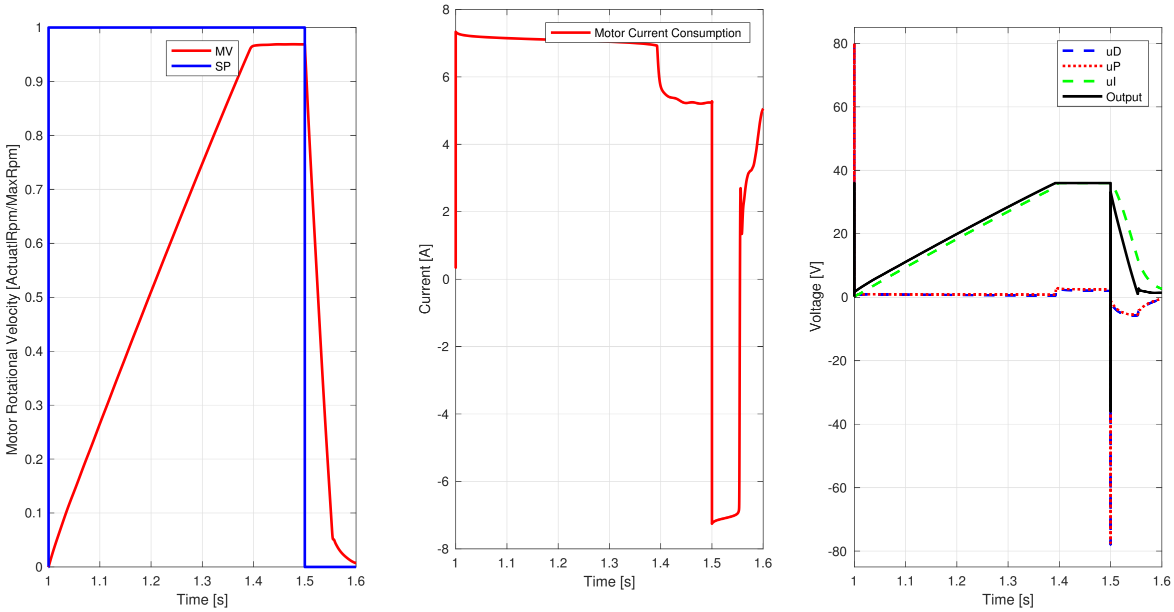
```
BrakeCondition=abs(MVCorr)-abs(SPCorr);
uP_B = K_B * BrakeCondition;
uI_B = limint ((K_B * BrakeCondition / Ti_B),minimum_B,maximum_B);
uDstate_B = int ( uD * N_B / Td_B );
uD_B = K_B * BrakeCondition * N_B - uDstate_B;
ideal_output_B = (uP_B + uI_B + uD_B);
Brake = limit (ideal_output_B, minimum_B, maximum_B);
```

The parameters of the PID controller is altered several times during the iterations of the crane design ending up with the values shown in table 4.2.

Table 4.2: Controller Parameters

Parameter	Value
K, K_B	80
Ti, Ti_B	0.01
N, N_B	1
Td, Td_B	1
$MaxRpm$	Max rpm of motor
$minimum$	Minimum Motor Voltage
$maximum$	Maximum Motor Voltage
$minimum_B$	0
$maximum_B$	1

A plot of a pulse response for the lower actuator showing how the controller is performing is presented in figure 4.20. The controllers for the other motors is performing similarly.



(a) Motor Response

(b) Motor Current Consumption

(c) Controller Beaviour

Figure 4.20: Lower Actuator Controller Pulse Response

The pulse response illustrated in figure 4.20 show that the motor is accelerating with maximum current consumption (7.1A for this test) during the complete acceleration period. The motor is then supplied with maximum voltage (36V for this test) until the signal is demanding the motor to stop. The motor is then decelerating with the maximum negative current consumption (-7.1A for this test) until the motor is almost standing still. This means that the motor is operating at its limits for the complete pulse response, and the motor is thus operating as responsively as possible. This analysis concludes that the proportional, integrating or derivative gain of the controller does not need to be set any higher.

4.4 3D Simulation

The 3D simulation tool in 20-Sim is used to visualise the results from the simulation model. It is often easier to understand how the crane will perform by using a 3D simulator, rather then just reading 2D plots. Figure 4.22 show how the 3D simulation is displayed. A short video of a 3D simulation is also included in the electronic attachments submitted with this thesis, see appendix A for an overview of the electronic attachments.

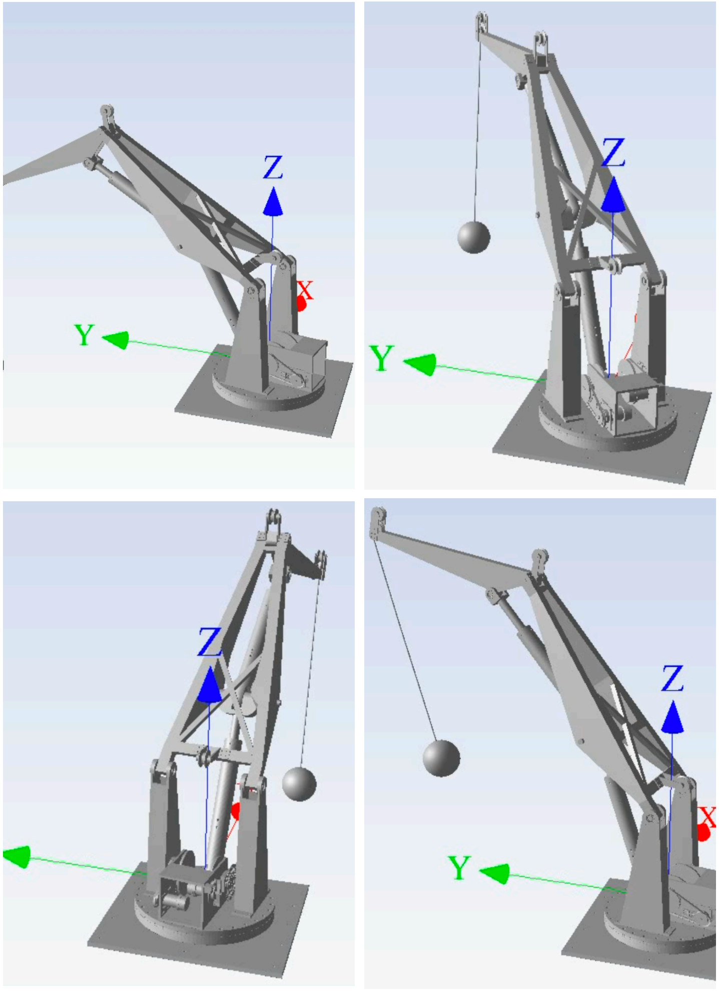


Figure 4.22: 3D Simulation of the Crane

Chapter 5

Final Crane Design

The final design of the crane is as explained in section 2.2 a result of an iteration/optimisation process. Parameters related to geometry, materials, choice of components, etc. has been investigated thoroughly before the final crane design is determined. The goal is to get a crane that performs well with regard to the framework presented in section 3.1. The simulation model presented in chapter 4 has been used simultaneously as the design has been developed and analyzed in Audodesk Inventor to test and verify different design parameters. Most of the design parameters will primarily be optimized for the crane configuration with the long arm, as this configuration will be the most demanding for all the parts that is common for the two arm configurations. The result of the design process is presented in the following sections.

5.1 Material Selection

Different materials will be used for different parts of the crane, these are determined during the structural analysis of each part which is presented in the sections to come. There will in total be used three material types in the crane parts that are produced locally. A presentation of the different materials and their properties is presented in table 5.1, 5.2 and 5.3.

Table 5.1: Aluminium Plates and Solids

Parameter	Value	Unit
Material Specification	EN AW 5754 - H22	
Density	2.66	g/cm^3
Modulus of Elasticity	68	GPa
Yield Strength	130	MPa
Tensie Strenght	270	MPa

Table 5.2: Aluminium Extruded Sections

Parameter	Value	Unit
Material Specification	EN AW-6060	
Density	2.7	g/cm^3
Modulus of Elasticity	69	GPa
Yield Strength	140	MPa
Tensie Strenght	170	MPa

Table 5.3: All Steel Parts

Parameter	Value	Unit
Material Specification	EN 1.4418, QT 900	
Density	7.7	g/cm^3
Modulus of Elasticity	200	GPa
Yield Strength	750	MPa
Tensie Strenght	1100	MPa

5.2 Structural Analysis Configuration

The structural analysis of the crane is evaluated with basis in the principle that the structure should be able to withstand a much greater load than the design load as stated in section 3.1. The structural analysis of the the crane is performed simultaneously as the design is determined. This is accomplished as explained in section 2.4 with the use of the FEA-tool in Autodesk Inventor.

A configuration and a loading condition is needed in order to evaluate the strength of the crane structure. The crane is configured with the long lower arm for the complete strength analysis, this is because this configuration will yield the highest loads. The crane is set up as shown in figure 5.1.

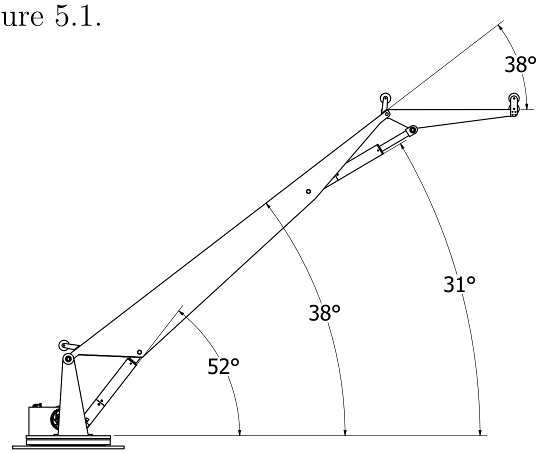


Figure 5.1: Crane Configuration for Structural Analysis

The design load that the crane is to lift is 16 kg which is approximately 157 N. To insure that the crane can withstand a much greater load than the design load and that the crane is able to withstand the dynamic loads the crane will be subjected to during movement of the crane arms and hoisting and lowering with the winch, a load of 500N is applied at the end of the upper arm during the strength analysis of the general crane structure.

The crane will not be analyzed as a complete assembly as it will be difficult to verify the analysis and extract the reaction forces between the different moving parts. The crane is thus divided into 6 parts: Upper arm, upper actuator, lower arm, lower actuator, base, and reel assembly.

5.3 Analysis of Actuator Geometry

Most of the parameters associated with the crane geometry are related to the placement of the actuators, ref. figure 3.9. These parameters could possibly influence both the movement range of the crane and the required force from the actuators necessary to be able to move the crane arms. This means that in order to develop a crane with a sensible design it is necessary to understand how the different parameters influence the crane performance. The simulation model presented in chapter 4 is used to accomplish this.

This parameter analysis will be most applicable for the crane configuration with the long arm. This is because this configuration is assumed to be the most demanding for all the parts that will be common for the two crane configurations. This analysis will therefore mostly cover the crane configuration with the long lower arm.

An initial "guess" on the parameters in the simulation model will have to be made in order for the simulation model to produce any simulation results. The initial parameters are based on simple sketches (ref. figure 3.9) and the design presented in Gyberg (2015). The initial parameters are presented in table 5.4.

Table 5.4: Initial Parameters Long Arm

Parameter	Size	Unit
L_1	2900	mm
L2	1000	mm
h	450	mm
a	100	mm
b	500	mm
u	300	mm
r	150	mm
w	2000	mm
c	200	mm
s	150	mm

Parameter	Size	Unit
CM1	50	kg
Ycm1	0	mm
Xcm1	0	mm
CM2	15	kg
k	500	mm
CM3	5	kg
n	400	mm
CM4	5	kg
p	400	mm
CM5	5	kg
o	400	mm

The parameters for the winch, wire, actuator, and rational inertias of the arms are not important for the analysis done in this section, the value of these parameters are therefore set without much consideration, and can be chosen quite randomly for the next simulations.

A series of simulations have been performed with varying values of the parameters concerning the connection points of the actuators in order to gain an understanding on how these parameters influence the crane.

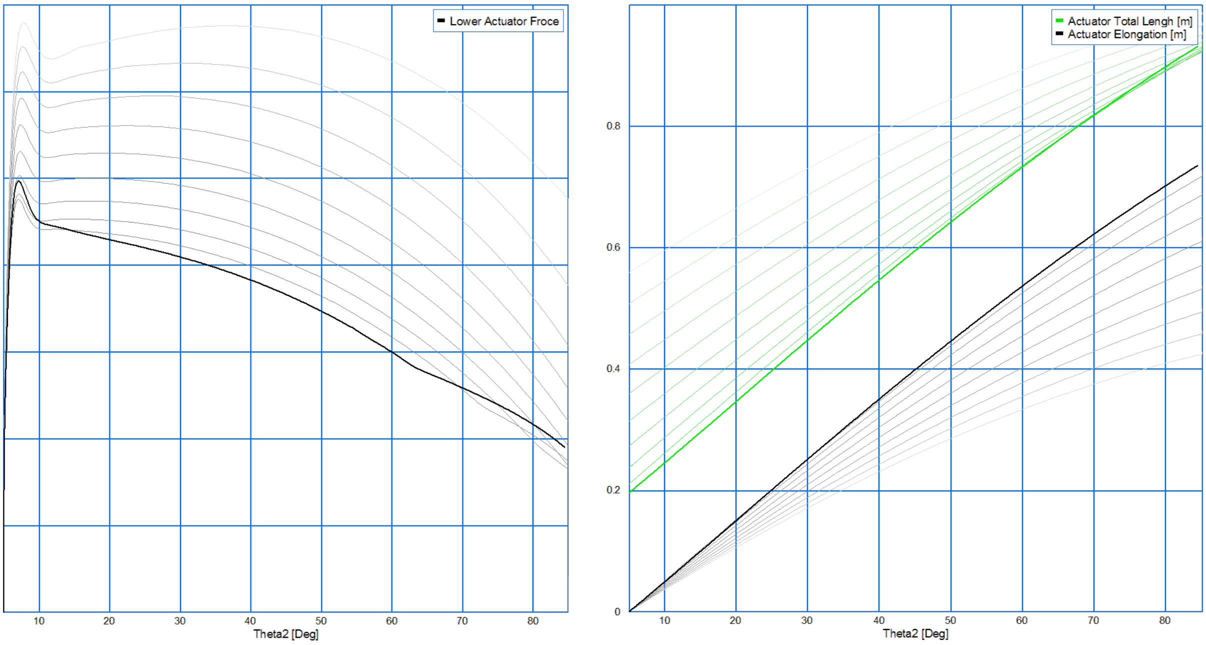
The simulation results concerning the lower actuator are obtained by lifting the lower arm from 5° to 85° with a constant actuator velocity, and with the upper arm set at the angle $\theta_3 = -45°$.

The simulation results concerning the upper actuator are obtained by lifting the upper arm from $-100°$ to $-10°$ with a constant actuator velocity, and with the lower arm set at the angle $\theta_2 = 45°$.

Two types of graphs are presented in the following simulation results. One is a graph showing the required force of the actuator as a function of the angle at the relevant arm. The other is a graph showing the required total length of the relevant actuator and the required actuator elongation, both as functions of the angle at the the relevant arm. The force graphs show only the relative magnitude of the required force. This is because the required force is greatly influenced by the weight and inertia of the arms which is not yet determined. The required force can therefore not be determined with a definite magnitude at this stage.

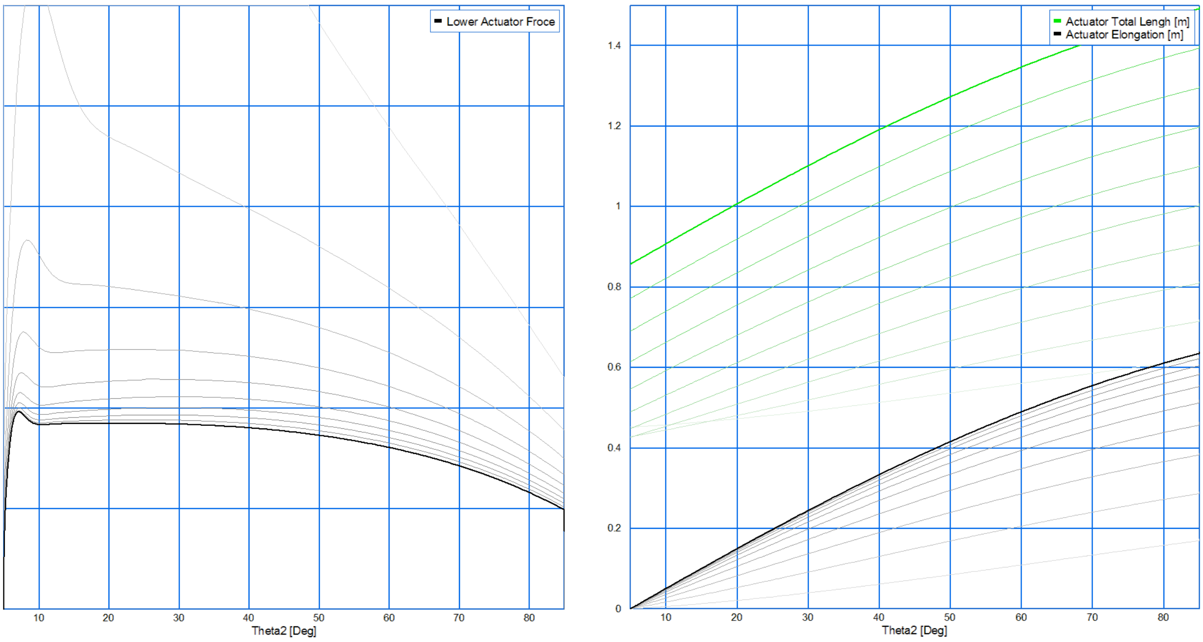
5.3.1 Placement of Lower Actuator at the Base

The connection point for the lower actuator relative to the lower arm connection at the base is determined by the two parameters h, and a. Figure 5.2 and 5.3 show the required actuator force (figure 5.2a and 5.3a), and the required length and elongation of the lower actuator (figure 5.2b and 5.3b) as a function of θ_2. The parameter a is set to start at 0m and increase by 0.05m for each simulation step. The parameter h is set to start at 0.1m and increase by 0.1m for each simulation step. There are in total 10 simulation steps presented in the figures 5.2 and 5.3. The graphs with the lightest colour are for the smallest values of b and u.



(a) Required Force from Actuator (b) Actuator Length and Elongation.

Figure 5.2: Variation of Parameter a



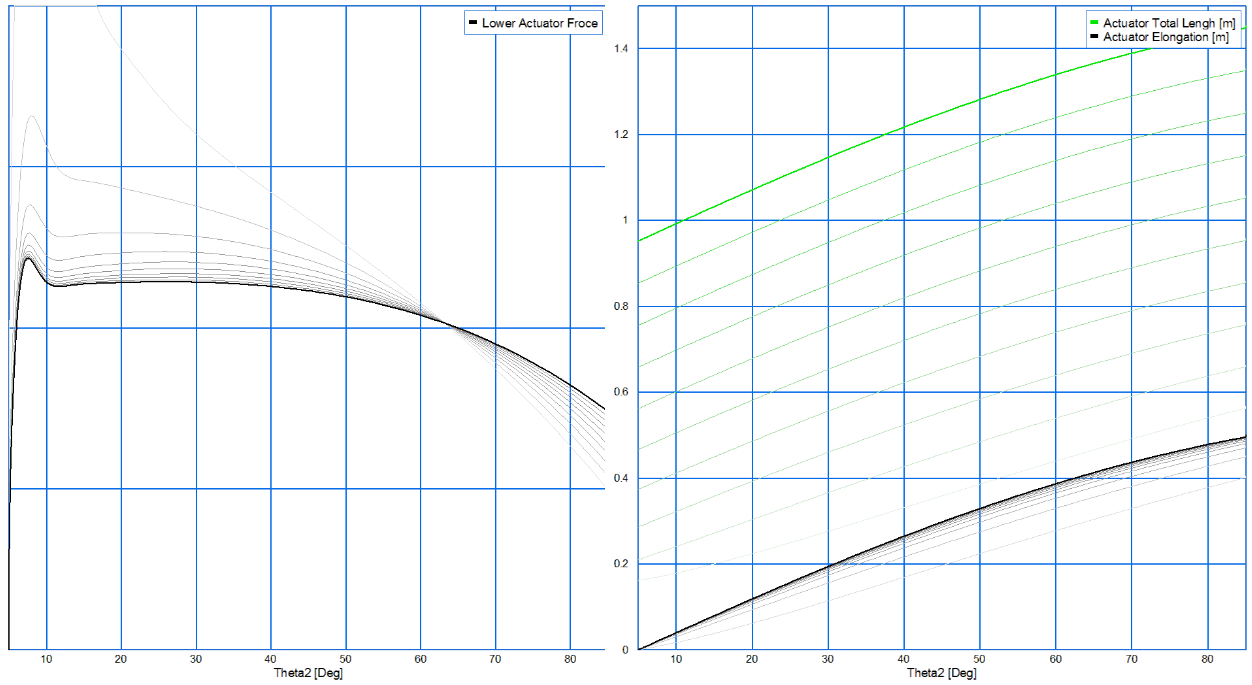
(a) Required Force from Actuator (b) Actuator Length and Elongation.

Figure 5.3: Variation of Parameter h

5.3.2 Placement of Lower Actuator at the Long Lower Arm

The connection point for the lower actuator on the lower arm is determined by the two parameters b, and u. Figure 5.4 and 5.5 show the required actuator force (figure 5.4a and 5.5a), and the length and elongation of the lower actuator (figure 5.4b and 5.5a) as a function of θ_2.

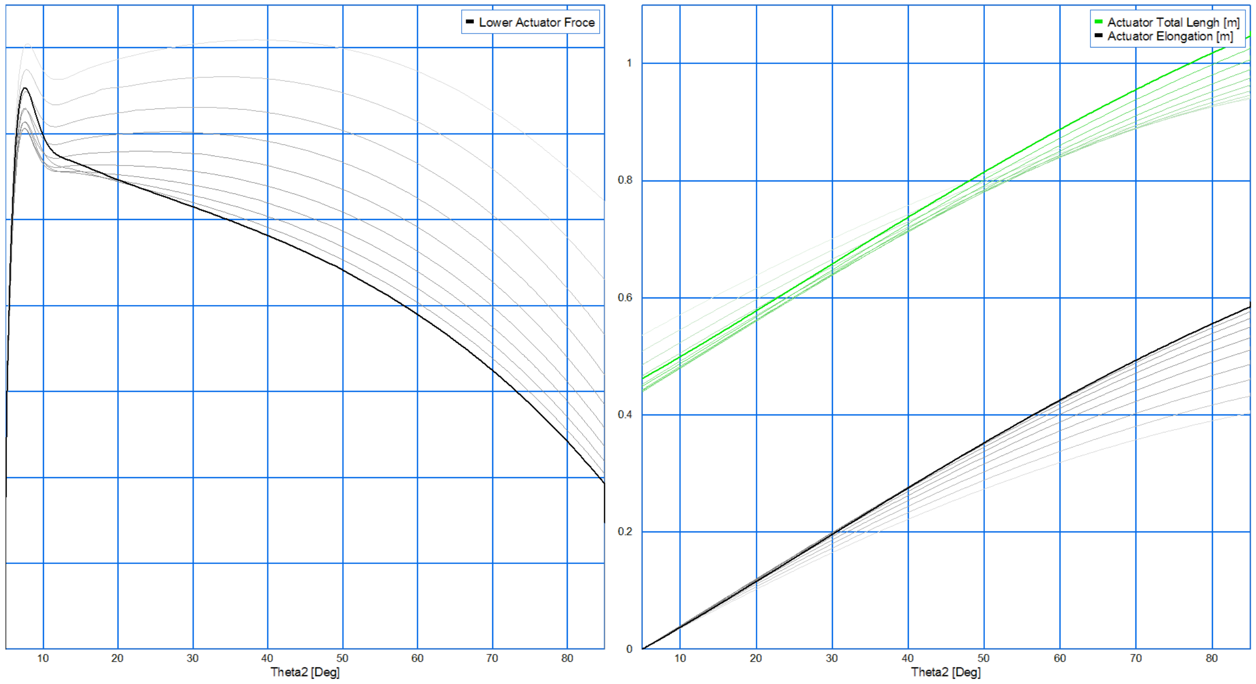
The parameter b is set to start at 0.1m and increase by 0.1m for each simulation step. The parameter u is set to start at 0.15m and increase by 0.05m for each simulation step. There are in total 10 simulation steps presented in the figures 5.4 and 5.5. The graphs with the lightest colour are for the smallest values of b and u.



(a) Required Force from Actuator

(b) Actuator Length and Elongation.

Figure 5.4: Variation of Parameter b



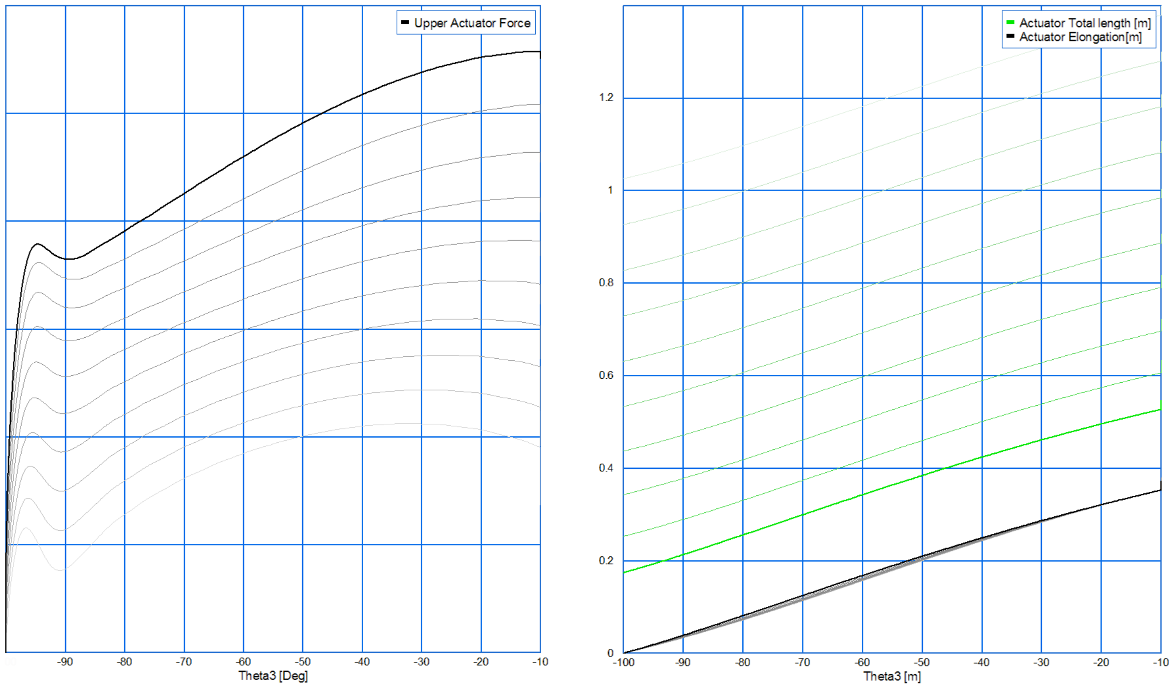
(a) Required Force from Actuator

(b) Actuator Length and Elongation.

Figure 5.5: Variation of Parameter u

5.3.3 Placement of Upper Actuator at the Long Lower Arm

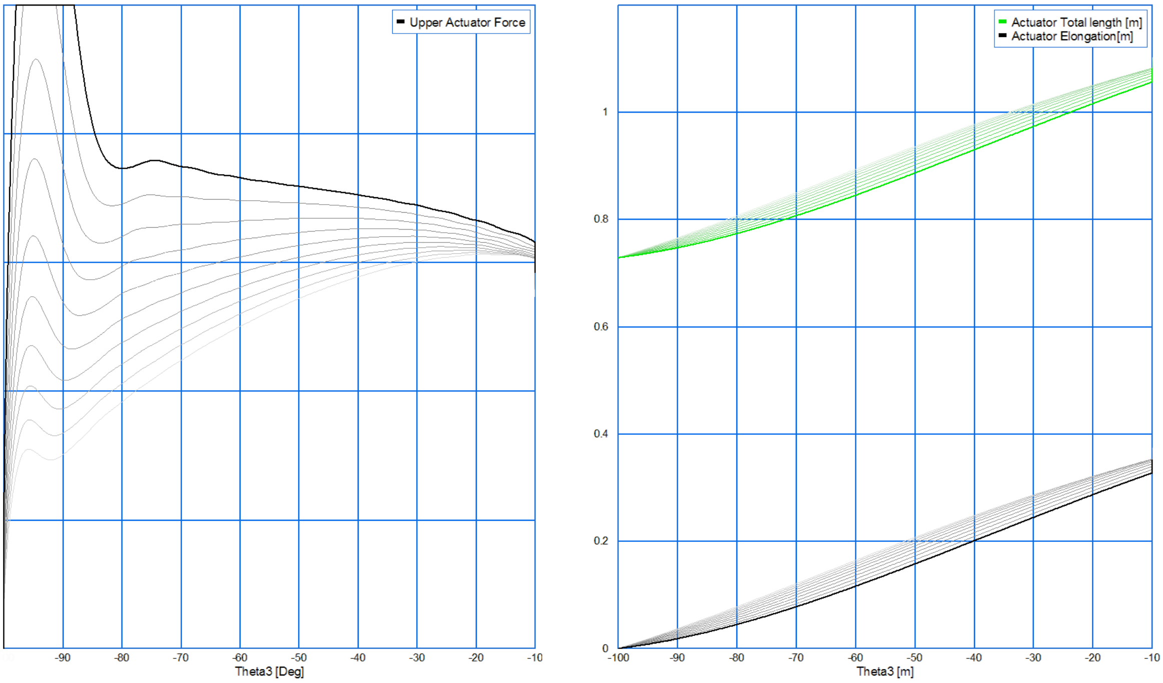
The connection point for the upper actuator at lower arm is determined by the two parameters w, and r. Figure 5.6 and 5.7 show the required actuator force (figure 5.6a and 5.7a), and the length and elongation of the lower actuator (figure 5.6b and 5.7a) as a function of θ_3. The parameter w is set to start at 1.7m and increase by 0.1m for each simulation step. The parameter r is set to start at 0.05m and increase by 0.05m for each simulation step. There are in total 10 simulation steps presented in the figures 5.6 and 5.7. The graphs with the lightest colour are for the smallest values of w and r.



(a) Required Force from Actuator

(b) Actuator Length and Elongation.

Figure 5.6: Variation of Parameter w



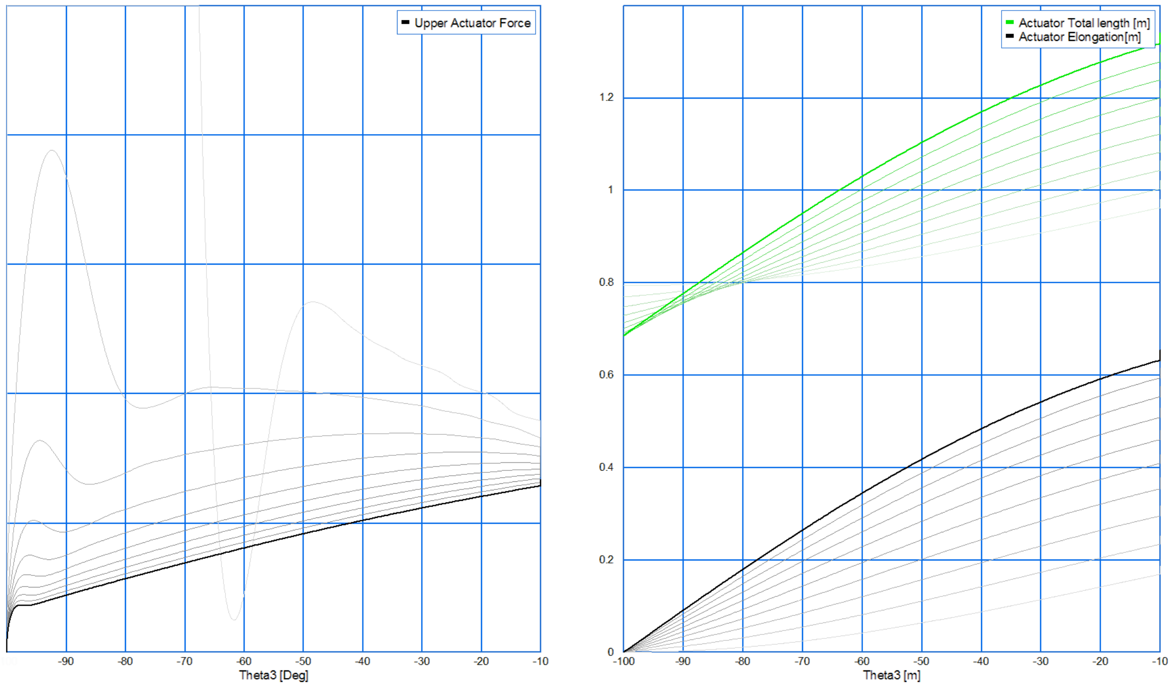
(a) Required Force from Actuator

(b) Actuator Length and Elongation.

Figure 5.7: Variation of Parameter r

5.3.4 Placement of Upper Actuator at the Upper Arm

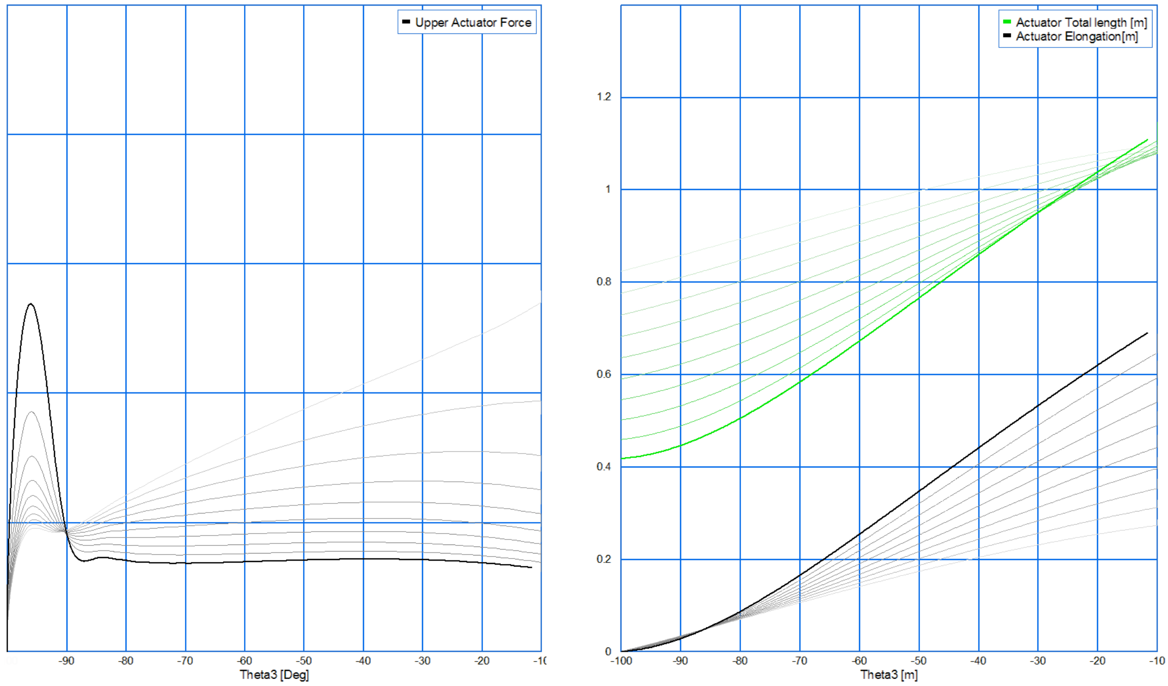
The connection point for the upper actuator at the upper arm is determined by the two parameters c, and s. Figure 5.8 and 5.9 show the required actuator force (figure 5.8a and 5.9a), and the length and elongation of the lower actuator (figure 5.8b and 5.9a) as a function of θ_3. The parameter c is set to start at 0.05m and increase by 0.05m for each simulation step. The parameter s is set to start at 0m and increase by 0.05m for each simulation step. There are in total 10 simulation steps presented in the figures 5.8 and 5.9. The graphs with the lightest colour are for the smallest values of b and u.



(a) Required Force from Actuator

(b) Actuator Length and Elongation.

Figure 5.8: Variation of Parameter c



(a) Required Force from Actuator

(b) Actuator Length and Elongation.

Figure 5.9: Variation of Parameter s

5.3.5 Lower and Upper Actuator Placement at the Short Lower Arm

As it is desired to reuse as many parts as possible when changing between the two crane arms, and also facilitate additional changes to the crane design in the future, the parameters concerning the placement of the actuator connection at the short lower arm are also worth analyzing.

The initial parameters presented in table 5.4 will not all be valid for the crane configuration with the short lower arm. An edited set of initial parameters valid for the short lower arm are therefore presented in table 5.5.

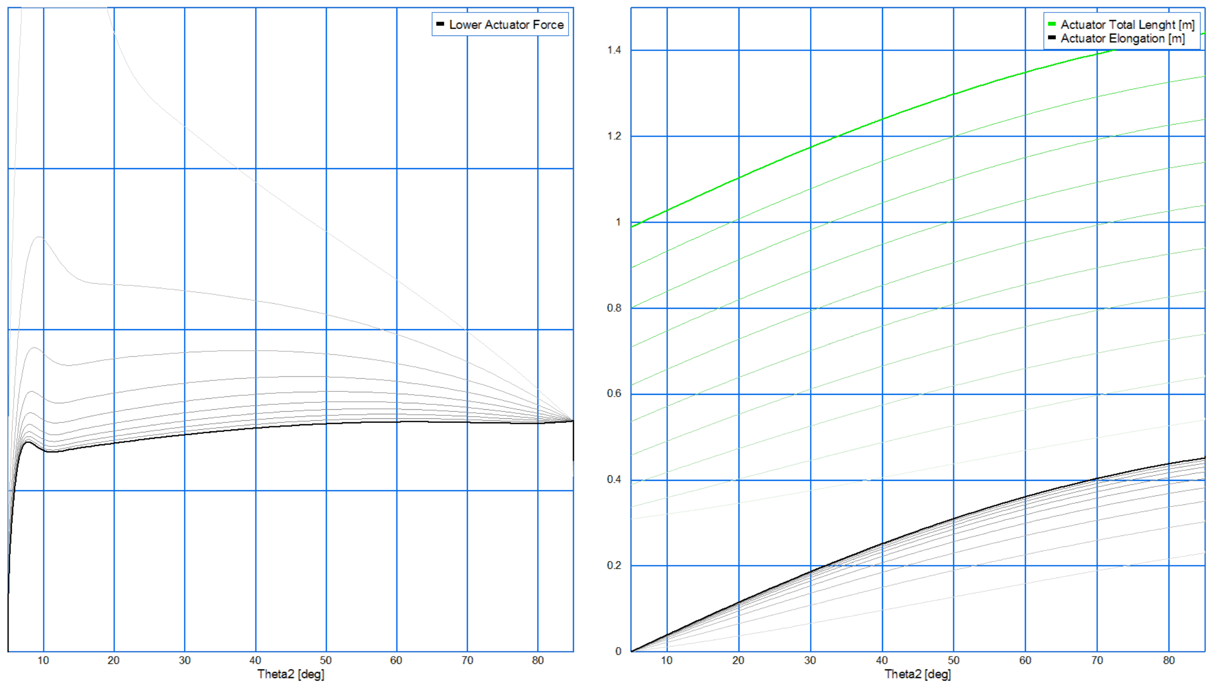
Table 5.5: Initial Parameters Short Arm

Parameter	Size	Unit
L_1	1300	mm
L2	1000	mm
h	450	mm
a	100	mm
b	500	mm
u	150	mm
r	150	mm
w	500	mm
c	200	mm
s	150	mm

Parameter	Size	Unit
CM1	50	kg
Ycm1	0	mm
Xcm1	0	mm
CM2	15	kg
k	500	mm
CM3	5	kg
n	400	mm
CM4	5	kg
p	400	mm
CM5	5	kg
o	400	mm

Note that $b = u$ and $u = r$ for this configuration of the crane. This is because the actuators are assumed to be too large to fit two actuator connection points for the short lower arm, ref Gyberg (2015) for estimated actuator lengths.

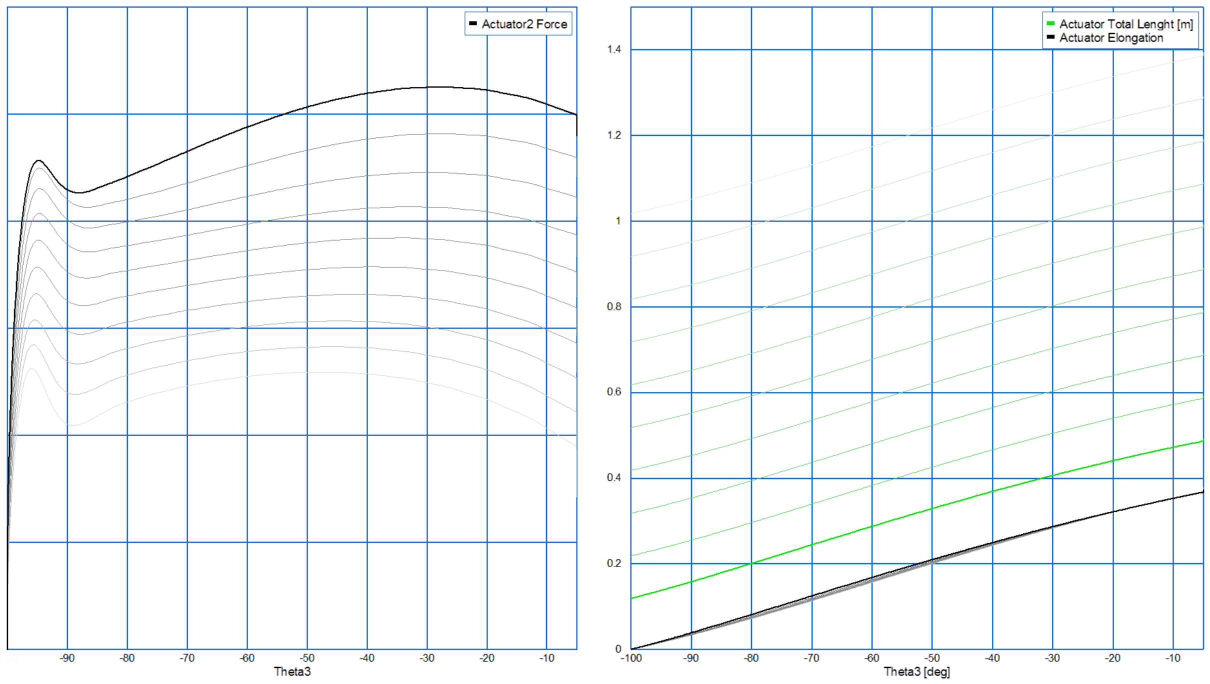
In the simulation results presented in figure 5.10 and 5.11 the parameter b is set to start at 0.1m and increase by 0.1m for each simulation step. In the simulation results presented in figure 5.12 and 5.13 the parameter u is set to start at 0.05m and increase by 0.05m for each simulation step. There are in total 10 simulation steps presented in the figures 5.10 to 5.13. The graphs with the lightest colour are for the smallest values of b and u.



(a) Required Force from Lower Actuator

(b) Lower Actuator Length and Elongation.

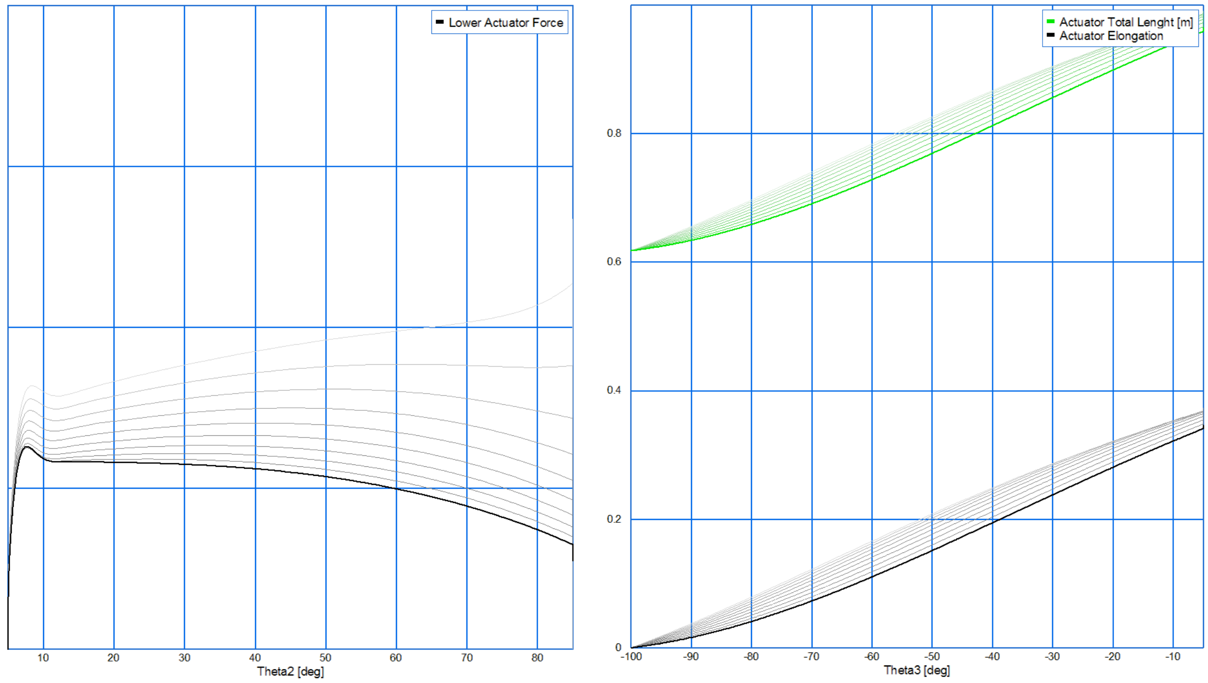
Figure 5.10: Lifting of Lower Arm, Variation of Parameter b.



(a) Required Force from Upper Actuator

(b) Upper Actuator Length and Elongation.

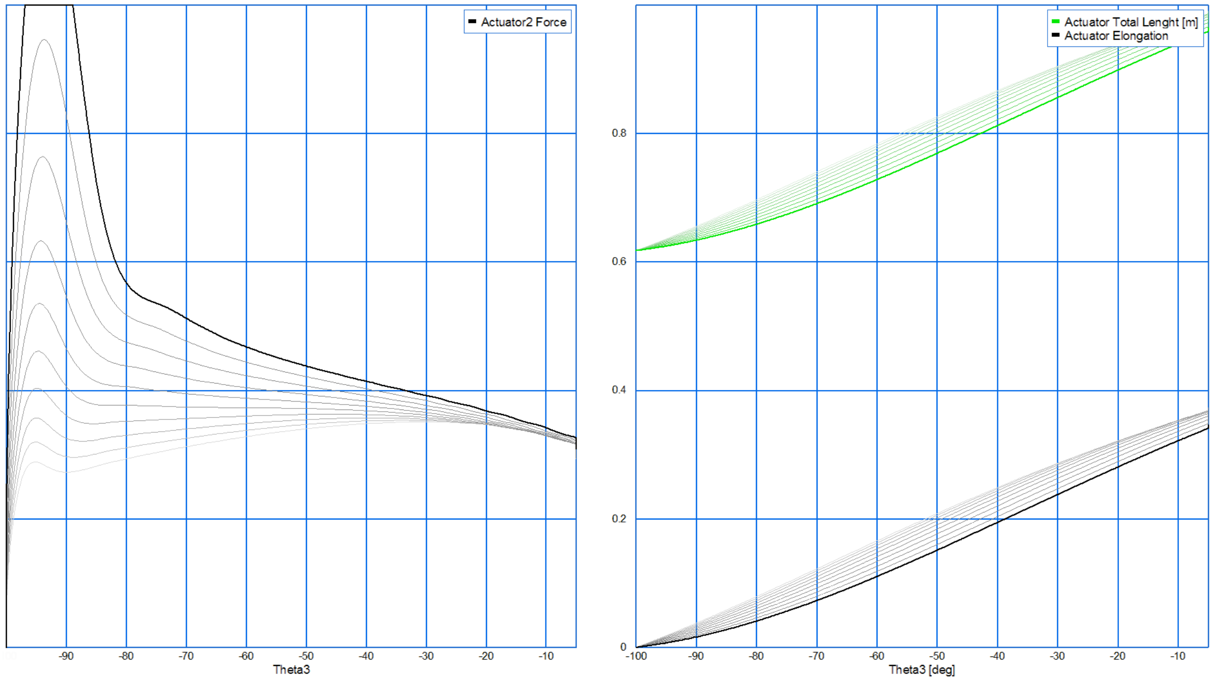
Figure 5.11: Lifting of Upper Arm, Variation of Parameter b.



(a) Required Force from Lower Actuator

(b) Lower Actuator Length and Elongation.

Figure 5.12: Lifting of Lower Arm Variation of Parameter u.



(a) Required Force from Upper Actuator

(b) Upper Actuator Length and Elongation.

Figure 5.13: Lifting of Upper Arm Variation of Parameter u.

5.4 Design of Crane Arms

The final design of the the crane arms is a result of an iteration process with basis in the analysis presented in section 5.3, crane concept presented in chapter 3, and the FEA presented in this section. By establishing the design of the crane arms, data for the weights and inertias can be calculated, the structural strength of the crane arms can be analyzed, and more accurate simulations can be performed to establish the required force from the actuators. The design, calculation of the weights and inertias, and the structural analysis are done with the use of the CAD software Autodesk Inventor.

The crane arm design will mostly be determined by selecting values for the parameters analyzed in section 5.3, all these parameters will have to be seen together as they all influence each other. Most of the parameters will primarily be optimized for the crane configuration with the long arm, but the short arm configuration is also taken into account to make sure that it is possible to fit the actuators with the small arm as well.

The parameter a is a parameter that does not directly influence the design of the crane arms, it is however a parameter which will have a considerable impact on both the required actuator elongation and force for a given angular movement range of the lower arm, see figure 5.2. The parameter a is therefore worth considering when designing the crane arms. A small value of a will significantly increase the movement range at a given actuator elongation. The parameter a will also affect the required lifting force from the lower actuator, where a smaller value of a will yield a higher required force. Parameter a is set at 100mm as a compromise when considering both the required force and the lower crane arm movement range.

5.4.1 Length of the Crane Arms

The framework presented in section 3.1 sets clear demands with regards to the crane arm lengths for the configuration with the long lower arm. *The crane should be able to simulate a real crane in a scale of 1:25.* The scaling conditions, table 3.1, specify that the crane with the the long arm is to be able to do a lift at 4m height and 1m length. The requirements for the length of the short lower arm is that it should be of a size that is practical to move around by hand. These requirements will affect both the crane arm lengths and and the height of the lower arm connection point at the base (e.g. parameters L_1, L_2 and h).

The first parameter to consider is h. This is because it is found in section 5.3 to have a significant impact on both the required force from the lower actuator, and the required actuator size and elongation. The parameter h is set at 495mm. This is a value where the force reduction for higher values of h is small, and the practicality of not increasing the parameter h further (considering the size of the crane base) is thought to be of higher importance than the small reduction in the required force achieved by increasing the parameter h further.

The length of the short lower arm in Gyberg (2015) was set at 1050mm. This length is in retrospect considered a bit on the short side and can be increased without the lower arm being impractical to move manually. Another consideration is that the size of the actuators probably will increase compared to Gyberg (2015) due to the increased design load, and the increased weight of the arms due to the updated conceptual design, ref chapter 3. Larger actuators will require a longer arm as it will be difficult to fit actuators any larger than presented in Gyberg (2015) to a lower crane arm with a length of 1050mm.

As the actuator size is a concern with the shorter lower crane arm, an early estimate of the actuator size is due. The size of the actuators determined in Gyberg (2015) are summarized in table 5.6.

As it is assumed that the length of the actuators will increase, the estimate of the length of the actuators fully retracted has therefore been increased to 700mm for both the upper and lower actuator. The elongation of the actuators is still assumed to be approximately 240mm.

The length of the short lower arm is then set to 1251mm to accommodate for the longer actuators.

It is decided that the upper arm should be smaller than the lower arm as this is most common on knuckle boom cranes. The upper limit for the length of the upper arm is therefore set to be 1000mm, as the length of the short lower arm is 1251mm. This limit will affect both the crane configuration with the short and long lower arm.

The required lifting height and length of the crane configuration with the long lower arm is the most important criteria to consider when the length of the long lower arm is set. It is natural that the angle of the lower arm (θ_2) is quite close to 90° when performing the 4m high lift. This is because the required length of the lift is only 1m, see table 3.1. It is assumed that it will be possible to achieve an angle of at least $\theta_2 = 80°$. With the length of the lower arm set to 2954mm and the length of upper arm set to 960mm, the lifting height and length will be 4067mm and 1228mm respectively, with angle $\theta_2 = 80°$ and $\theta_3 = -40°$. See figure 5.14 for reference.

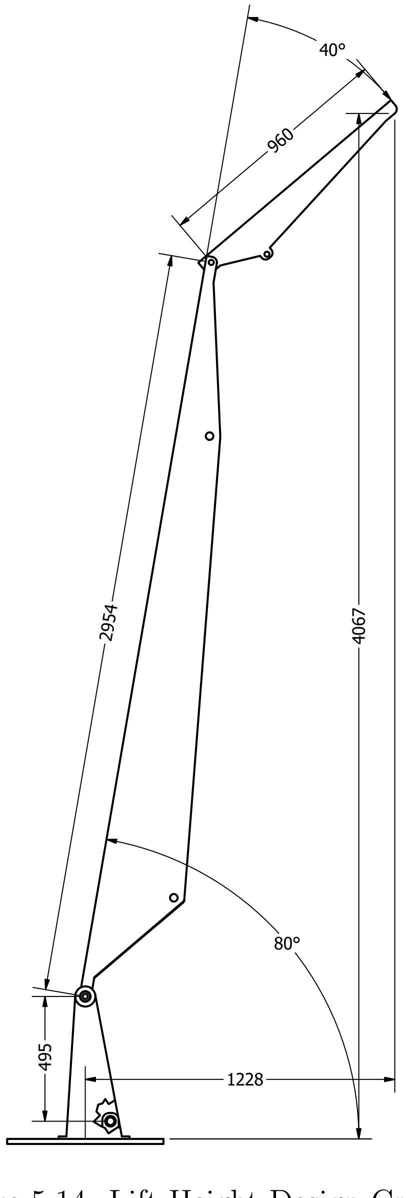


Figure 5.14: Lift Height Design Criteria

Table 5.6: Actuator Size from Gyberg (2015)

Parameter	Size	Unit
Lower Actuator Length fully retracted	588.5	mm
Lower Actuator Length fully extended	828.5	mm
Upper Actuator Length fully retracted	573.5	mm
Upper Actuator Length fully retracted	813.5	mm

5.4.2 Upper Arm

The upper arm is common for both the crane configuration with the long and the short lower arm. It is therefore important to consider the design of the upper arm and the actuator geometry of the upper actuator with both lower arm configurations.

A simplified version of the upper arm is used in the structural analysis, see figure 5.15. The arm is simplified by removing the rollers guiding the wire and replacing the bearings with solid steel representations, this is done to simplify the analysis and reduce the computational time required to perform the analysis. The arm is placed horizontally and loaded with a load of 500N at the end of the arm. A gravitational load is also applied. The movement of the upper arm is constrained in radial direction of the bearings at both the actuator connection point and the upper arm connection point with the lower arm.

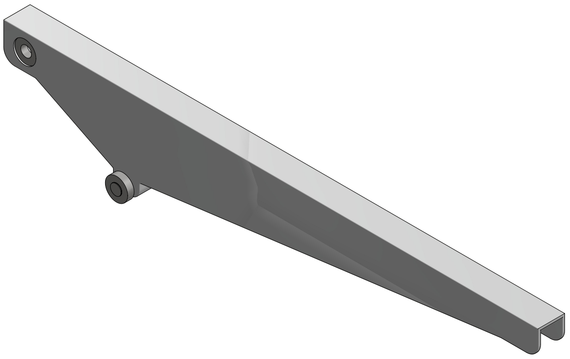


Figure 5.15: Simplified Upper Arm for Structural Analysis

The most important part of the upper arm design regarding optimization of the required actuator force and elongation is to determine values for the parameters s and c. The parameter s is the parameter that have the largest influence on the size of the upper actuator when a movement range of θ_3 close to $-100°$ is desired, ref figure 5.9b. It is also found (figure 5.9a) that the parameter s does not affect the required force much if chosen above 100mm. The parameter s is therefore set at 118mm. This value of s is considered a value that will ensure a versatile design of the upper arm, with focus at enabling a large movement range of the upper arm. It is also unnecessary to choose a higher value of s when considering the strength of the upper arm.

The parameter c has a great affect on the required force from the upper actuator, see figure 5.8a. A lower limit of c is therefore set at 150mm. The parameter c should however not be set too high as the movement range of the upper crane arm will be limited, see figure 5.8b. The parameter c is set at 190mm. This value is also considered to ensure a versatile design, by not requiring too much force from the actuator and also ensuring a large movement range of the upper arm.

The upper crane arm is built up, as explained in section 3.2.2, around a square extruded aluminium profile. The profile will be reinforced with a solid aluminium section at both the connection point with the lower arm and the upper actuator and stiffening plates is placed underneath the square profile. A plate thickness of 5mm of both the square aluminium profile and the the reinforcements underneath the upper arm is found to be more than adequate with regard to strength. The small rod that connects the actuator to the upper arm is decided to be made of a 20mm steel rod to ensure small deflections and stresses where the actuator connects to the upper arm.

The final structural analysis is presented in figure 5.16 and 5.17.

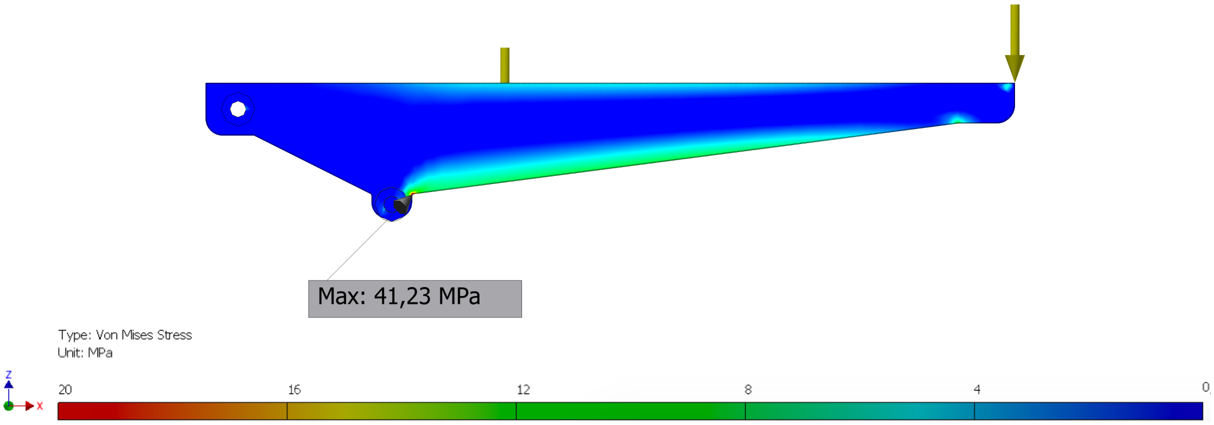


Figure 5.16: Upper Arm Von Mises Stress

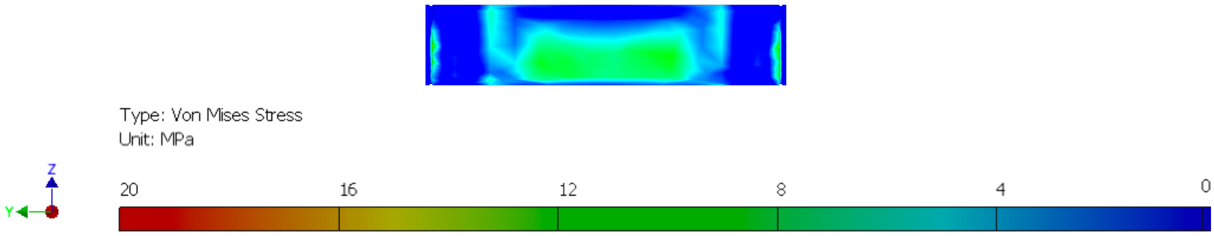


Figure 5.17: Actuator Pin Von Mises Stress

The structural analysis prove that the strength of the upper arm is more than adequate. The stress will be highest underneath of the upper arm where it is mostly varying around 10 MPa, giving the upper arm a safety factor of 13 with regard to yield strength, ref. table 5.1 for the material properties. The highest stress is found to be 41 MPa which is also well below the yield strength of the material. It should also be noted that this is a small stress concentration in the connection point between the reinforcement of the actuator connection and the reinforcement at the underside of the square section of the upper arm.

In order to verify the structural analysis a convergence analysis and an evaluation of the reaction forces is done.

For the convergence analysis the mash of the part is refined as explained in section 2.4 to ensure that the simulation converges, see figure 5.18 for plot of the result from the convergence analysis.

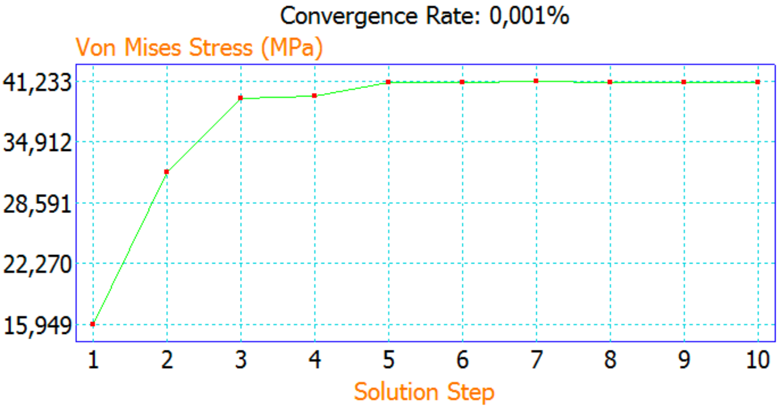


Figure 5.18: Upper Arm FEA Convergence

Figure 5.18 makes it clear that further refinement of the mash will not yield any higher stress, thus the analysis converges.

The reaction forces are presented in table 5.7.

Table 5.7: Upper Arm Reaction Forces

Parameter	Value	Unit
Upper-Lower Arm Connection X-direction	-2.40	kN
Upper-Lower Arm Connection Z-direction	-0.57	kN
Actuator Connection X-direction	2.40	kN
Actuator Connection Z-direction	1.12	kN

The weight of the upper arm equals a gravitational load of approximately 0.05 kN, ref table 5.9. As presented in equation 5.13, one can see that a control has been made to ensure that the reaction forces in all directions, and that torque around the connection point between the lower and upper arm equals zero.

$$
\begin{aligned}
\sum F_X =& 2.40kN - 2.40kN = 0kN \\
\sum F_Z =& 1.12kN - 0.50kN - 0.05kN - 0.57kN = 0kN \\
\sum M_{ArmConnection} =& -0.50kN * 0.96m - 0.05kN * 0.42m + 2.40kN * 0.12m \\
& + 1.12kN * 0.19m = 2 * 10^{-4}kNm \approx 0kNm
\end{aligned}
\tag{5.1}
$$

This analysis concludes that the result of the structural analysis is probable, and that the structure of the upper arm satisfies the requirement that the structure should be able to withstand a much greater load than the design load.

The connection between the upper arm and actuator, and the connection between the upper arm and the lower arm is established with the use of roller bearings. The selected bearings are sealed making it unnecessary with external seals around the bearings. Both connections uses 20mm steel rods, so the same bearing is selected for both connections. The bearing is presented in table 5.8, see also SKF6004-2Z (2016) for reference.

Table 5.8: Roller Bearing SKF6004-2Z

Parameter	Value	Unit
Outer Diameter	42	mm
Inner Diameter	20	mm
Width	12	mm
Max Static Load	5	kN
Max Dynamic Load	9.9	kN

The resulting force from the X- and Z-component of the reaction force at the actuator connection is approximately 2.6 kN. This will yield a static load of each bearing at the actuator connection of 1.3 kN which is well below the rated static load of the bearing. This is also the case for the connection between the upper and lower arm where the load is even smaller, ref table 5.7.

The wire guidance rollers and corresponding brackets and rods are designed without any analysis. This is based on the analysis of the main part of the upper arm where the stress is found to be well below the yield stress for all parts used. The rollers are proposed to be made by machining an extruded or solid aluminium cylinder, but can also be bought as a finished product. The rollers is proposed to be supported by 10mm steel rods, and connected to the upper crane arm with 5mm aluminium plates. Sealed ball bearings similar to the ones used for the actuator and lower arm connection points is proposed to be used for the rollers as well.

The final design of the upper arm is summarized in figure 5.19 and table 5.9. All dimensions in figure 5.19 are in mm.

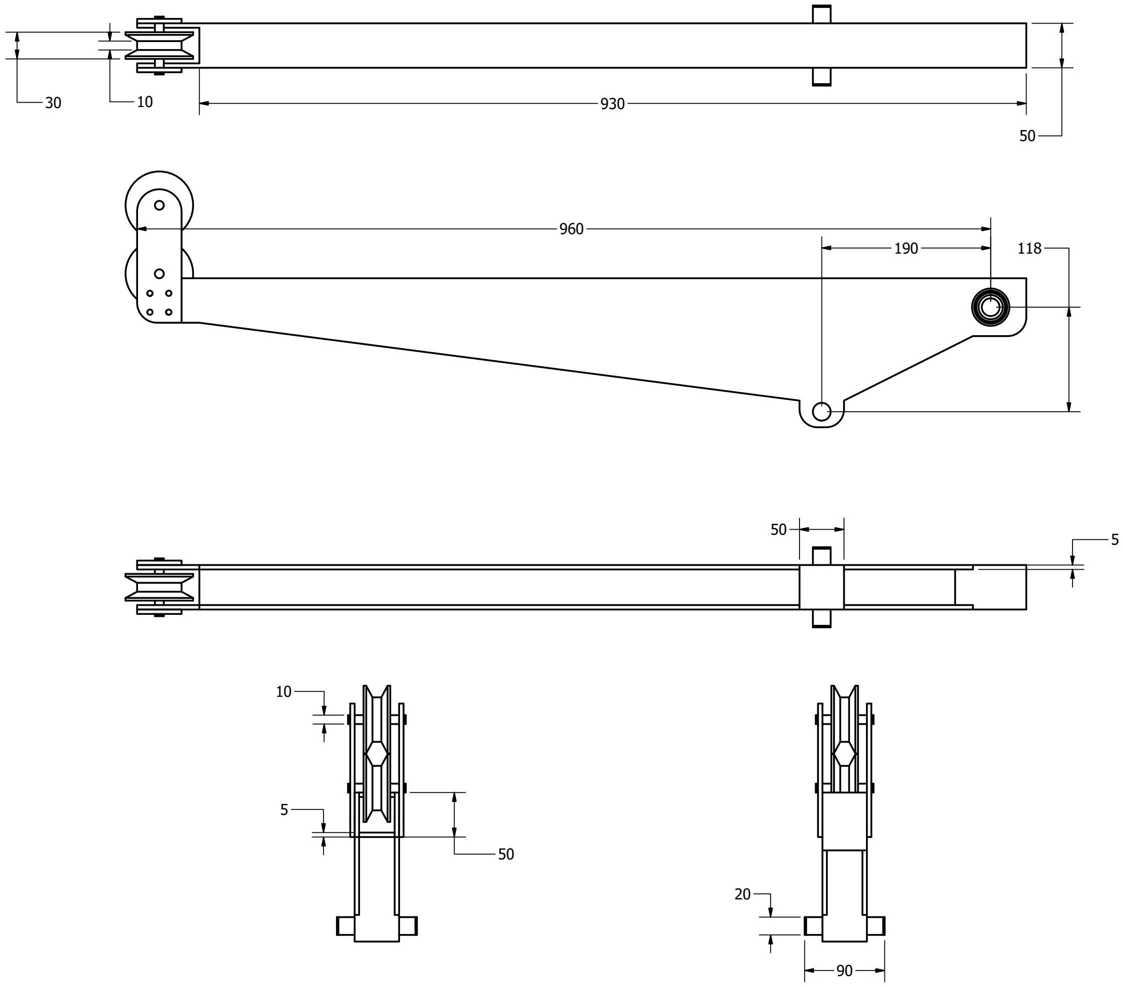


Figure 5.19: Upper Arm Final Design, all dimensions in are in mm.

Table 5.9: Upper Arm Properties

Parameter	Value	Unit
Total Mass, CM_3	5.5	kg
Centre of gravity, n	420	mm
Rotational inertia around X-axis, I_{xx}	1.6	kgm^2
Rotational inertia around Z-axis, I_{zz}	1.5	kgm^2

5.4.3 Long Lower Arm

The design of the long lower arm is, as with the upper arm, a result of an iteration process where the parameters concerning the actuator geometry and the structural strength of the arm is evaluated.

The strength of the long lower arm is evaluated by the same principals and methods as with the upper arm. A simplified version of the long lower arm is used in the structural analysis, see figure 5.20.

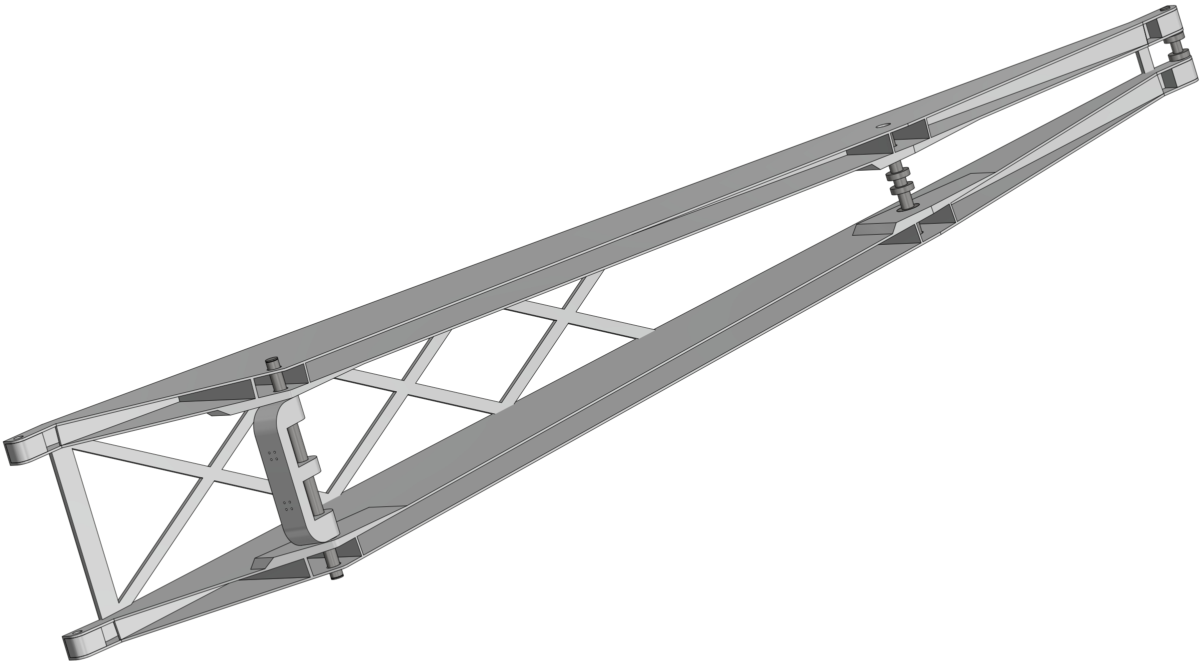


Figure 5.20: Simplified Long Lower Arm for Structural Analysis

The arm is simplified by removing the rollers guiding the wire and replacing the bearings with solid steel representations. The simplifications is done to simplify the analysis and reduce the computational time required to perform the analysis. The top bracket of the lower actuators is also included to get an accurate representation of the loading of the lower actuator connection rod at the lower arm. The arm is placed at an angle of $\theta_2 = 38°$ and loaded with the reaction forces obtained from the analysis of the upper arm, see also figure 5.1 for an illustration of the configuration of the crane during the structure analysis. A gravitational load is also applied. The movement of the lower arm is constrained by constraining the lower actuators top bracket in direction normal to the part of the bracket that connects to the actuators, and in radial direction of the bearings connecting the lower arm to the base.

The most important part of the lower arm design with regard to optimizing the required force and elongation of both the upper and lower actuator is to determine values for the parameters b, u, w and r.

The parameters w is the one that will have the largest influence on the required length of the upper actuator, see figure 5.6b. As the estimated fully extended actuator length is 940mm and the upper crane arm has to achieve at least an angle of $\theta_3 = -40°$ the value of w should not be chosen any smaller than 2100mm. The required force from the upper actuator is increasing with higher values of w, see figure 5.6a. A value of $w = 2150$mm therefore is seen as a fair choice.

The required force from the upper actuator in the lower angular movement range of the upper arm will see a significant increase if the parameter r is set too high, see figure 5.7a. The movement range and required length of the upper actuator is affected less by the choice of the parameter r, see figure 5.7b. The parameter r is therefore set at a fairly low value, $r = 100$mm.

The parameter b is the parameter of the lower actuator geometry that has the largest impact on the minimum length of the lower actuator, see figure 5.4b. The parameter b is set to be 445mm for the long lower arm. This value of b will, with all the other parameters at the initial values, give a min value of $\theta_2 \approx 20°$ with the fully retracted length of the actuator is 700mm, ref figure 5.4b. This is considered a more than small enough value of the angle θ_2 for the long lower crane arm. The parameter $b = 445mm$ will also mean that the lower actuator will not suffer from the high increase of the required lifting force in the lower angular range of the lower arm that a smaller value of b will yield, see figure 5.4a.

The value of u should with reference to the analysis in section 5.3.2 be set fairly high as it will give a considerable reduction on the required force from the lower actuator, see figure 5.5a. The downside by setting a high value for u is that it will result in a considerable increase in the structure at the lower arm. The parameter u is set to be 302.5mm, this is a value that is considered to be at the the best compromise between increase in required force and in the in required structure in the lower arm.

The long lower crane arm is built up similarly as the upper arm as explained in section 3.2.2. This includes the square extruded aluminium profile which is reinforced with a solid aluminium section at both the connection point with the upper arm and the base. The square profile is also reinforced on the underside and at the actuator connection points with plate stiffeners. A plate thickness of 5mm of both the square aluminium profile and the the reinforcements underneath the upper arm is found to be more than adequate with regard to strength. The rod that connects both the upper and lower actuator to the long lower arm is made of a 25mm steel rod, this is to ensure small deflections and stresses at this critical point. The rod that connects the upper and lower arm is made of a 20mm steel rod for the same reasons.

The final structural analysis is presented in figure 5.21, 5.22, 5.23 and 5.24.

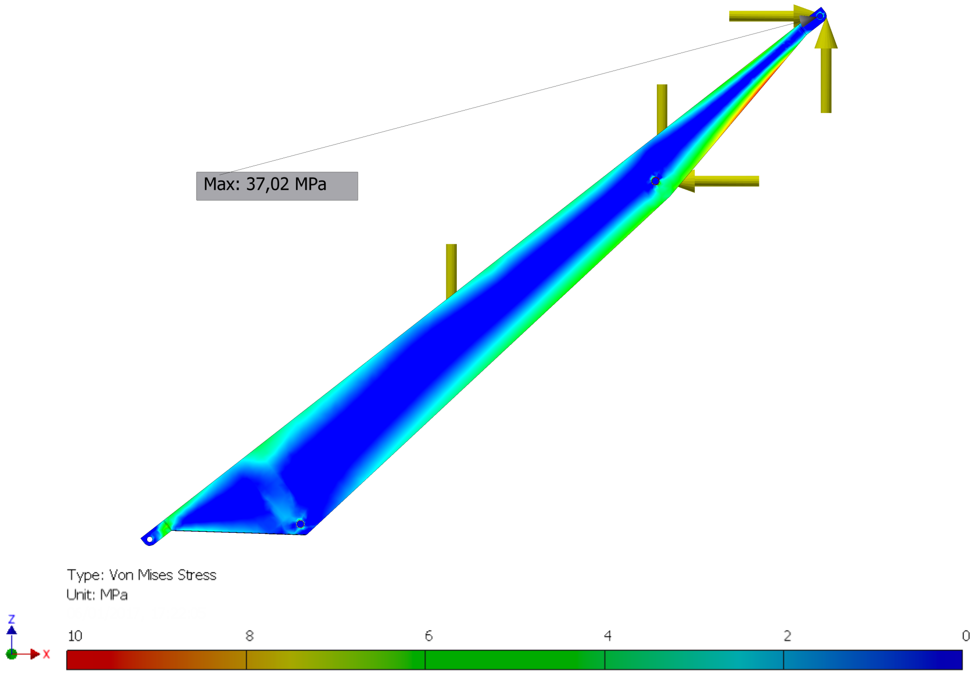


Figure 5.21: Lower Arm Von Mises Stress

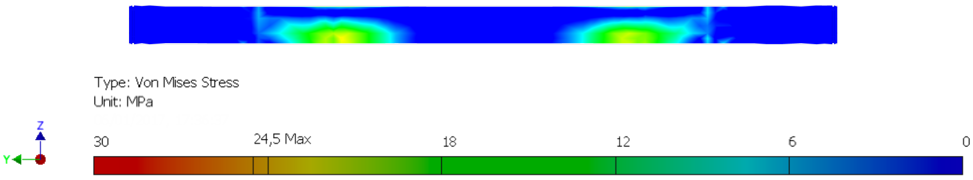


Figure 5.22: Lower Actuator Pin Von Mises Stress

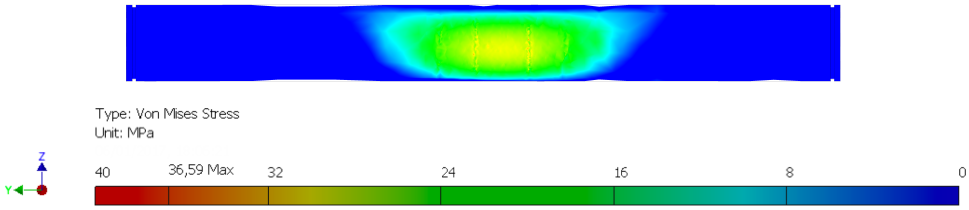


Figure 5.23: Upper Actuator Pin Von Mises Stress

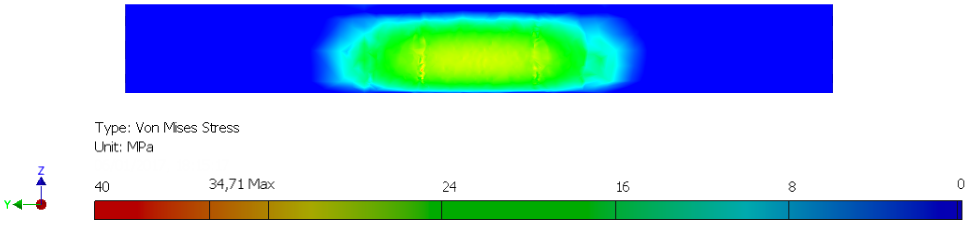


Figure 5.24: Upper-Lower Arm Conecting Pin Von Mises Stress

The structural analysis prove that the strength of the lower arm is more than adequate. The stress will be highest in the underside of the upper part of the lower arm where it is mostly varying around 10 MPa, giving the upper arm a safety factor of 13 with regard to yield strength, ref table 5.1 for the material properties. The highest stress is found to be 37 MPa which is also well below the yield strength of the material. It should also be noted that this is a small stress concentration where the rod that connects the upper and lower arm is connected to the lower arm.

In order to verify the structural analysis a convergence analysis and an evaluation of the reaction forces is done.

For the convergence analysis the mash of the part is refined as explained in section 2.4 to ensure that the simulation converges, see figure 5.25 for plot of the result from the convergence analysis.

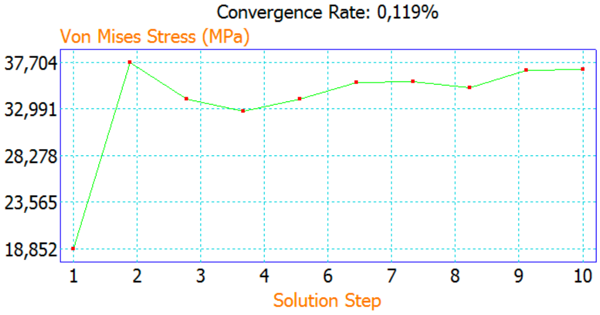


Figure 5.25: Lower Arm FEA Convergence

From figure 5.25 it is clear that the convergence of the analysis of the lower arm is not as unambiguously as for the upper arm, but is is clear that the convergence is fairly good, and the mash is therefore considered to be ok.

The reaction forces are presented in table 5.10.

Table 5.10: Lower Arm Reaction Forces

Parameter	Size	Unit
Lower Arm - Base Connection X-direction	-4.20	kN
Lower Arm - Base Connection Z-direction	-4.30	kN
Lower Actuator Connection X-direction	4.20	kN
Lower Actuator Connection Z-direction	5.05	kN

The weight of the lower arm is 52kg this will equal a gravitational load of approximately 0.40 kN when the angle of the lower arm is considered, ref. table 5.12. As presented in equation 5.2, one can see that a control has been made to ensure that the reaction forces in all directions, and that torque around the connection point between the lower arm and the base equals zero.

$$\begin{aligned}\sum F_X =&4.20kN - 4.20kN + 2.40kN - 2.4kN = 0kN\\ \sum F_Z =&5.05kN - 4.10kN - 0.40kN - 1.12kN + 0.57kN = 0kN\\ \sum M_{ArmConnection} =&5.05kN * 0.52m - 4.20kN * 0.05m + 0.57kN * 2.34m\\ &+ 2.40kN * 1.23m - 2.40kN * 1.80m - 1.12kN * 1.77\\ &- 0.4kN * \cos(38°) * 1.29m = -7.21 * 10^{-3}kNm \approx 0kNm\end{aligned} \tag{5.2}$$

The verification of the FEA analysis concludes that the result of the structural analysis is probable, and at the structure of the long lower arm satisfies the requirements that the structure should be able to withstand a much greater load than the design load.

The connection between the long lower arm and the upper and lower actuator is established with the use of roller bearings. The bearings selected are sealed so that it will not be necessary with seals around the bearings. Both connections uses 25mm steel rods, so the same bearing is selected for both connections. The bearing is presented in table 5.11, see also SKF6005-2Z (2016) for reference.

Table 5.11: Roller Bearing SKF6005-2Z

Parameter	Value	Unit
Outer Diameter	47	mm
Inner Diameter	25	mm
Width	12	mm
Max Static Load	6.5	kN
Max Dynamic Load	11.9	kN

The resulting force from the X- and Y-component of the reaction force at the lower actuator connection is approximately 6.6 kN. This will yield a static load of each bearing at the actuator connection of 3.3 kN which is well below the rated static load of the bearing. This is also the case for the upper actuator connection where the load is even smaller, ref table 5.7.

The wire guidance rollers and corresponding brackets and rods are designed the same way as on the upper arm.

The design of the long lower arm is summarized in figure 5.26 and table 5.12. All dimensions are in mm in figure 5.26

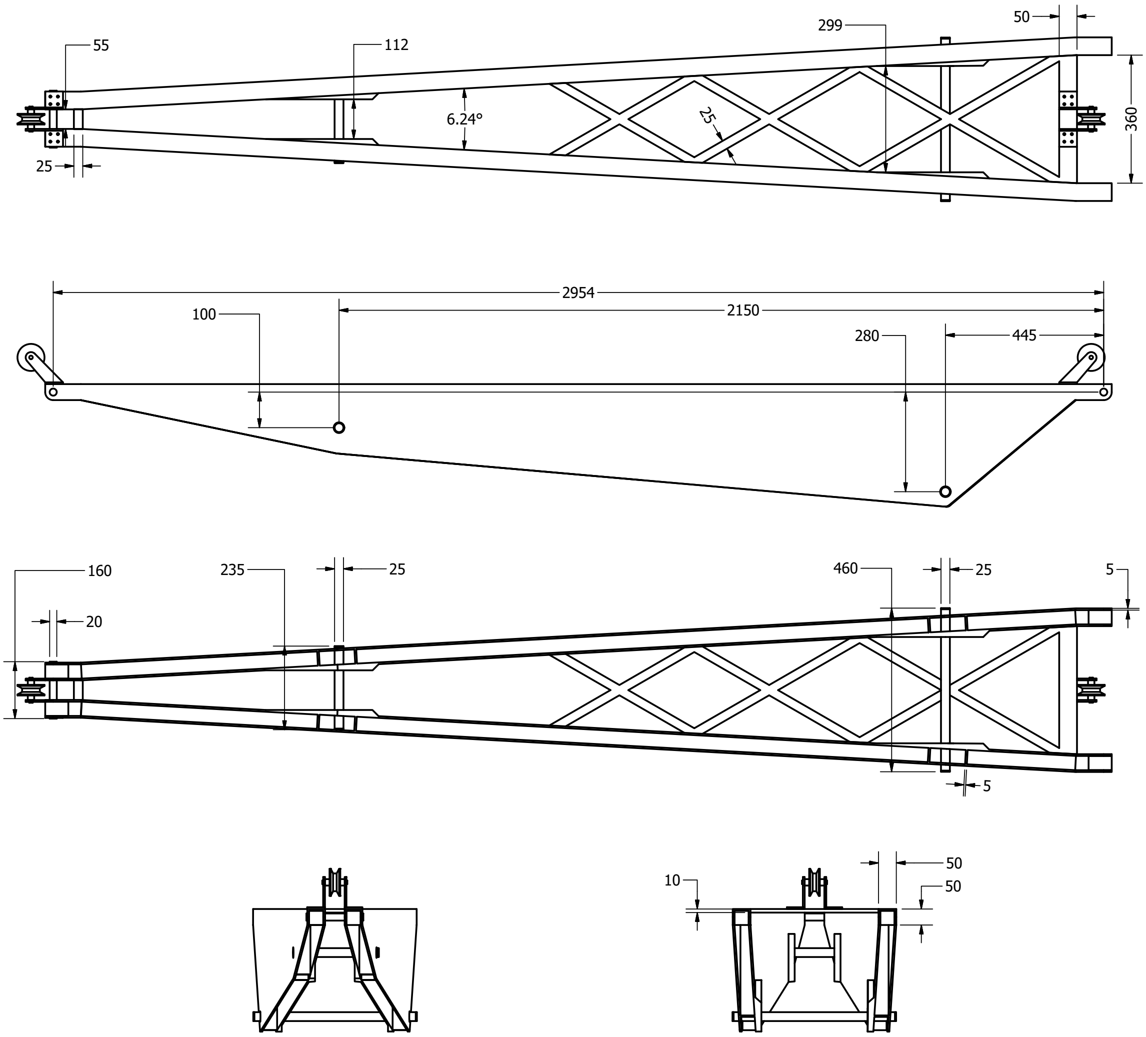


Figure 5.26: Long Lower Arm Final Design, all dimensions are mm

Table 5.12: Long Lower Arm Properties

Parameter	Size	Unit
Total Mass, CM_2	52.0	kg
Centre of gravity, k	1286	mm
Rotational inertia around X-axis, I_{xx}	120.1	kgm^2
Rotational inertia around Z-axis, I_{zz}	120.2	kgm^2

5.4.4 Short Lower Arm

The design of the short lower arm is mostly based on the optimisation of the parameters b and u. The structure of the short lower arm is built up by the same principals, plate thicknesses and materials as for the upper arm and the long lower arm. The structural strength of the short lower arm is therefore considered adequate without performing a separate structural analysis for the short lower arm. This conclusion is reached with basis in the analysis of the structure of the long lower arm and the upper arm

The actuator geometry provided by the short lower arm is provided by the parameters b, u. These two parameters will influence the actuator geometry for both the upper and lower actuator when the crane is configured with the short lower arm.

The parameters b is the one that will have the largest influence on the required actuator length, see figure 5.10b and 5.11b. As the estimated fully retracted actuator length is 700mm and it is desired to be able to lower the lower arm below $\theta_2 = 40°$ a value of $b < 500$mm is required, ref figure 5.10b. This is also the case for the upper arm if it is to reach an angle of $\theta_3 < -80°$, ref figure 5.11b. A value of b less then 300mm is not a good solution as the increase in the required force from the lower actuator will be high, ref figure 5.10a. The value of b is set at 461mm as it is found to be the best compromise.

The parameter u will have the greatest influence on the required force from the upper actuator in the lower angular range of the upper arm, ref figure 5.13a. The required force from the lower actuator will also be affected slightly in the upper angular movement range of the lower arm, see figure 5.12a. A value of $u = 140$mm is seen as the best compromise.

The design of the short lower arm is summarized in figure 5.27 and table 5.13. All dimensions are in mm in figure 5.27

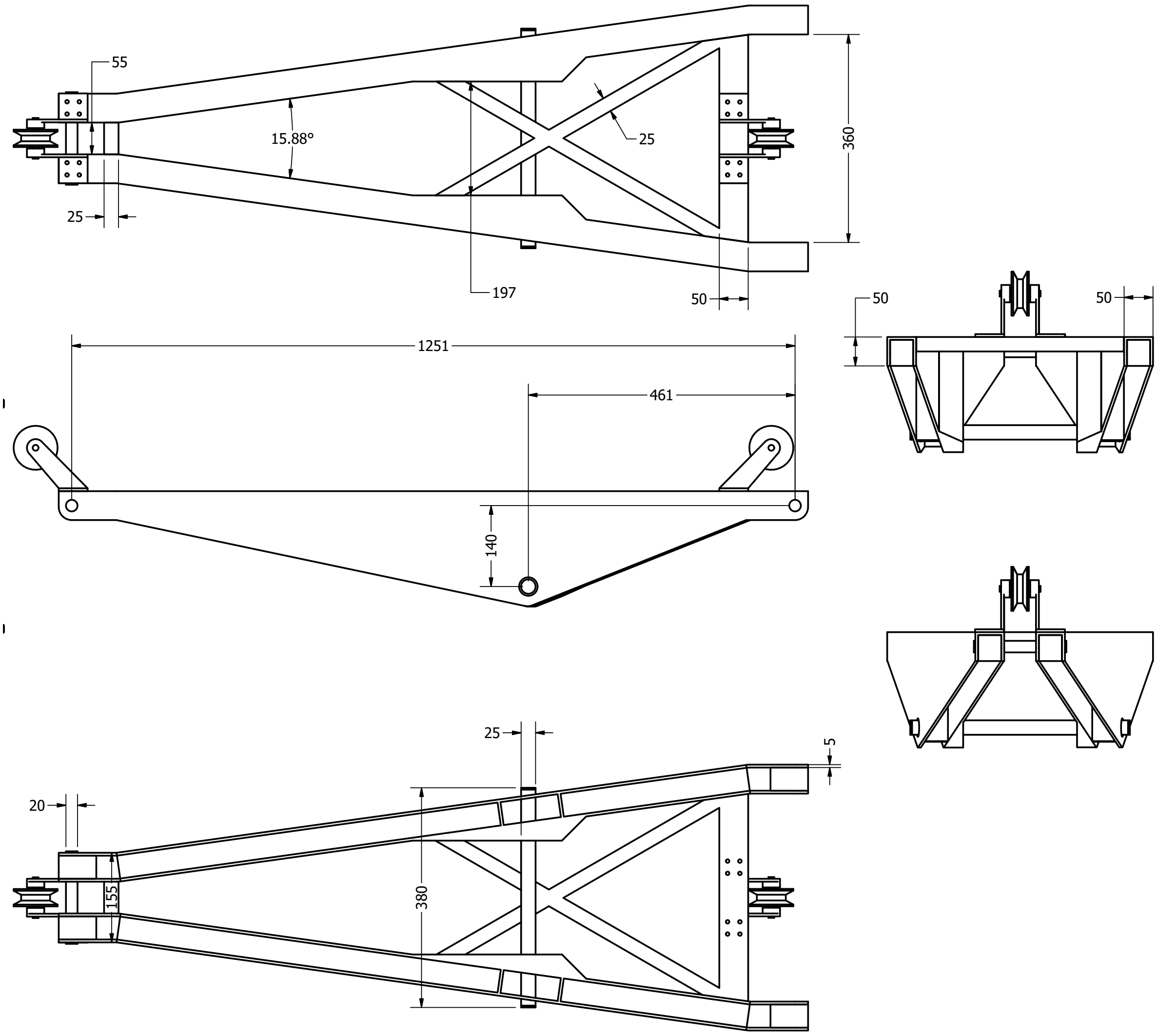


Figure 5.27: Short Lower Arm Final Design, all dimensions are in mm.

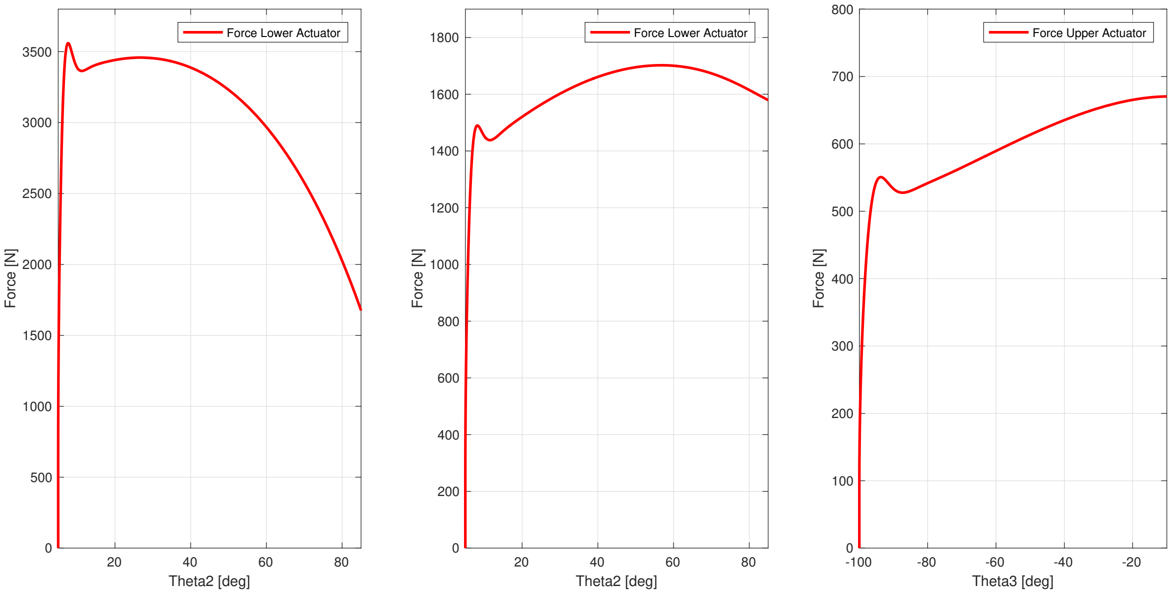
Table 5.13: Short Lower Arm Properties

Parameter	Size	Unit
Total Mass, CM_2	23.3	kg
Centre of gravity, k	533	mm
Rotational inertia, I_{xx}	9.2	kgm^2
Rotational inertia, I_{zz}	9.4	kgm^2

5.5 Actuators

With the design of the crane arms completed and the actuator geometry established the simulation model is used to find how much force the actuators are required to deliver.

Figure 5.28 shows the required lifting force the from the lower and upper actuator as a function of the angle of the relevant crane arm. The tests for the lower actuator is done by lifting the lower arm from 5° to 85°, with $\theta_3 = -45°$. The test for the upper actuator is done by holding the lower arm at $\theta_2 = 45°$. The parameters in the simulation model is updated with the values presented in section 5.4.



(a) Lower Actuator, Long Arm (b) Lower Actuator, Short Arm (c) Upper Actuator

Figure 5.28: Required Force from Actuators

The range $\theta_2 > 10°$ and $\theta_3 > -90°$ in figure 5.28 is the only part of the range worth considering. This is because the range $\theta_2 < 10°$ and $\theta_3 < -90°$ is affected by the acceleration of the crane arms and actuators which will not be correct in these tests due to the actuator properties have not been decided yet.

It is clear from figure 5.28a and 5.28b that the required force is from the lower actuator is about twice when the long arm is in use as when the short arm is in use. The actuator concept where one actuator is used for the short arm and two of the same actuator is used for the long arm is therefore the solution that is chosen. This means that it is only required to produce one type of the lower actuator, and that two of this actuator is used with the long arm.

The actuator mechanism consists mainly of an electric motor, gearbox and a trapezoidal screw, as shown in section 3.2.3. It is important that all of these components are considered together when the choice of these components is made. The components and the considerations made when the components where chosen are presented in the sections to come.

5.5.1 Trapezondial screw

A trapezoidal screw and nut is used to convert the rotational action of the electric motor to linear action at the actuator rod. The screw and nut purchased for this project is manufactured by Conti. There are several aspects worth considering when choosing the trapezoidal screw. Two of the most central aspects are the efficiency of the screw, and the self locking property of trapezoidal screws.

When choosing the trapezoidal screw it is obviously desirable to have as high efficiency as possible. The efficiency of the screw is a function of the lead angle and the coefficient of friction of the screw, ref figure 5.29.

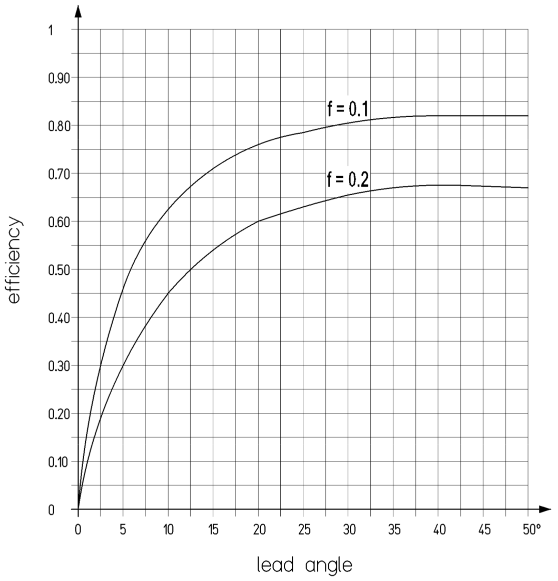


Figure 5.29: Trapezodial Screw Effisiency, Ref: Conti (2016)

The self locking property is almost as important as the efficiency of the screw, this is because a self locking screw will not need the electric motor to hold the crane static, thus requiring the motors to work continuesly. The self locking property is, as the efficiency, a function of the lead angle of the screw and the coefficient of friction between the nut and screw. The screw will be self locking when $M < 0$ in equation 5.3, Härkegård (2004).

$$M = rV\left(\frac{\tan(\beta) - \mu}{1 + \mu\tan(\beta)}\right) \tag{5.3}$$

Where r is the radius of the screw, V is the load which the screw mechanism is lifting, β is the screw lead angle, and μ is the coefficient of friction.

Equation 5.3 is solved for β when $M = 0$ to find the maximal lead angle that will make the screw self locking. The result is shown in equation 5.4

$$\beta = \tan^{-1}(\mu) \tag{5.4}$$

The friction coefficient will mainly depend on the surface finish, material and lubrication. While the material and their surface finish is known, the lubrication condition is unknown. The material of the trapezoidal nut is set to be bronze, and steel for the screw. The reason for changing the nut material from the self lubricating plastic used in Gyberg (2015) is due to the increased load on the screw, see the loading analysis of the screw below. The coefficient of friction at startup between bronze and steel is found to be ≈ 0.15 in Bouyer and Fillon (2011). This is for a well lubricated surface. It is not practical to have a constant oil supply to the screw, so the screw in the actuators will therefore be lubricated with grease. It is assumed that this will increase the friction coefficient compared to Bouyer and Fillon (2011). By assuming a friction coefficient of $\mu = 0.2$ the largest lead angle that will give a self locking screw is $\beta = 11.3°$, ref equation 5.4. The uncertainty of the friction coefficient is partly justified by the torque required to turn the electric motor and gearbox when no power is supplied. It is therefore concluded that a lead angle of $\beta < 10°$ will be sufficient to give a self locking screw mechanism.

Another property that is important to consider when choosing the trapezoidal screw and nut is that the screw and nut is able to withstand the load that they are subjected to. The relevant sizing criteria given by Conti (2016) is included in appendix G. This sizing criteria is used in order to choose a screw and nut that is able to withstand the loads it is subjected to.

The sliding condition is the first sizing criteria that is considered. This is done by calculating the product of the contact surface pressure, p, in N/mm^2 and the sliding speed, V_{st} in m/min. If the value of the product is below 21 the sliding condition is considered optimal with minimal wear of the nut. If the value is between 21 and 80 some wear of the nut must be expected and good lubrication is recommended. The maximum allowed value of the product $p * V_{st}$ is 250, but any operation when the value is above 80 is not recommended by Conti (2016). See appendix G.

The contact surface pressure, p, is calculated by equation 5.5

$$p = \frac{F[N]}{A_t[mm]} \tag{5.5}$$

Where F is the axial force that the trapezoidal screw is subjected to and A_t is the area of the treads of the trapezoidal nut.

The sliding speed V_{st} is calculated by equation 5.6

$$V_{st} = \frac{V_{tr}[m/min]}{\sin\beta} \tag{5.6}$$

Where V_{tr} is the linear speed of the trapezoidal nut.

The upper actuator is required to deliver 700N, and each of the lower actuators are required to deliver 1750N each, ref figure 5.28. The requirements for the screws is therfore set to be 2kN for the lower actuator and 1kN for the upper actuator when considering the sliding condition of the trapezoidal screw.

The axial load of the trapezoidal screw is also taken into account. This is done with the sizing criteria shown in appendix G. The free length of the screw is set to be 350mm and the constraint type is set to be type 1, as this will most accurately represent the actuator concept.

The critical rpm is also considered with the rpm sizing criteria shown in appendix G. The safety factor for critical rpm is set to be 4.

The following screw and nut have been chosen for all the actuators, ref Conti (2016).

Table 5.14: Trapezoidal Scew and Nut

Parameter	Value	Unit
Screw Product number	KUE22BR	
Nut Product number	FFR22BR	
Screw Diameter	22	mm
Screw Lead	10	mm
Screw Lead Angle	9.26	degrees
Nut Tread Surface Area	1030	mm^2

The selected screw and nut will give an upper limit for the linear sliding speed, V_{tr}, of the lower actuator nut of approximately 30 mm/s for $p*V_{st} = 21$ and 110mm/s for $p*V_{st} = 80$. The upper limit for the sliding speed of the upper actuator is 60mm/s for $p * V_{st} = 21$ and 220 for $p * V_{st} = 80$. The selected screw is able to withstand an axial load of 7 kN when a safety factor of 2 is applied, and rotational speed of at least 3000 rpm.

The motor and gearbox selected for the upper actuator will give a maximal rotational velocity of 374.5 rpm at the trapezoidal screw, ref. table 5.16 and 5.17. And 218.5 rpm for the trapezoidal screw of the lower actuator, ref. table 5.16 and 5.18. These values are both well below the critical rpm of the trapezoidal screw. The maximal sliding speed of the upper actuator will then be 62.4 mm/s for the upper actuator and 36.4 mm/s for the lower actuator. This means that the trapezoidal screw is expected to operate just outside of the optimal area when the crane is lifting the design load at maximum velocity. This is considered to be ok, as there is plenty of margin until the sliding condition will reach the level of $p * V_{st} = 80$.

To simulate the screw accurately it is important to consider the parameters that simulate the friction of the screw, ref equation 4.24. The static friction coefficient is all ready assumed to be $\mu_s = 0.2$, in the relation $F_{f_s} = F_N \mu_s$. The dynamic friction coefficient has to be smaller than the static as the friction is assumed to decrease when the screw starts moving. The dynamic or sliding friction coefficient is assumed to be $\mu_d = 0.1$, in the relation $F_{f_d} = F_N \mu_d$. The coefficient of dynamic friction is based on values supplied by Conti (2016) and should be considered as an assumption. Tests on the trapezoidal screw will have to be performed to determine this coefficient exactly. By using the static and dynamic frictional relations as guidelines, and knowing that the friction will increase slightly with higher velocities the friction model in equation 4.24 can be tuned to match the properties of the trapezoidal screw. The normal force at the screw in this model is set to be equal the actuator force. This is accurate as the parameters are tuned accordingly. The friction parameters are presented in table 5.15, and the friction as a function of sliding speed is illustrated in figure 5.30 where F_N is set to be 1750 N.

Table 5.15: Trapezoidal Screw Friction Parameters

Parameter	Value
s	100
μ_s	$1.7 \cdot 10^{-3}$
μ_c	$5 \cdot 10^{-4}$
μ_v	$1 \cdot 10^{-6}$

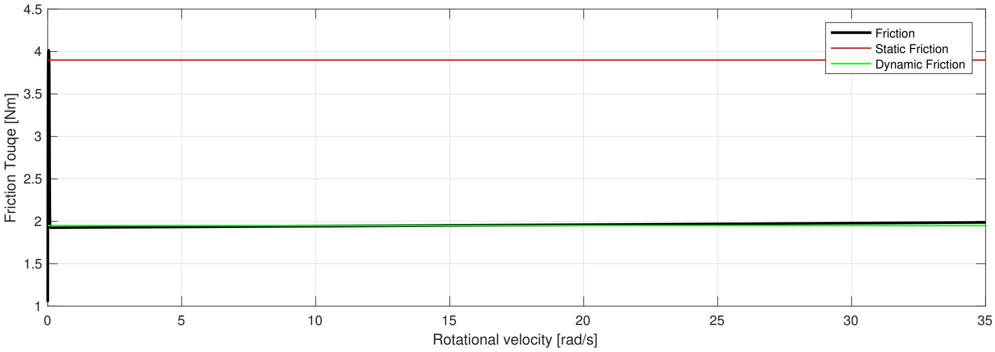


Figure 5.30: Trapezodial Screw Friction

5.5.2 Motor and Gearbox

Several aspects has been considered when the motor and gearbox has been chosen. The motors and gearboxes should be reliable and able to be controlled accurately. It is also beneficial that the motor and gearbox vendor supply enough information on the motor and gearbox that it is possible to implement the parts accurately in the simulation model. The vendor Maxson Motor has been chosen for this project.

It has been decided the the Z-postion of the crane tip should be able to follow the scaled wave condition presented in table 3.1 when performing the 4m high lift. This is in accordance with the requirement set in the framework which states that the crane should be able to facilitate testing and research of wave compensating control systems. To make sure that the crane will have the best possible basis for this sort of testing and research it is decided that the requirement should be fulfilled by only operating the upper actuator.

The wave hight, Z_{pos}, wave hight velocity, Z_{veloc}, and the wave height acceleration, Z_{acc}, of the wave in the scaled sea condition can be modeled as shown in equation 5.7

$$\begin{aligned} W_{Z_{pos}} &= 0.1cos\left(\frac{2\pi}{2.8}t\right) \\ W_{Z_{veloc}} &= -0.1\frac{2\pi}{2.8}sin\left(\frac{2\pi}{2.8}t\right) \\ W_{Z_{acc}} &= -0.1\left(\frac{2\pi}{2.8}\right)^2 cos\left(\frac{2\pi}{2.8}t\right) \end{aligned} \tag{5.7}$$

By setting the position controller, ref. section 4.3, to give input the upper actuator speed controller, the tip of the crane can be set to follow the wave signal. Different combinations of motors and gearboxes can then be tried out to find a combination that works well.

The requirements for the lower actuator is not as well defined. It is however desired that all the actuators are as similar as possible, it is therefore decided, if possible, that the electric motor and gearbox for both the upper and lower actuator should have the same diameter and lenght. This means all the parts of the actuators (except the top and bottom connection) will be interchangeable. The lower actuator(s) will have to, as a minimum requirement, be able to lift the design load with the upper actuator fully extended. This condition is tested for both the short and long lower arm, and a motor-geabox combination that works well is selected.

The result from the testing of motor-geabox combinations in the simulation model is that the same electric motor can be is used for all actuators, and only the gearbox needs to be different for the upper and lower actuator(s). The difference in the gearbox is only the reduction, and the outer dimensions of the gearbox is identical for both the upper and lower actuator(s). This adds to simplicity and modularity which is one of the design requirements of the crane. The motor, and its properties, is presented in table 5.16, see MaxsonRE50 (2016) for more details on the motor. The gearbox, and its properties, is presented in table 5.17 and 5.18, see MaxsonGP52C (2016) for more details on the gearboxes.

Table 5.16: Actuator Motor Properties

Parameter	Value	Unit
Motor Model / Number	RE 50 / 370355	
Nominal voltage	36	V
Max Continues Current	7.1	A
Max Continues torque	418	mNm
No load speed	5680	rpm
Terminal resistance	0.244	Ω
Terminal inductance	0.177	mH
Torque constant	60.4	mNm/A
Speed constant	158	rpm/V
Rotor inertia	5.6e−5	kgm^2
Wheight	1.1	kg

Table 5.17: Upper Actuator Gearbox Properties

Parameter	Value	Unit
Gearbox Model / Number	GP 52C / 223084	
Reduction	$\frac{91}{6}$	
Max Continues torque	15	Nm
Efficiency	83	%
Inertia	16.8e−7	kgm^2
Wheight	0.6	kg

Table 5.18: Lower Actuator(s) Gearbox Properties

Parameter	Value	Unit
Gearbox Model / Number	GP 52C / 223087	
Reduction	26	
Max Continues torque	15	Nm
Efficiency	83	%
Inertia	9.1e−7	kgm^2
Wheight	0.6	kg

The selected motors are equipped with encoders in order to read the speed of the motor which will give a good basis for implementing different control systems in the crane lab. The encoders is presented in table 5.19, see also MaxsonHEDS5540 (2016) for more information on the encoders.

Table 5.19: Actuator Motor Encoder Properties

Parameter	Value	Unit
Encoder Model / Number	HEDS 5540 / 110517	
Counts per turn	500	
Number of channels	3	
Max. operating frequency	100	kHz
Max. speed (rpm)	12000	rpm

5.5.3 Actuator Design

The actuator design is developed with several iterations with basis in the conceptual design, simulations performed when the the trapezoidal screw and motor-gearbox combinations where selected and the FEA simulation presented in this section. The goal is to design an actuator that is as compact as possible while being strong enough to ensure that the structure of the actuator can handle the forces it is exposed to. It is also important that the length and elongation is seen in correlation with the crane arm design so that the crane will have a sensible movement range.

The length of the trapezoidal screw is what determines the elongation of the actuator. It is found beneficial to extend the elongation from the previously assumed 240mm to 256mm. The angular range of the crane with the actuator design presented in figure 5.38 is then as illustrated in figure 5.31

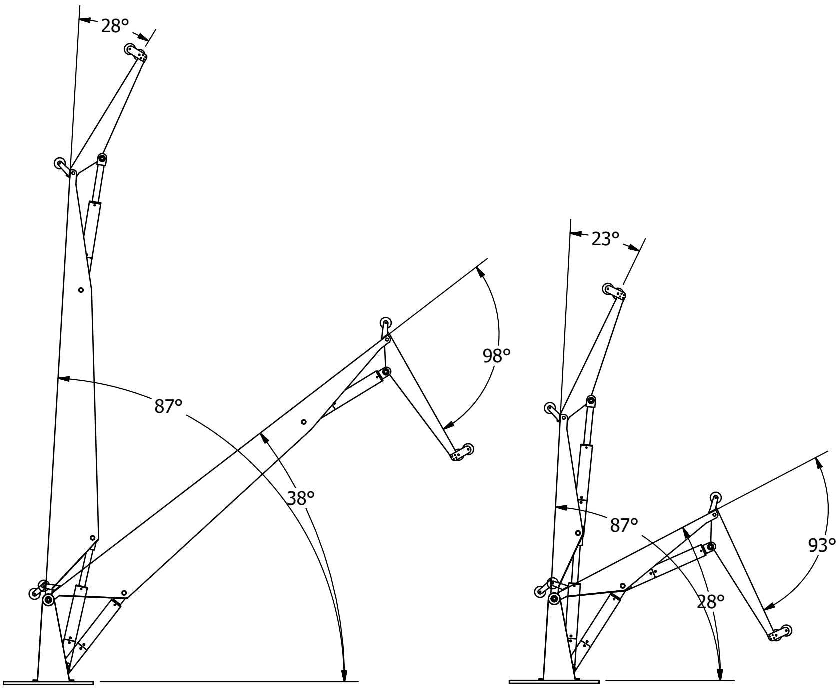


Figure 5.31: Crane Angular Movement Range

A simplified version of the upper and dual lower actuator is used in the structural analysis, see figure 5.32 and 5.33. The actuators are simplified by removing the motor and gearbox, the actuator cover and the bussing in the end of the actuator cover. These parts will not contribute much to the strength of the actuator, and removing them will make the analysis simpler and faster. Their contribution to the gravitational load are however included. The actuators are placed vertically and loaded with a vertical load of 6kN for the dual actuator and 3kN for the upper actuator. The movement of the actuators are constrained in the radial direction at the lower connection point.

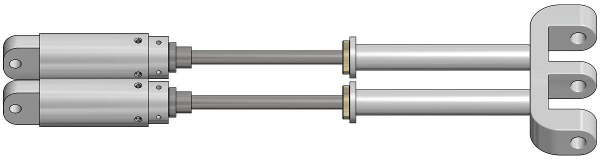


Figure 5.32: Simplified Dual Lower Actuator for Structural Analysis

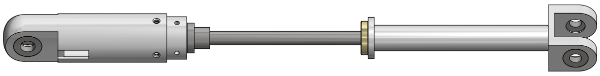


Figure 5.33: Simplified Upper Actuator for Structural Analysis

The upper and lower actuators are with the exception of the lower and upper connection brackets built up identical, this is illustrated best by figure 3.7 and 5.38. All locally produced parts is made from solid aluminium sections except the screw fastener and the bushing which is made of steel, see table 5.1 and 5.3 for material properties. The lower connection bracket for the upper actuator is slightly larger then for the lower actuator. The reason for the increased size for the lower connection bracket for the upper actuator is that the upper actuator has to have roller bearings in its connection to the lower arm. This is in order to have a "frictionless" connection with the short lower arm configuration. This is because the the upper connection of the lower actuator and the lower connection of the upper actuator share the same connection point when the short lower arm is in use. The upper brackets is designed as shown in section 3.2.3 and made from solid aluminium sections.

The final structural analysis is presented in figure 5.34 and 5.35.

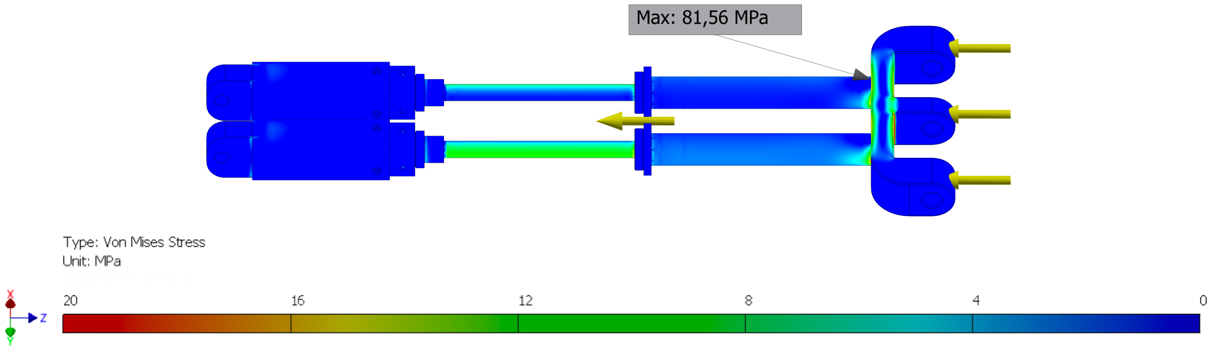


Figure 5.34: Lower Actuator Von Mises Stress

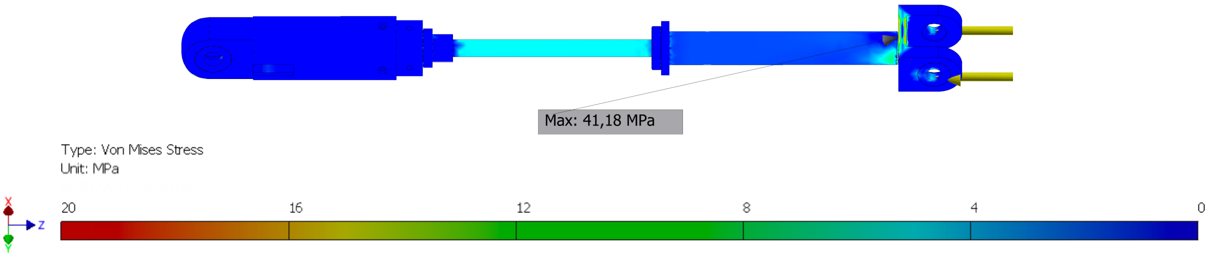


Figure 5.35: Upper Actuator Von Mises Stress

The structural analysis prove that the strength of both the upper and dual lower actuator is more than adequate. The strength of the single lower actuator is also considered adequate as the single lower actuator will be more or less similar only with an even stronger upper bracket then the single upper actuator. The stresses is found to be at the highest level in the connection with the upper actuator bracket and the actuator rod. Where it is mostly varying around 10MPa for both the upper and dual lower actuator. This will give a safety factor of 13 with regard to yield strength, ref. table 5.1 for the material properties. The highest stress is found to be 41 MPa for the upper actuator and 82 MPa for the dual lower actuator which is also well below the yield strength of the material. It should also be noted that this is a small stress concentration in the connection point between the upper actuator bracket and the actuator rod. The reason that the stress is higher in the dual lower actuator is assumed to be because of the bending moment that is induced by the wider dual actuator bracket.

In order to verify the structural analysis a convergence analysis and an evaluation of the reaction forces is done.

For the convergence analysis the mash of the parts is refined as explained in section 2.4 to ensure that the simulation converges, see figure 5.36 and 5.37 for plot of the result from the convergence analysis.

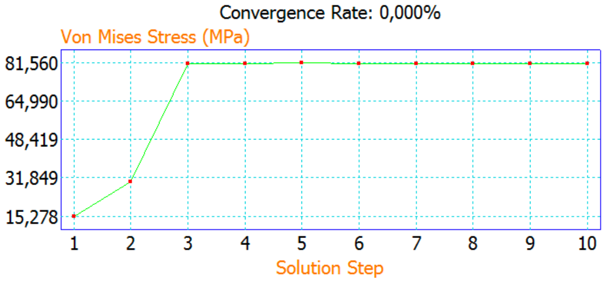


Figure 5.36: Lower Actuator FEA Convergence

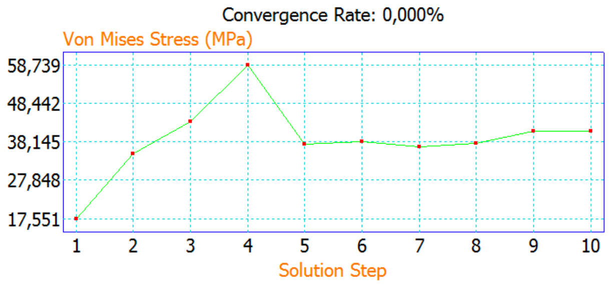


Figure 5.37: Upper Actuator FEA Convergence

From figure 5.36 and 5.37 it is clear that further refinement of the mash will not yield any higher stress, thus the analysis converges.

The reaction forces are presented in table 5.20.

Table 5.20: Actuator Reaction Forces

Parameter	Size	Unit
Lower Actuator Z-direction	6.12	kN
Upper Actuator Z-direction	3.07	kN

The weight of the dual lower actuator equals a gravitational load of approximately 0.12 kN, ref table 5.21, and the weight of the upper actuator equals a gravitational load of 0.07 kN, ref table 5.23. A control of the reaction forces is presented in equation 5.14

$$\begin{aligned}\sum F_{Z_{LowerActuator}} &= 6.12kN - 6.00kN - 0.12kN = 0kN \\ \sum F_{Z_{UpperActuator}} &= 3.07kN - 3.00kN - 0.07kN = 0kN\end{aligned} \tag{5.8}$$

This analysis concludes that the result of the structural analysis is probable, and at the structure of the actuators satisfies the requirement that the structure should be able to withstand a much greater load than the design load.

The design of the actuators is presented in figure 5.38 and the design properties in table 5.21, 5.22, and 5.23. All dimensions are in mm in figure 5.38.

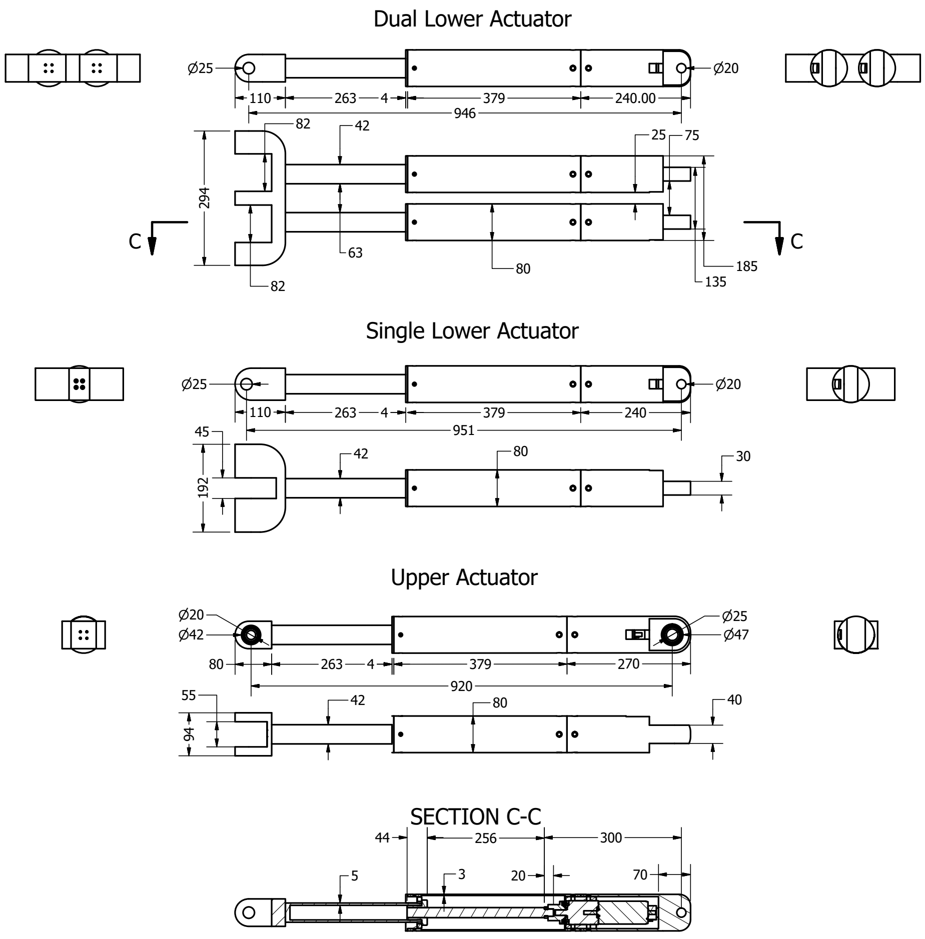


Figure 5.38: Actuator Final Design, all dimensions are in mm.

Table 5.21: Dual Lower Actuator

Parameter	Size	Unit
Total Mass, CM_4	11.8	kg
Centre of gravity, p	429	mm
Rotational inertia , I_{xx}	2.9	kgm^2
Rotational inertia, I_{zz}	3.0	kgm^2
Top Bracket Weight	2.6	kg

Table 5.22: Single Lower Actuator

Parameter	Size	Unit
Total Mass, CM_4	8.5	kg
Centre of gravity, p	440	mm
Rotational inertia , I_{xx}	2.3	kgm^2
Rotational inertia, I_{zz}	2.3	kgm^2
Top Bracket Weight	2.6	kg

Table 5.23: Upper Actuator

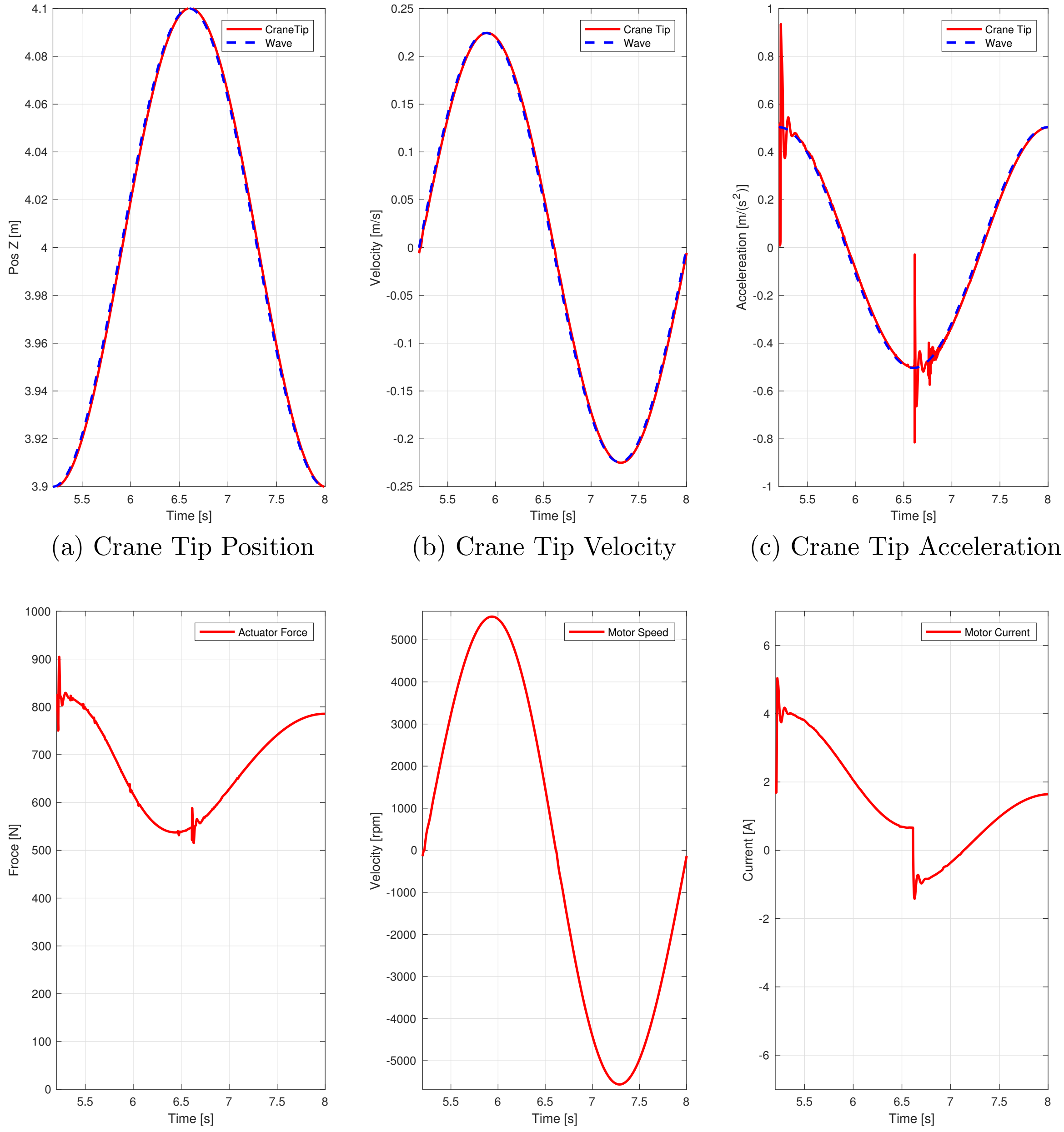
Parameter	Size	Unit
Total Mass, CM_4	6.8	kg
Centre of gravity, o	324	mm
Rotational inertia , I_{xx}	1.1	kgm^2
Rotational inertia, I_{zz}	1.1	kgm^2
Top Bracket Weight	0.4	kg

Table 5.24: Common Properties

Parameter	Size	Unit
Screw Ass. Rot.Inert.	1.1e−4	kgm^2
Rod and Nut Weight	0.9	kg

Some of the test results produced by the simulation model showing the anticipated capability of the upper and lower actuator(s) are presented below. All the presented simulation results are the final results with with all the selected design parameters concerning the actuators implemented.

Figure 5.39 shows the result from the simulations done when the motor and gearbox for the upper actuator where selected. The upper arm is set to follow a cosine wave related to the scaled sea condition, see equation 5.7, with the design load of 16kg attached. The lower arm is set static at $\theta_2 = 80°$.

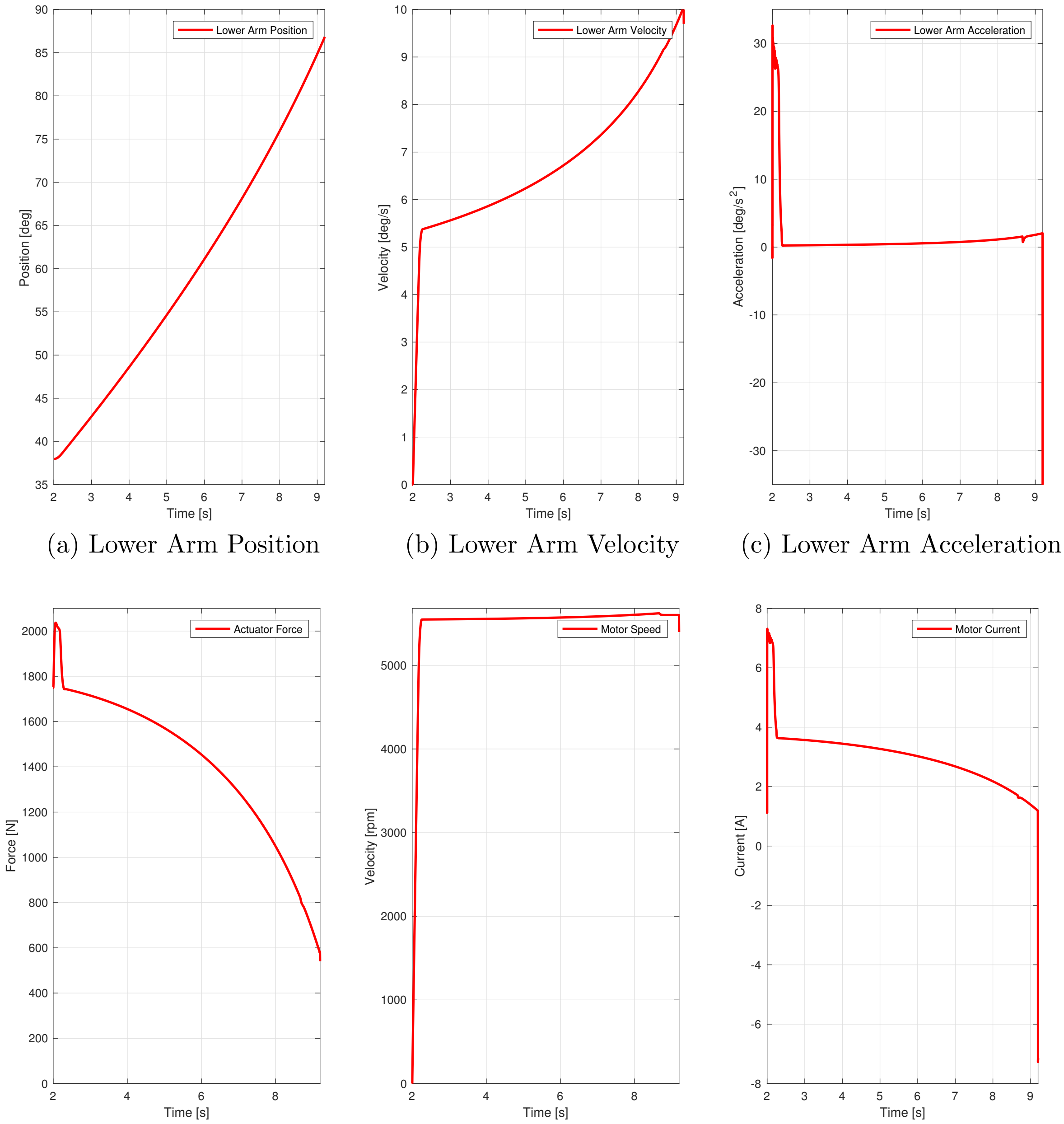


(a) Crane Tip Position (b) Crane Tip Velocity (c) Crane Tip Acceleration

(d) Upper Actuator Force (e) Motor Rotational Velocity (f) Motor Current Usage

Figure 5.39: CraneTip Following Wave Pattern, only Upper Actuator Active

Figure 5.40 shows the result from the simulations done when the motor and gearbox for the lower actuator where selected. The simulation is performed for the configuration with the long lower arm with the dual lower actuator. Note that the graphs showing data for the actuator is only displaying data for one actuator. The results are similar for the short lower arm with the single lower actuator. The upper arm is static at $\theta_3 = -28°$, and the lower arm is stating in the lower position, and lifting until the upper position with the design load of 16kg attached.



(a) Lower Arm Position

(b) Lower Arm Velocity

(c) Lower Arm Acceleration

(d) Lower Actuator Force

(e) Motor Rotational Velocity

(f) Motor Current Usage

Figure 5.40: Long Lower Arm Lifting, only Lower Actuators Active

5.6 Base and Winch Assembly

The size of the base is is a result of several iterations where the goal was to be able to fit all components, while still keeping the base as compact as possible. The size of the base is of course seen together with the width of the lower crane arm, and the available slewing ring sizes. Rollix has been selected as a supplier of the slewing ring, see Rollix (2016) for details on their product range. With these considerations the diameter of the base is set to 620mm.

5.6.1 Base Rotation

As explained in section 3.2.1 the rotation of the base will be provided with a slewing ring with an internal gear. An electric motor will then be connected to the gear and thus be able to turn the crane.

Slewing Ring

The slewing ring is selected mostly with regard to size, but the friction and the load capacity of the slewing ring has also been considered.

The slewing ring is mainly exposed to an axial load and a tilting moment. The static tilting moment can be estimated to be the static torque at θ_2 resulting from the gravitational forces from both arms and actuators and the load suspended from the crane tip. For now the base is assumed not to contribute to the tilting moment. The maximal static tilting moment (long lower arm configuration) is obtained by the simulation model to be 821 Nm. A safety factor of three is applied to be sure that the dynamic tilting moment will not exceed the estimate. By applying the safety factor the tilting moment is set to be approximately 2.5kNm. The static axial load can be estimated to be the total weight of the crane. The base is at this point estimated to have a weight of 50kg, this will result in a total weight with the design load attached of approximately 130kg, resulting in a static axial load of approximately 1.3kN. As the slewing ring will have to be designed for a dynamic load the static load is multiplied with a safety factor of three, as with the tilting moment, resulting in a axial load of 4kN. It is clear that with these estimates that all the slewing rings supplied by Rollix will be more than strong enough to withstand the loads the crane will be able to apply, see Rollix (2016) for load curves of the slewing rings.

Rollix supply mainly two series of slewing rings, a standard series, and a light series. The light series is the most compact series with the lowest weight and the lowest friction. It is therefore deiced that the light series slewing ring will be the best choice. The properties of the slewing ring is presented in table 5.25.

Table 5.25: Slewing Ring Properties

Parameter	Value	Unit
Model Number	32 0541 01	
Outer Diameter	616	mm
Number of Teeth	76	
Weight Total	43	kg
Weight Outer Ring	23	kg
Rotational Inertia Outer Ring, I_{zz}.	2.2	kgm^2

The friction of the slewing ring is given by Conti (2016) to be a sum of the unloaded friction, Crv, given by figure 5.41 and the friction related to the load of the slewing ring, Crc, given by equation 5.9.

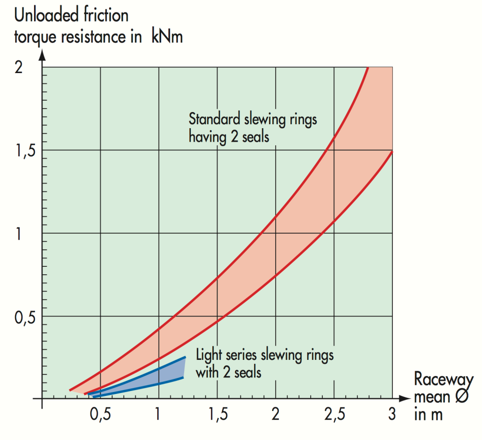


Figure 5.41: Slewing Ring Unloaded Friction

$$Crc = \left(\frac{13.11 M_T}{\varnothing_m} + 3F_A + 11.34F_R\right) \varnothing_m \, 10^{-3} \quad [kNm] \tag{5.9}$$

Where M_T is the tilting moment in kNm, F_A is the axial force in kN, F_R is the radial force in kN, and $\varnothing_m$ is the raceway mean diameter in m.

By using the conservative estimates for M_T and F_A given above, Crc can be determined to be 40 Nm. Crv is also determined to be approximately 40Nm by the use of figure 5.41. The total friction of the slewing ring is therefore determined to be approximately 80Nm. This value of the friction is considered conservative, and without any test results of the actual slewing ring it will be difficult to produce a more accurate estimate of the friction. The friction torque as a function of the rotation speed is not given by Conti (2016), this means that it will have to be estimated. The friction as a function of the rotational speed is modeled by the same model as the friction of the trapezoidal screws, see equation 4.24. The parameters of equation 4.24 have been tuned to estimate a friction where the static friction is only slightly higher than the coulomb friction, and the viscous friction will increase slightly with increasing velocity. The frictional torque obtained by the framework in Conti (2016) is used as a guideline. The result is presented in table 5.26 and figure 5.42.

Table 5.26: Slewing Ring Friction Parameters

Parameter	Value
F_N	1000
s	1000
μ_s	0.08
μ_c	0.07
μ_v	0.005

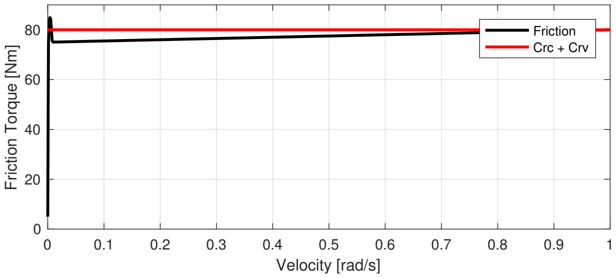


Figure 5.42: Slewing Ring Friction

Motor and Gear

It is desired that the rotation of the base is fast and responsive. A pendular movement of the load suspended from the crane tip has to be expected and It is determined that the base rotation should be able to stabilise a pendulum movement occurs happens when the load is moved 10° from the vertical plane, and hanging 1m below the crane tip. This requirement is simplified to a requirement at that the horizontal movement of the crane tip should be able to follow the horizontal movement of the pendulum when both crane arms are positioned in the their highest positions.

As the angular movement of the pendulum is described by equation 5.10, the horizontal movement of the pendulum can be described by equation 5.11

$$\theta = \theta_{max} \sin\left(\sqrt{\frac{g}{L}}t\right) \tag{5.10}$$

$$X_{pos} = \sin\left(\theta_{max} \sin\left(\sqrt{\frac{g}{L}}t\right)\right) \tag{5.11}$$

This requirement is used to determine the motor and gear that is used to provide the base rotation. Several tests have been performed with different combinations of motors and gears. The motor and gearbox presented in table 5.27 and 5.28 are results of the tests. The gearbox is connected to a gear with 13 teeth that connects the motor-gearbox to the slewing ring. The gear has a rotational inertia of 3.53e−4 kgm^2. See MaxsonRE50 (2016) and MaxsonGP52C (2016) for more information on the selected motor and gearbox.

Table 5.27: Base Motor Properties

Parameter	Value	Unit
Motor Model / Number	RE 50 / 370355	
Nominal voltage	36	V
Max Continues Current	7.1	A
Max Continues torque	418	mNm
No load speed	5680	rpm
Terminal resistance	0.244	Ω
Terminal inductance	0.177	mH
Torque constant	60.4	mNm/A
Speed constant	158	rpm/V
Rotor inertia	5.6e−5	kgm^2
Wheight	1.1	kg

Table 5.28: Base Gearbox Properties

Parameter	Value	Unit
Gearbox Model / Number	GP 52C / 223099	
Reduction	186	
Max Continues torque	30	Nm
Efficiency	68	%
Inertia	17.7e−7	kgm^2
Wheight	0.9	kg

5.6.2 Winch Assembly

The winch assembly is the crane component that will enable the crane to hoist and lower the cargo hanging from the wire. The assembly will be build up as shown in section 3.2.1.

The reel is designed in such a way that it can hold a sufficient amount of wire/rope, while not being too large. It has also been important that the reel will be dimensioned so that it can accommodate varoius wire/rope types and dimensions.

If a steel wire is used it is proposed that is has a diameter of 2mm. A steel wire with this diameter will have a static tension of approximately 50 MPa with a 16kg load, and a weight of approximately 25g/m. The wire can be selected with an even smaller diameter but that is thought to be impractical. If a rope is used, it should not exceed a diameter of 10 mm in order to fit properly on the reel.

Several design parameters will have to be determined related to the winch. These are the dimensions of the reel, the reduction in the belt transmission, and the properties of the motor and gearbox driving the winch. Several different combinations of these parameters have been tested in the simulation model under the criteria that the winch should be able to heave compensate for the scaled wave condition, see table 3.1 and equation 5.7 for the wave condition.

The result of the simulations has resulted in the reel dimensions presented in table 5.29, the belt transmission presented in table 5.30, and motor and gearbox combination presented in table 5.31 and 5.32. See MaxsonEC60 (2016) and MaxsonGP52C (2016) for more details on the motor and gearbox.

Table 5.29: Reel Properties

Parameter	Value	Unit
Total Outer Diameter	200	mm
Full Reel Diameter	100	mm
Empty Reel Diameter	40	mm
Internal Width	130	mm
Maximal Amount of Stored Wire, 2mm wire/ 10mm Rope	191/7,6	m

Table 5.30: Belt Transmission Properties

Parameter	Value	Unit
Pulley at Reel Diameter	120	mm
Pulley at Reel Rotational Inertia	$7.4\text{e}{-4}$	kgm^2
Pulley at Motor Diameter	60	mm
Pulley at Motor Rotational Inertia	$5.5\text{e}{-5}$	kgm^2
V-belt type	Z25	

Table 5.31: Winch Motor Properties

Parameter	Value	Unit
Motor Model / Number	EC 60 / 411678	
Nominal voltage	24	V
Max Continues Current	5.5	A
Max Continues torque	289	mNm
No load speed	4250	rpm
Terminal resistance	0.307	Ω
Terminal inductance	0.188	mH
Torque constant	53.4	mNm/A
Speed constant	179	rpm/V
Rotor inertia	12.1e−5	kgm^2
Wheight	0.47	kg

Table 5.32: Winch Gearbox Properties

Parameter	Value	Unit
Gearbox Model / Number	GP 52C / 223089	
Reduction	43	
Max Continues torque	30	Nm
Efficiency	75	%
Inertia	17.3e−7	kgm^2
Wheight	0.77	kg

A brake is fitted in order to secure the load if the motor or belt transmission should fail, the brake is also practical for the control system of the winch because the motor will not have to hold the load static.

The brake system will consist of a disk brake actuated by a servo as shown in section 3.2.1. The brake calliper will have a reduction in such a way that movement of the calliper arm will result in a smaller movement at the brake pads. This will function as a amplifier for the force supplied to the brake pads, compared to the force supplied to the calliper arm. The total braking torque can then be determined by equation 5.12

$$T_{bracing} = \mu_b F_{pads} r_{disc} \tag{5.12}$$

A disk brake from a bicycle is tested in the simulation model and found to be more than powerful enough. The brake will consist of a 160mm rotor and a mechanical calliper, see AvidBB5 (2016). The reduction in the calliper is measured to be approximately 13, which means that 20mm movement at the calliper arm, will result in approximately 1.5mm movement of the brake pads. The coefficient of friction between the brake pads and the disk brake is difficult to determine without any testing. This coeficient is in the design face assumed to be $\mu_b = 0.1$ which is considered conservative. The total braking moment can then be determined by equation 5.12

The brake is controlled by a wire connected to a servo which has a max torque of 40kg/cm, see PowerHD-1235MG (2016). This means that with a 10mm servo arm, the total force delivered to the brake calliper is approximately 390N. This will yield a braking force of approximately 5kN at the brake pads.

By assuming a friction coefficient of $\mu_b = 0.1$ the maximum braking torque can be estimated to be 40Nm with equation 5.12. This estimate is then used to calibrate the friction model for the brake. The friction parameters chosen is presented in table 5.33, and illustrated in figure 5.43

Table 5.33: Winch Brake Friction Parameters

Parameter	Value
F_N	5000
s	1000
μ_s	0.008
μ_c	0.006
μ_v	0.0002

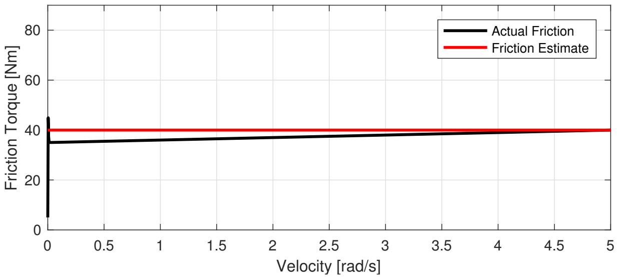


Figure 5.43: Winch Brake Friction

A simplified version of the winch assembly is used in order to evaluate the structural strength, see figure 5.44. The winch assembly is simplified by removing the brake, the belt transmission, the winch motor and gearbox, the brake and replacing the bearings with solid steel representations, this is done to simplify the analysis and reduce the computational time required to perform the analysis. The winch assembly is constrained in all directions on the face that connects the winch with the base. The rotation at the brake bracket is also constrained. The reel is loaded with a vertical load of -500N and a torque of 120Nm, this is approximately three times the load when the brake is applied.

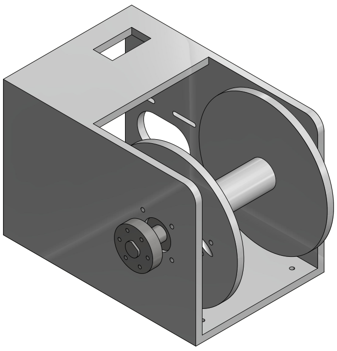


Figure 5.44: Simplified Winch Assembly for Structural Analysis

The frame for the winch assembly is built up by 10mm aluminium plates. The reel is built up by a 40mm aluminium solid cylinder which is machined for the 20mm steel rod it rides on. The 20mm steel rod is also machined to have a flat side, this is in order to transfer the torque from the brake and belt transmission to the reel. The bake will be connected to the steel rod with a solid steel bracket.

The final structural analysis is presented in figure 5.45 and 5.46.

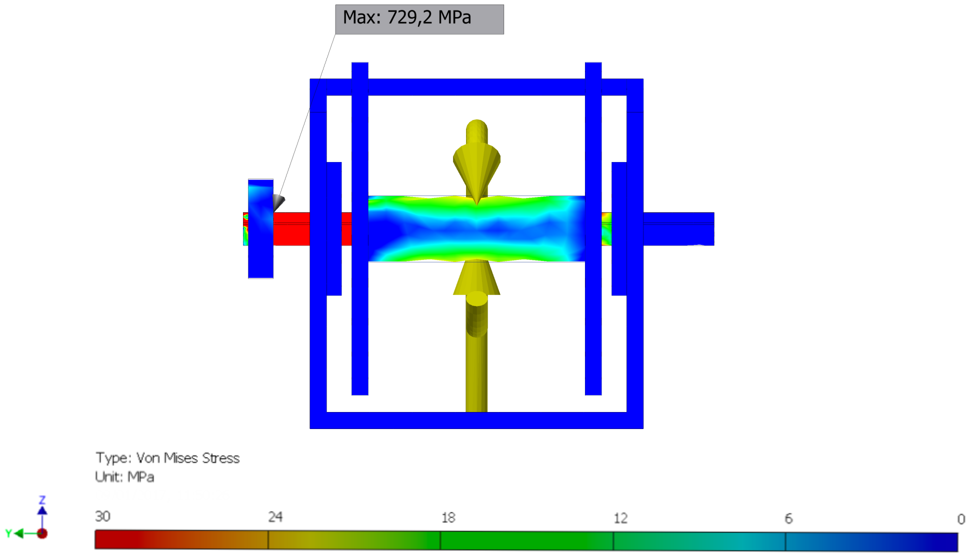


Figure 5.45: Winch Assembly Von Mises Stress

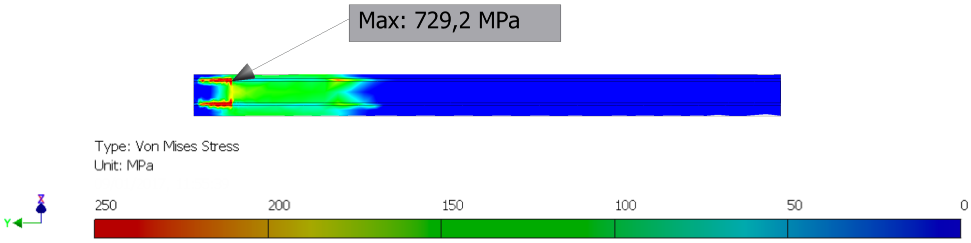


Figure 5.46: Reel Rod Von Mises Stress

The structural analysis prove that the strength of the winch assembly is more than adequate. The stresses in the frame for the winch assembly is fond to be below 5 MPa, and the stresses in the reel reach approximately 20 MPa. This will give the reel a safety factor of 6.5, ref table 5.1 for the material properties. The rod that connects the reel with the belt transmission and the brake will see the highest stresses. The stresses in the rod is mostly due to the torque of 120Nm that is applied. The stress in the rod is mostly varying around 150 MPa, giving the rod a safety factor of 5, ref table 5.3 for the material properties. The highest stress is found to be 729 MPa which is also below the yield strength of the material. It should also be noted that this is a small stress concentration in the connection point between the bracket that holds the brake disk and the rod, and that it is highly unlikely that the brake is able to produce 120Nm of torque.

In order to verify the result of the structural analysis a convergence analysis and an evaluation of the reaction forces is done.

For the convergence analysis the mash of the parts is refined as explained in section 2.4 ensure that the simulation converges, see figure 5.47 for plot of the result from the convergence analysis.

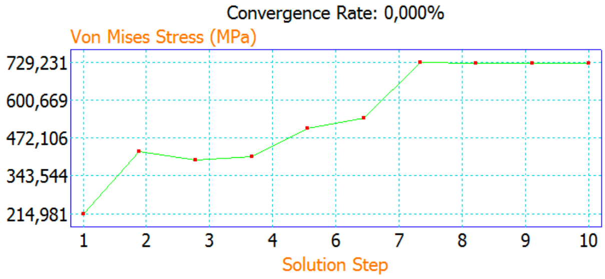


Figure 5.47: Winch Assembly FEA Convergence

From figure 5.47 it is clear that further refinement of the mash will not yield any higher stress, thus the analysis converges.

The reaction forces are presented in table 5.34.

Table 5.34: Winch Assembly Reaction Forces

Parameter	Size	Unit
Winch frame Z-direction	-500	N
Torque at brake disk bracket	-120	Nm

A control of the reaction forces is presented in equation 5.13

$$\begin{aligned}\sum F_Z &= 500N - 500N = 0kN \\ \sum M_{BrakeDiskBracket} &= 120Nm - 120Nm = 0Nm\end{aligned} \tag{5.13}$$

It is with the verification of the analysis concluded that the result of the structural analysis is probable, and at the structure of the winch assembly satisfies the requirement that the structure should be able to withstand a much greater load than the design load.

The bearings that connects the reel rod to the winch frame is the same as the one used for the connection between the upper and lower arm, see table 5.8.

5.6.3 Base Structural Design

The base structural design is developed with several iterations with basis in the conceptual design presented in section 3.2.1 and the FEA simulation presented in this section.

A simplified version of the base is used in the structural analysis, see figure 5.48. All parts except the frame from the winch assembly has been removed, but the weight of the complete winch assembly is added to the gravitational load. The slewing ring, motor and gearbox for base rotation and the lower base plate has also been removed. The roller bearings have been replaced with solid steel representations. The base is subjected to gravity and the reaction forces from the long lower arm. The base is constrained in Z direction at the outer boundary at the bottom of the upper base plate.

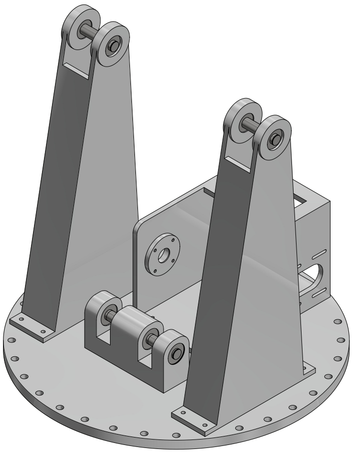


Figure 5.48: Simplified Base for Structural Analysis

The structure of the base is built up mostly with 10mm and 20mm aluminium plates. The upper and lower base plate is 20mm, and the plates that constitutes the lower arm connection brackets are 10mm, as are the plates that constitutes the frame of the winch assembly. The lower actuator connection bracket is made from solid aluminium. The small rods that connects the lower actuator and the lower arm to the base is made of a 20mm steel rod, this is to ensure small deflections and stresses at this critical point.

The final structural analysis is presented in figure 5.49.

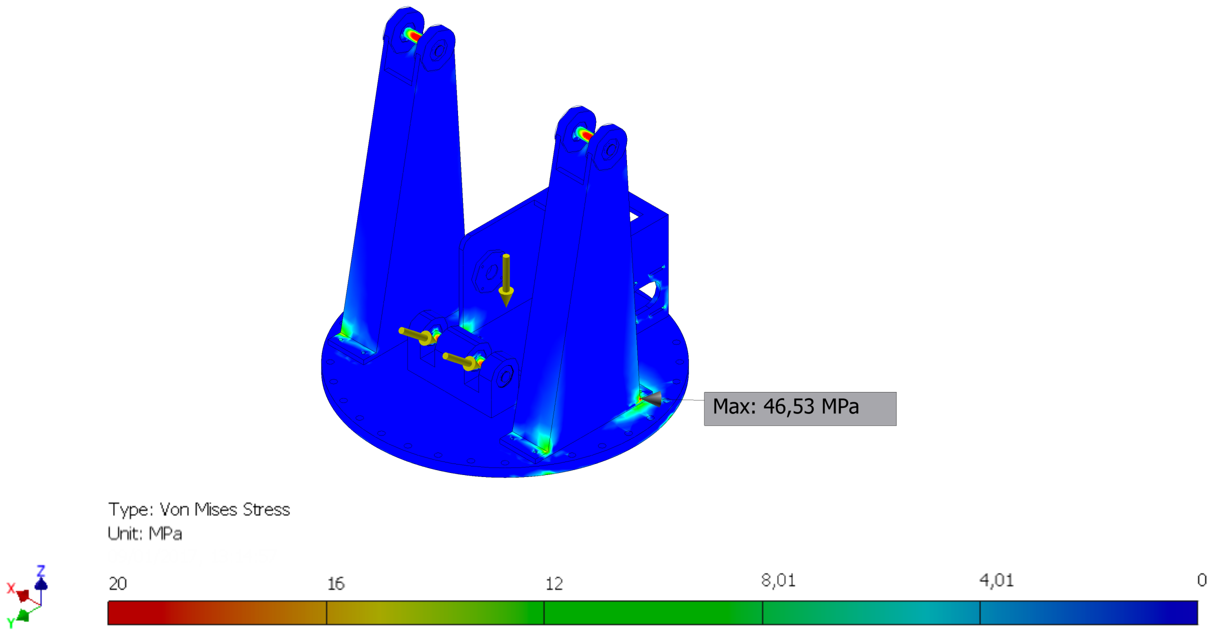


Figure 5.49: Base Von Mises Stress

The structural analysis prove that the strength of the base is more than adequate. The highest stress levels in the aluminium parts is fond in the connection between the the lower arm brackets and the upper base plate. The stresses are in these areas approximately 10 MPa, with some small stress concentrations reaching up to 46.5 MPa. A stress of 10 MPa will give the structure a safety factor of 13. The stresses in the rods is varying around 30 MPa, giving the rods a safety factor of 25 with regard to yield strength.

In order to verify the structural analysis a convergence analysis and an evaluation of the reaction forces is done.

For the convergence analysis the mash of the parts are refined as explained in section 2.4 to ensure that the simulation converges, see figure 5.50.

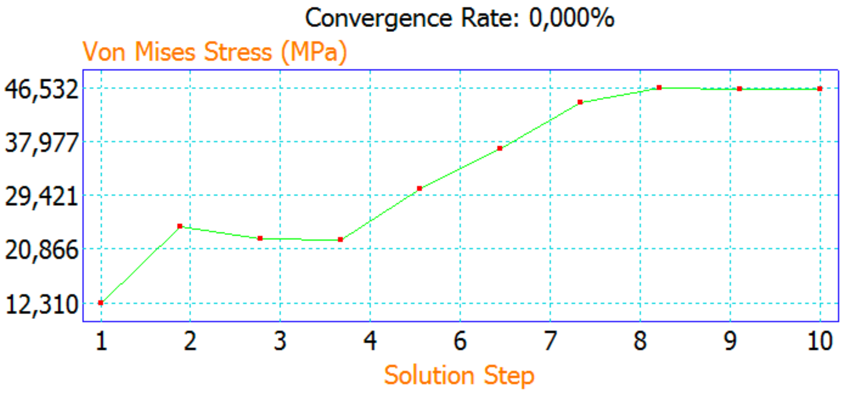


Figure 5.50: Base FEA Convergence

From figure 5.36 and 5.37 it is clear that further refinement of the mash will not yield any higher stress, thus the analysis converges.

The reaction forces are presented in table 5.35.

Table 5.35: Base Reaction Forces

Parameter	Size	Unit
Upper Base Plate Z-direction	-1.18	kN

The weight of the base when the lower base plate and the slewing ring is removed equals a gravitational load of approximately 430N, ref table 5.36 and 5.25. A control of the reaction forces is presented in equation 5.14

$$\sum F_Z = 4.3kN - 5.05kN - 0.43kN - 1.18kN = 0kN \tag{5.14}$$

It is with this analysis concluded the result of the structural analysis is probable, and the structure of the base satisfies the requirement that the structure should be able to withstand a much greater load than the design load.

5.6.4 Base Final Design Result

The final design of the base and winch assembly is presented in figure 5.51 and table 5.36. All dimensions are in mm in figure 5.51.

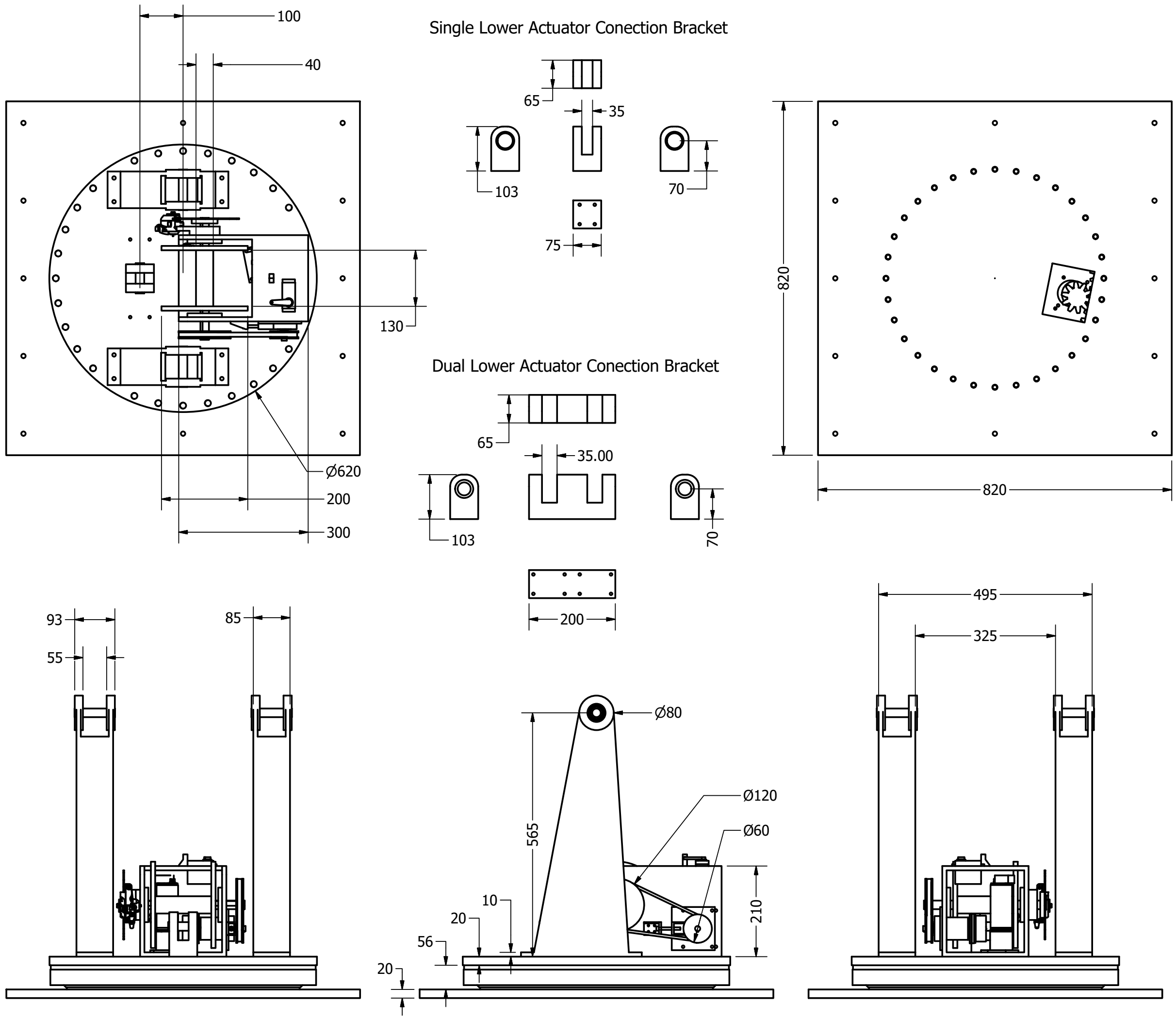


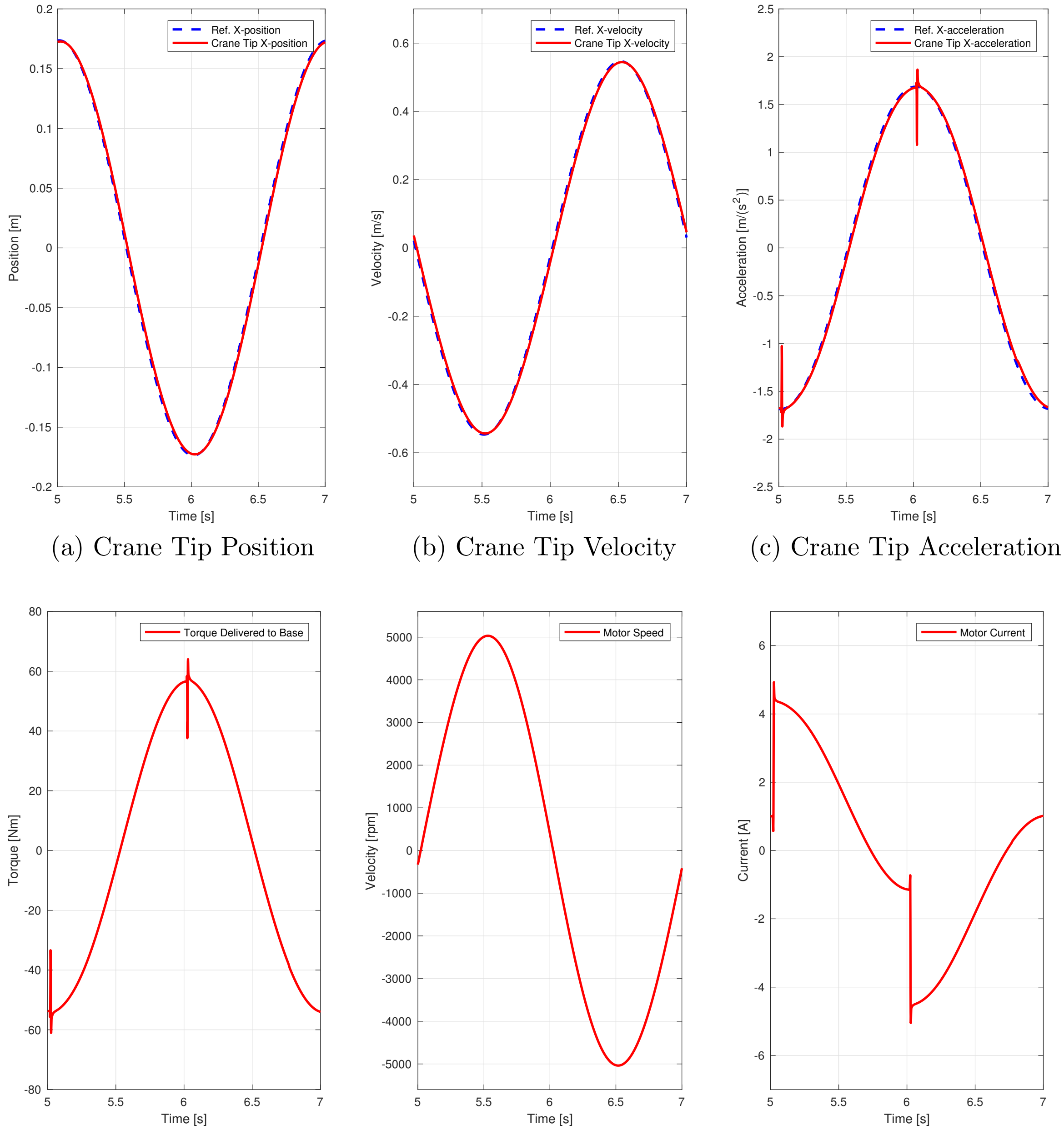
Figure 5.51: Base Final Design, all dimensions are in mm.

Table 5.36: Base Properties

Parameter	Size	Unit
Total Mass,	122	kg
Mass Rotating part, CM_1	67	kg
Centre of gravity, X_{CM_1}	0	mm
Centre of gravity, Y_{CM_1}	-23	mm
Rotational Inertia Base Rotation, I_{zz}	1.8	kgm^2
Rotational Inertia Reel,	9.9e−3	kgm^2

Some of the test results produced by the simulation model showing the anticipated capability of the base rotation and the winch are presented below. All the presented simulation results are the final results with with all the selected design parameters concerning the base and winch implemented.

Figure 5.52 shows the final test results obtained when the motor and gearbox for the base rotation were selected. The X-positon of the crane tip is set to follow the pendulum movement of the load, see equation 5.11. The crane is configured with the long lower arm, and both arms are placed in their upper positions, $\theta_2 = 87°$ and $\theta_3 = -28°$.

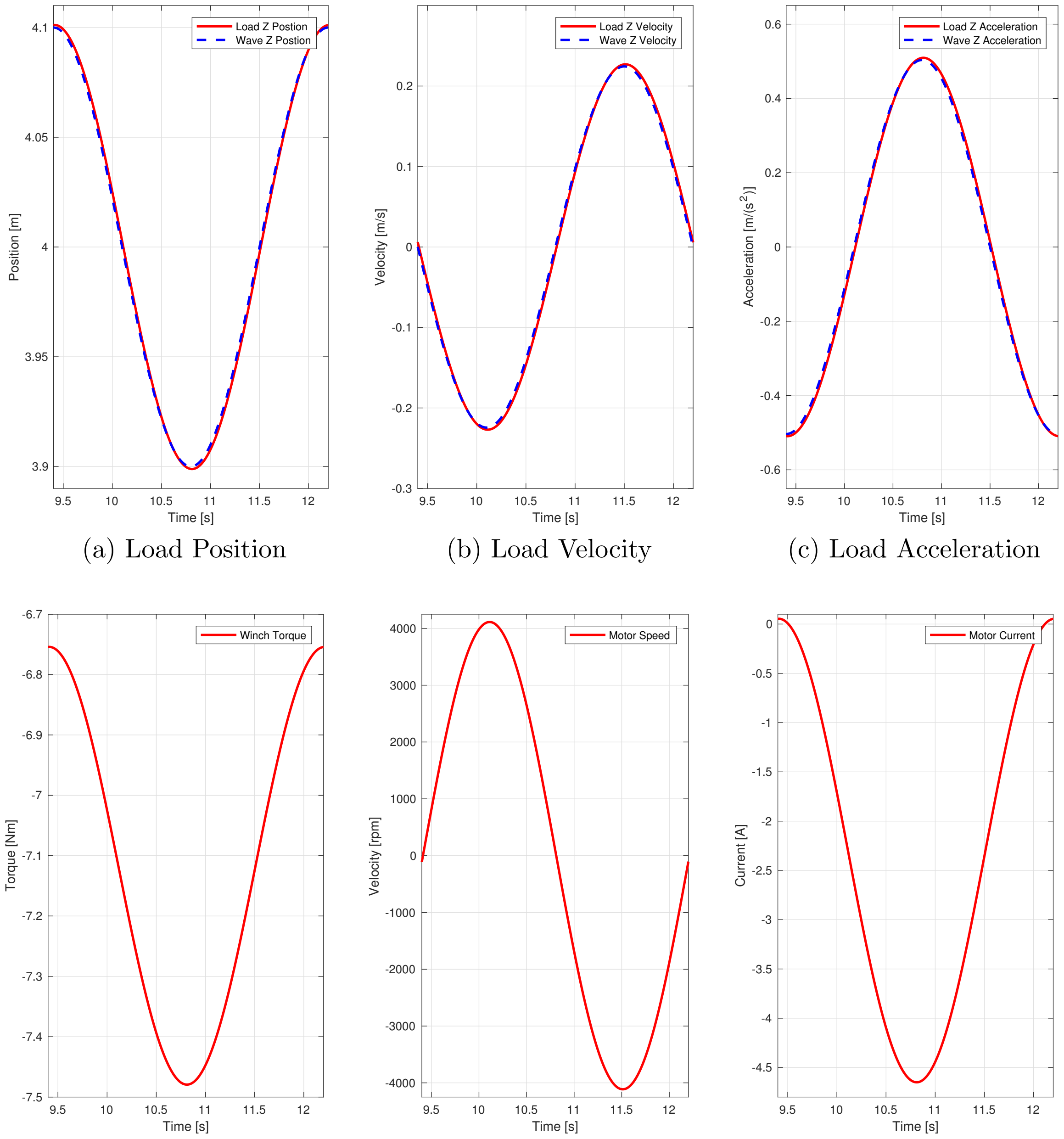


(a) Crane Tip Position (b) Crane Tip Velocity (c) Crane Tip Acceleration

(d) Torque at Slewing Ring (e) Motor Rotational Velocity (f) Motor Current Usage

Figure 5.52: Crane Tip Following Pendulum Horizontal Position

Figure 5.53 shows the final test results obtained when the motor and gearbox for the winch were selected. The Z-positon of the load suspended in the wire from the crane tip is set to follow a cosine wave related to the scaled sea condition, see equation 5.7. The test is done with 150m of 2mm steel wire stored at the winch. The crane is configured with the long lower arm, and both arms are placed in their upper positions, $\theta_2 = 87°$ and $\theta_3 = -28°$.



(a) Load Position

(b) Load Velocity

(c) Load Acceleration

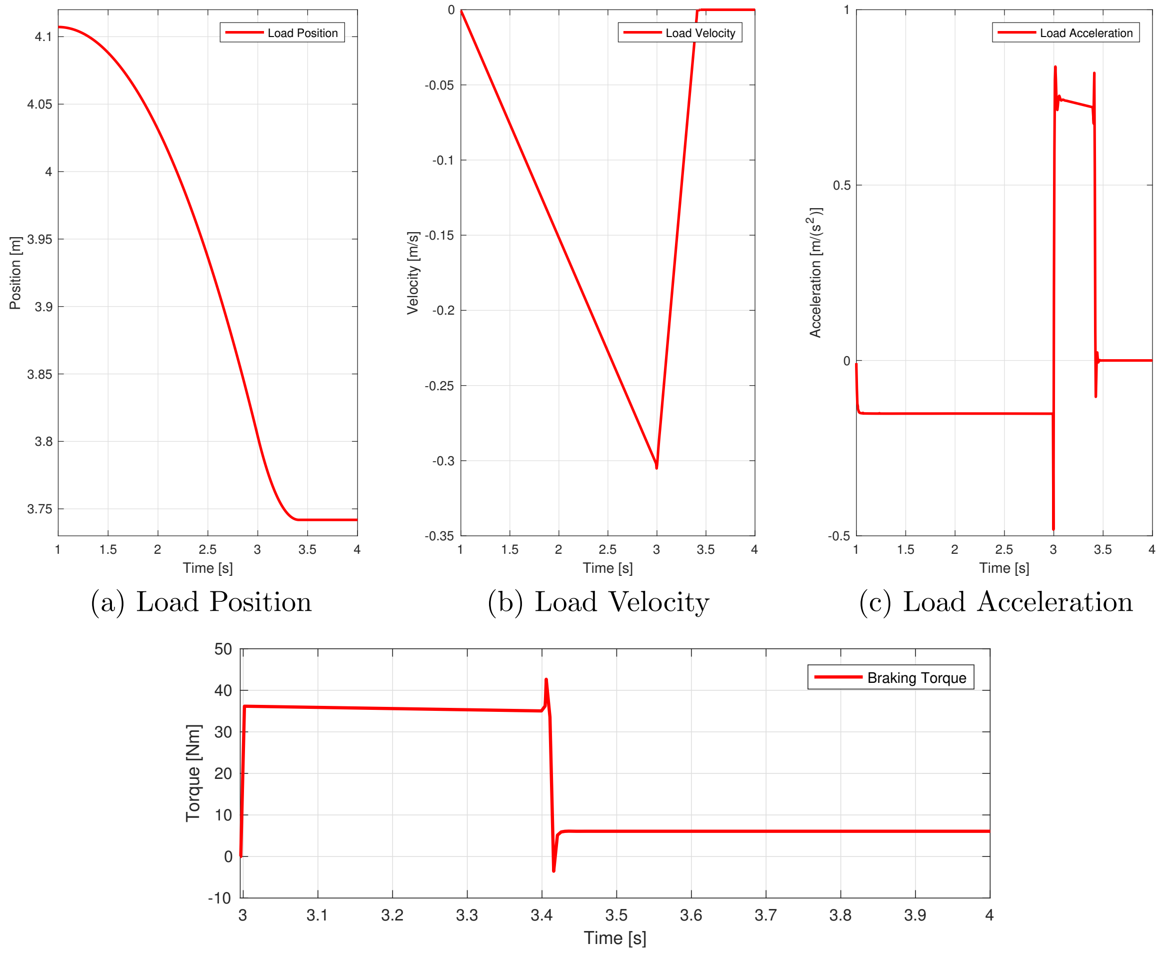
(d) Torque at Reel

(e) Motor Rotational Velocity

(f) Motor Current Usage

Figure 5.53: Load Following Wave Z position

Figure 5.54 shows the final test results for the brake. The test simulates a motor failure at t=1s. The load is then allowed to fall for 2 seconds before the brake is engaged at t=3s. The test is done with 150m of 2mm steel wire on the reel.



(a) Load Position

(b) Load Velocity

(c) Load Acceleration

(d) Braking Torque at Reel

Figure 5.54: Brake Test

5.7 Final Crane Design Overview

A final presentation of the crane design for both the crane configuration with the short and long lower arm is presented in figure 5.55. The figure is showing the complete 3D model for both configurations. The production and assembly of the crane is done with basis in this 3D model which is used to extract production drawings.

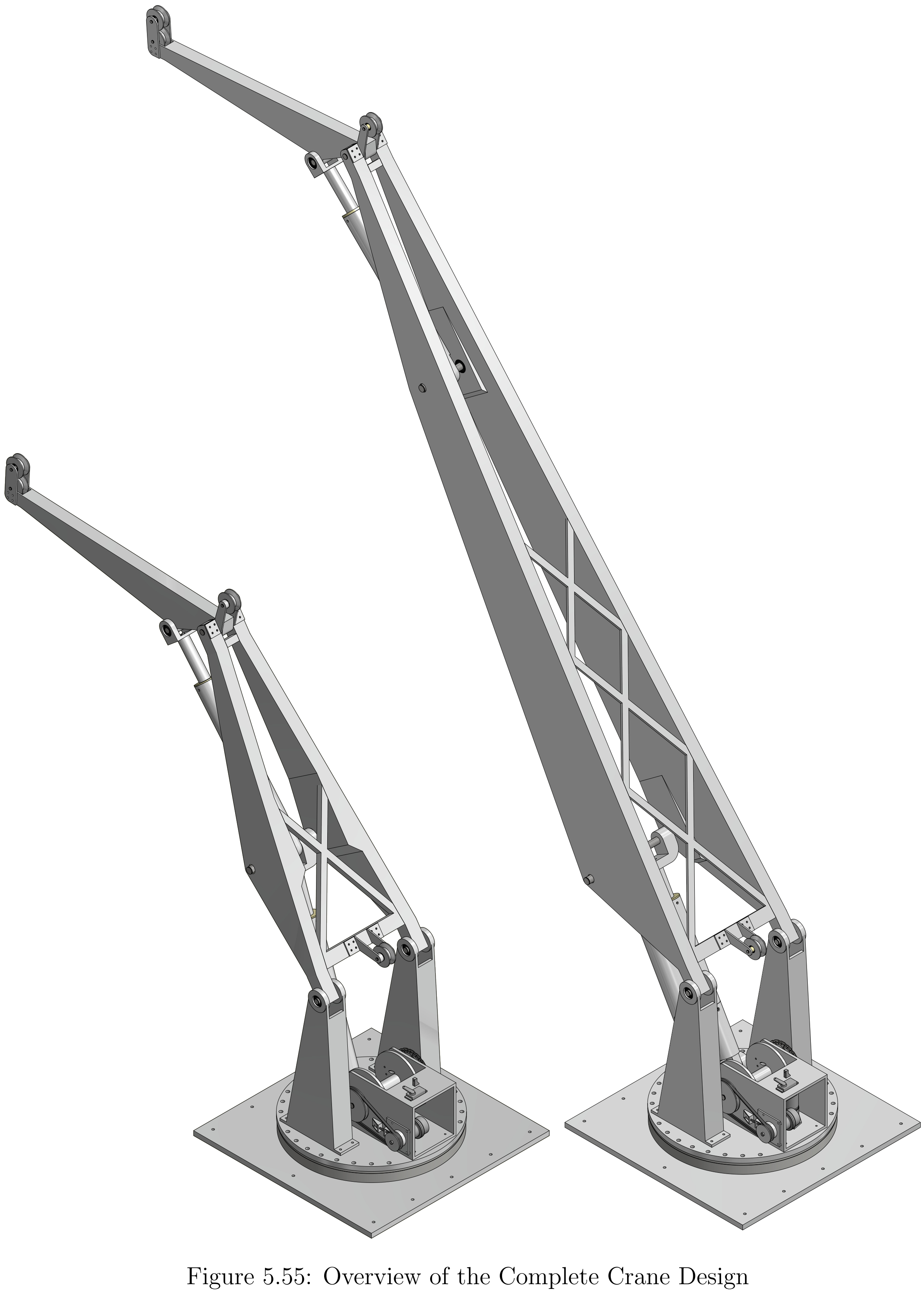


Figure 5.55: Overview of the Complete Crane Design

Chapter 6

Lab Setup

The crane as presented in section 5.7 is in production at the time this thesis is written, a complete lab setup can therefore not be presented at this time. The production and assembly of the short lower arm, upper arm, base (excluding the winch assembly), single lower actuator and upper actuator has been completed, thus allowing a lab setup for the crane configured with these parts only.

6.1 Equipment

In order to make the crane lab work in the best possible way several different components will have to work together. Below is a short presentation of the different equipment used in the lab.

6.1.1 Crane

The crane is of course the most central part in the lab. The crane is installed in the lab on a heavy concrete table, see figure 6.1 for an illustration of the crane installed in the lab. The concrete table which the crane is mounted on is standing on rubber feet. Testing of the crane show that the rubber feet is allowing the table to move/vibrate during operation of the crane. The movement/vibrations in the table may interfere somewhat with the test results from the lab.

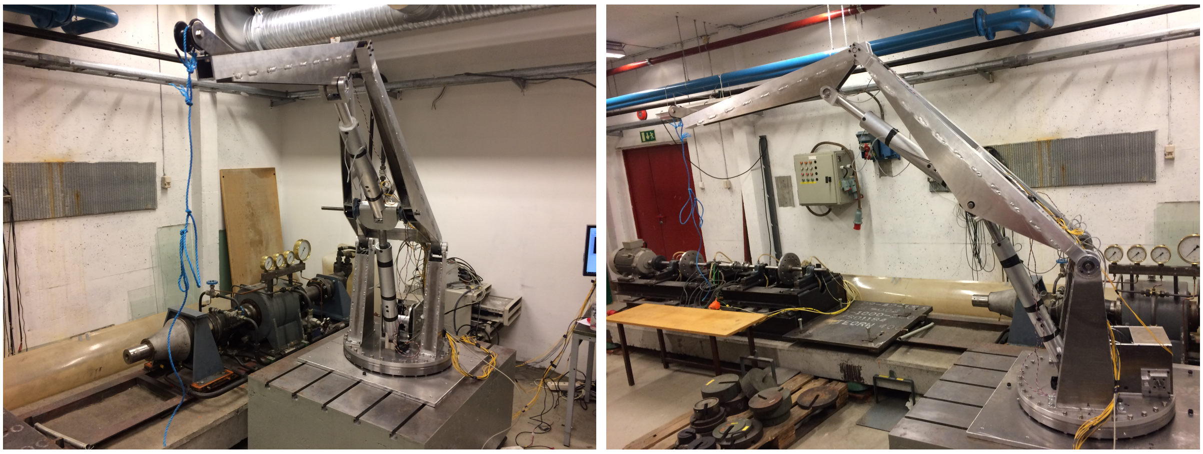


Figure 6.1: Crane as Installed In Lab

6.1.2 Sensors

In order to control and record the crane movement a selection of sensors have been implemented. Each motor has an encoder attached which can be used to read the speed of the motor in real time. As the three motors that currently is mounted to the crane are of the same type, the same encoder is used for all. This is the HEDS 5540, se also table 5.19 and MaxsonHEDS5540 (2016) for more information.

Both actuators have been equipped with proximity sensors which is set up to register when the actuator is at the upper and lower position. These will be used to program a stop functionality to control that the actuator elongation does not exceed the upper or lower limit of the trapezoidal screw travel. This functionality is similar to what has been implemented in the simulation model. As the sensors need to be placed with some safety margin to the end of the trapezoidal screw the angular range of the crane arms have been reduced slightly compared to what is presented in figure 5.31. The final angular range of the crane is as presented in table 6.1

Table 6.1: Final Angular Range of Crane Arms

Parameter	Value
θ_2	$[\,30.2°,\ 86°\,]$
θ_3	$[\,-90.5°,\ -24.4°\,]$

The sensors used is an inductive sensor made by Contrinex and has the model number DW-AD-623-04, se also Contrinex (2016) for more details and figure 6.2 for an illustration of how the sensors is mounted to the actuators.

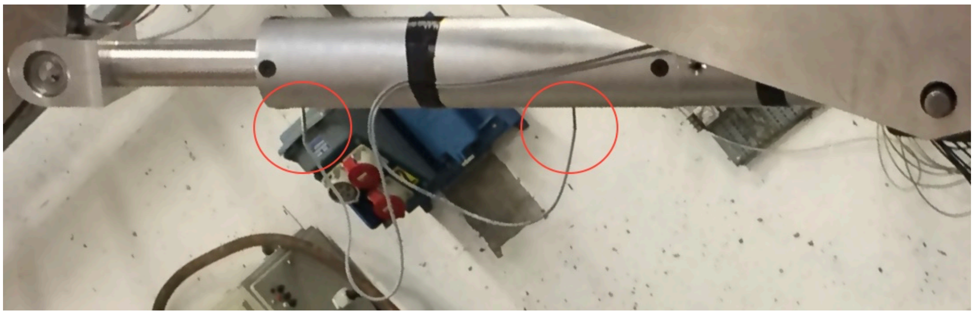


Figure 6.2: Proximity Sensors Installed in Actuator

Three gyroscopes have been mounted on the crane in order to take readings of the angular velocities of the lower and upper arm, as well as the base rotation. The gyroscopes are from Analog Devices and have model number ADXRS646, see also AnalogDevices (2016) for more information on the gyroscopes. The gyroscopes are connected to an ad-converter from National Instruments with model number NI USB-6210, see also National-Instruments (2016) for more information on the ad-converter.

6.1.3 Controllers

Two types of controllers is used in order to control the motion of the crane.

The main controller is the Bachman M1 controller. The controller consist of two parts: The MH212/S processor module, and the universal Input/Output Module GIO212. See Bachmann (2016) for more information on the controllers. A program is written for the control system which runs on the processor module (MH212/S), see appendix F. GIO212 is connected to the processor module and routes the input and output from the program to the desired components.

There are no available module for the Bachman M1 control system that will be able to control the DC motors directly. Three external motor controllers have therefore been used. This is the ESCON Module 50/5 4-Q Servocontroller, see also ESCON (2016) for more information. This controller is a speed/torque controller made for the selected motors.

The configuration of the controllers is included in appendix C, which show a complete wiring diagram, and descriptions of the connections.

6.1.4 Joysticks

Two two-axis joysticks is used to interface the crane lab operator to the crane control system. The two joysticks provide a total of 4 channels, corresponding to the four degrees of freedom for the complete crane including the winch. Only three of the channels have been used in this setup, as the winch is not completed yet. The joysticks are based around 5K-potentiometers, and is set up to give an analog signal $(0 - 10 \ \ [V])$ corresponding to the position of the joystick. The joysticks are from robotchop.com and have the model number RB-Sct-283, see also RobotShop (2016) for more information.

6.1.5 PC

A pc is necessary in order to communicate with the different controllers and hardware installed in the lab. The PC is connected to the Bachmann controller through a crossed ethernet cable, and runs the Bachmann SolutionCenter software in order to access the Bachman controller. The ESCON controller is accessed through a usb connection and with the use of ESCON studio software. The ad-converter from National Instruments is also connected to the PC with the use of a usb cable, and the software Lab View is used to access and record the data from the gyroscopes.

6.1.6 Power Supplies

Most of the lab equipment need DC power at a specific voltage. In total three different power supplies have been used in order to supply the power needed for the equipment.

The three ESCON motor controllers use 36V, and up to 7A each. Note that the ampere consumption can be significantly higher in short bursts. These bursts happen when the motors start turning, or change direction. A total of three 36V, 8A power supplies have been connected in parallel with the use of diodes to make one 36V, 24A power supply. The power supplies are then connected to three 22000 μF condensators which enables the power supplies to take the high power bursts. The reason for this setup is because it was needed to use equipment that was already purchased, and the 36V, 8A power supplies where not able to efficiently drive one motor each.

The proximity sensors and the joysticks use 10V power supplied by a basic 2.5 A power suply.

The gyroscopes are supplied with power from the ad-converter which takes power from the USB connection.

The Bachmann controller use 24V power and is supplied by a basic 5A power supply.

6.2 Controller Setup

A basic control system for the crane has been developed and implemented in the lab. The control system is based around the ESCON Module 50/5 4-Q Servocontroller and the Bachman M1 controller. The ESCON controller is set up as a speed controller, with stop and enable function. A control program, written by the author, (appendix F) is implemented into the Bachman Controller. The control program takes inputs form the user though the joysticks, and from the crane though the proximity sensors and sends out three signals to each of the ESCON motor controllers. Two of these signals are digital, enable and stop, and one are analog ($\pm 0 - 10$V), which is the speed set value. The speed set value is proportional with the input from the joystick, and the ESCON controller will then set and control the speed of the motor with reference to the speed set value. This setup is considered similar to the control setup used in the simulation model presented in section 4.3. The program is set up so that it is not possible to extend or retract the actuators beyond its limits, this is done by enabling the stop function of the ESCON controllers when a signal is given by the proximity sensors. The complete wiring of the controllers is included in appendix C

6.3 Initial Testing in the Lab

The Initial testing of the crane installed in the lab show that the crane is performing as intended, and that the control system is enabling accurate control of both crane arms and the base rotation. The performance of the crane with different loads attached has studied and shows that the crane performance is not decreasing much with loads up to approximately 20kg. The highest load that the crane is observed to lift is 26kg. During the testing of the crane there have not been observed any deformations or deflections of the crane structure. There are however some free travel in the base rotation which should be considered when designing future control systems. There are also some sideways (Y-direction) free travel in the connections between the arms, the lower arm and the base, and the actuators and the arms. This can be improved by inserting bushings with correct size in the joints. The parameters for the friction models of the trapezoidal screw and the slewing ring where selected in the design face with high uncertainty. Tests have therefore been performed to compare the actual power usage of the motors installed, and the power usage of the motors in the simulation model. The friction parameters are then adjusted so that the power usage is matching. The results of the adjustment of the friction parameters is presented in the next section.

6.3.1 Adjustment of Friction Parameters

Recall from chapter 4 that the friction model for both the trapezoidal screw and slewing ring where modeled as a combination of static, colomb and viscous friction. The comparison between the simulation model and the real crane is however showing that the friction is best modeled as colomb friction only. The friction model could therfore have been reduced to a model as shown i equation 6.1 from Pedersen and Engja (2010). The original friction model is however still used, as it will by setting $\mu_s = \mu_c$ and $\mu_v = 0$ give the same results as 6.1

$$F_f = \mu F_N \tanh(sv) \tag{6.1}$$

The final friction parameters for the trapezoidal screws is set as shown in table 6.2 and illustrated in figure 6.3, recall from section 4.2.2 that the value for F_N in the trapezoidal screw friction model is not a parameter, but a variable. The value of F_N presented in table 6.2 is set as an example.

Table 6.2: Trapezoidal Screw Final Friction Parameters

Parameter	Value
F_N	1750
s	100
μ_s	0.0023
μ_c	0.0023
μ_v	0

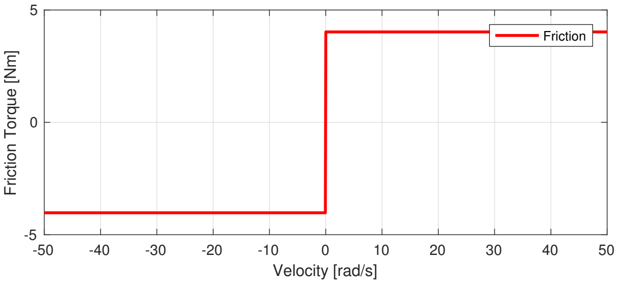


Figure 6.3: Trapezoidal Screw Final Friction

The final friction parameters for the slewing ring is set as shown in table 6.3 and illustrated in figure 6.4.

Table 6.3: Slewing Ring Final Friction Parameters

Parameter	Value
F_N	1000
s	1000
μ_s	0.06
μ_c	0.06
μ_v	0

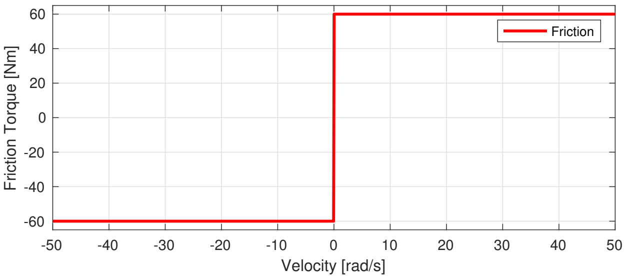


Figure 6.4: Slewing Ring Final Friction

Chapter 7

Final Results

A series of tests have been performed with the lab setup presented in chapter 6. The focus of the tests have been to evaluate the accuracy of the simulation model compared to the real crane model. The performed tests will also show the capability of the crane as it is installed in the lab at the time of the writing of this theis. The results presented in this chapter will only cover the movement of the upper arm, short lower arm and base. The reason for this is that this is the only parts of the crane that is produced and assembled at the time of the writing of this thesis.

7.1 Lab Tests Setup

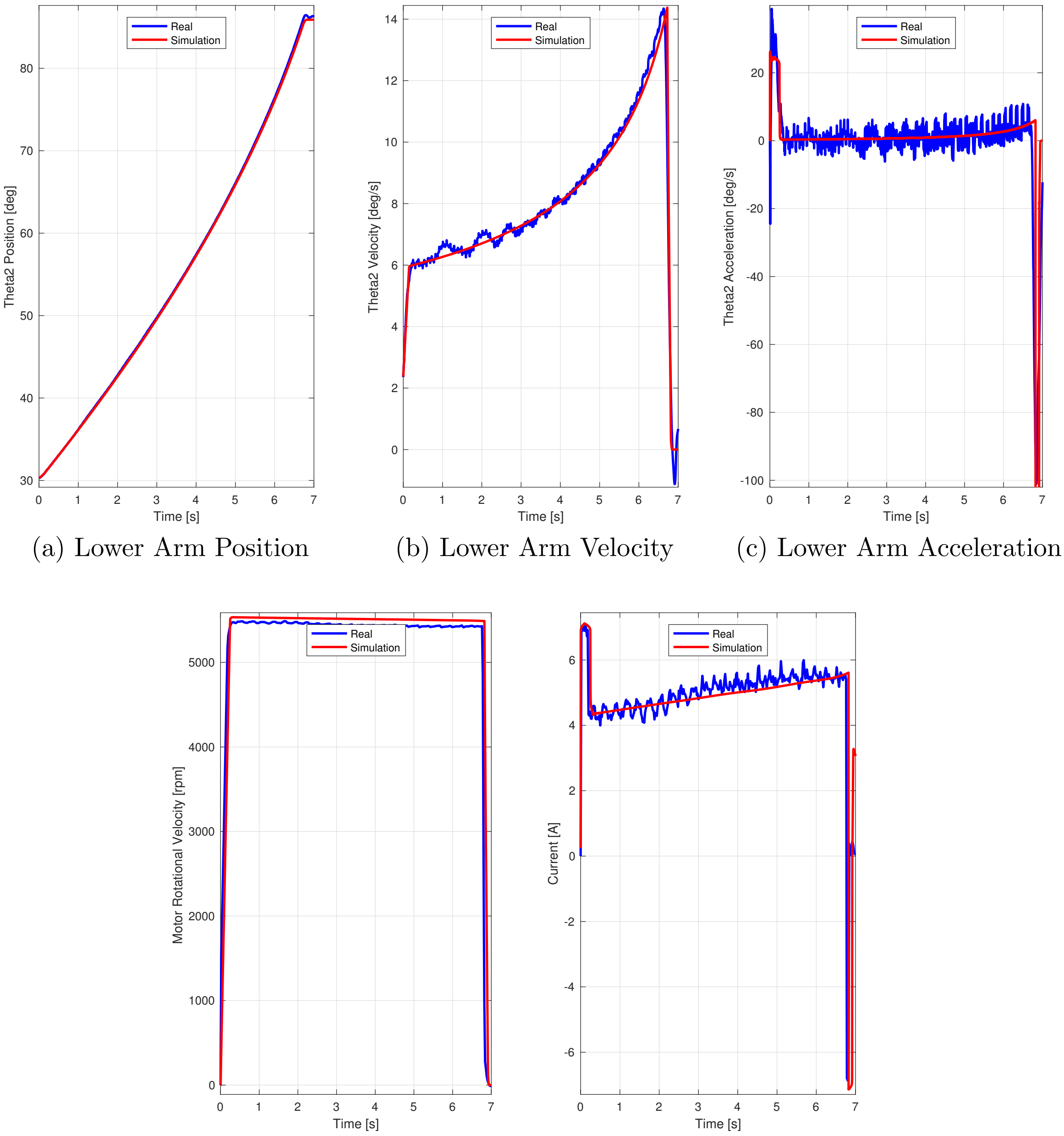
The tests in the lab is performed by setting a step or pulse signal to the relevant motor, and recording the response from the gyroscopes and motor controllers. A total of three tests is presented in this chapter, one test for each degree of freedom of the crane. It should be noted that the recorded data from both the gyroscopes and the motor controllers are affected by noice which may affect the results from the tests. The crane is loaded with 16kg hanging approximately 1m from the crane tip in all the tests presented in this chapter. A 5mm rope made from an unknown plastic fibre has been used to connect the load to the crane.

7.2 Simulation Model Setup

The simulation model is configured with the parameters presented in chapter 5, this means that there can be some differences in the parameters used in the simulation model and the parameters of the real crane as some inaccuracies during production have to be accounted for. The only parameters that have been altered compared to chapter 5 are the friction parameters which is set to the values presented in section 6.3.1. The 5mm plastic fibre rope with unknown properties is represented by a 2mm steel wire in the simulation model.

7.3 Test 1 - Lifting of Lower Arm

This testes is performed by sending a step signal (from 0% to 100% power at time = 0) at the motor controller for the lower actuator of both the real crane and the simulation model. This will run the motor in the lower actuator at full power, thus illustrating the maximal capacity of the actuator. The actuator is allowed to run until the upper stop swich is activated. The stating condition of the crane is $\theta_1 = 0°$, $\theta_2 = 30.2°$ and $\theta_3 = -90.5°$. The result of the test is presented in figure 7.1



(a) Lower Arm Position

(b) Lower Arm Velocity

(c) Lower Arm Acceleration

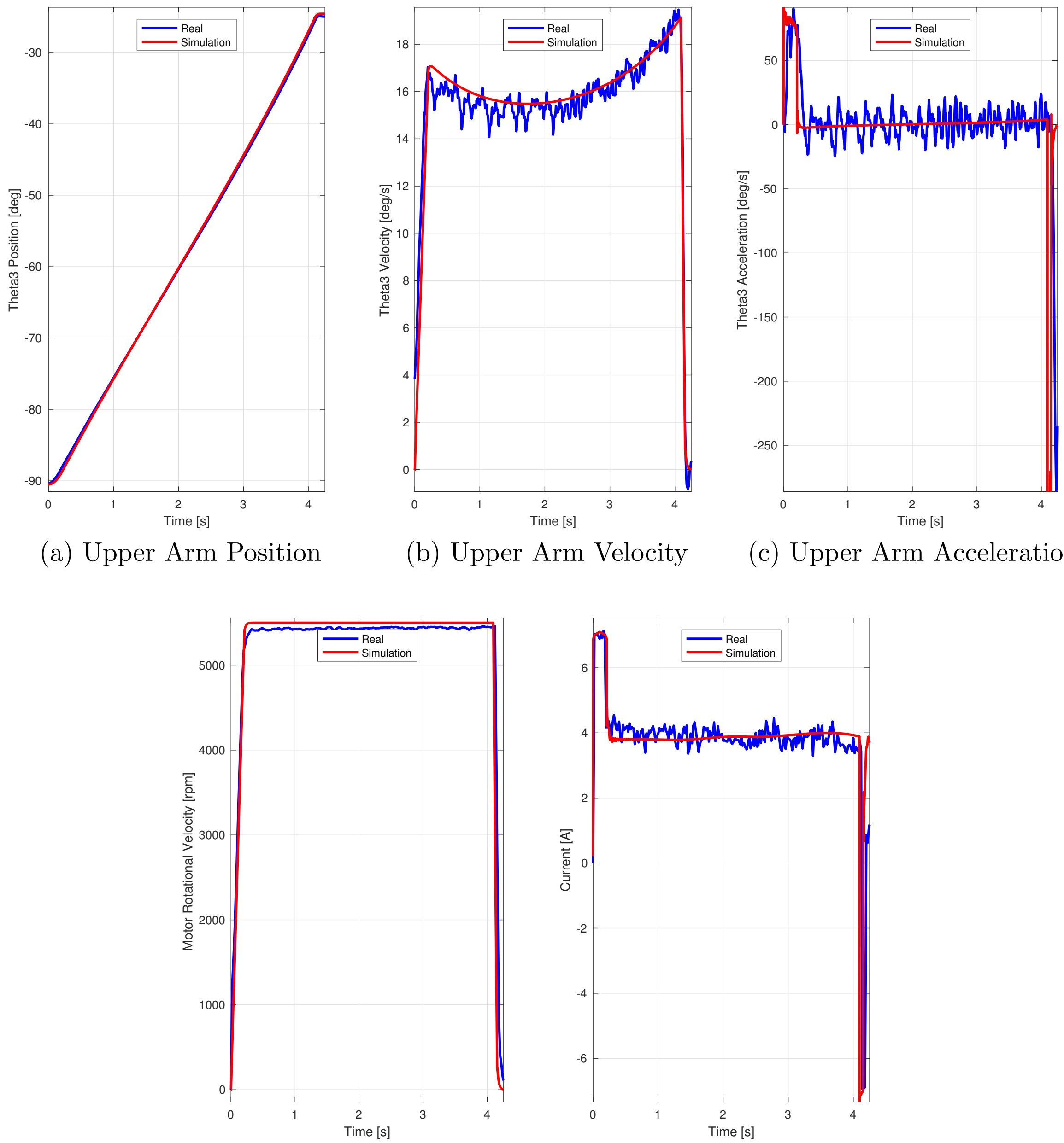
(d) Motor Rotational Velocity

(e) Motor Current Usage

Figure 7.1: Test 1 - Lifting of Lower Arm

7.4 Test 2 - Lifting of Upper Arm

This testes is performed by sending a step signal (from 0% to 100% power at time = 0s) at the motor controller for the upper actuator of both the real crane and the simulation model . This will run the motor in the upper actuator at full power, thus illustrating the maximal capacity of the actuator. The actuator is allowed to run until the upper stop swich is activated. The stating condition of the crane is $\theta_1 = 0°$, $\theta_2 = 30.2°$ and $\theta_3 = -90.5°$. The result if the test is presented in figure 7.2



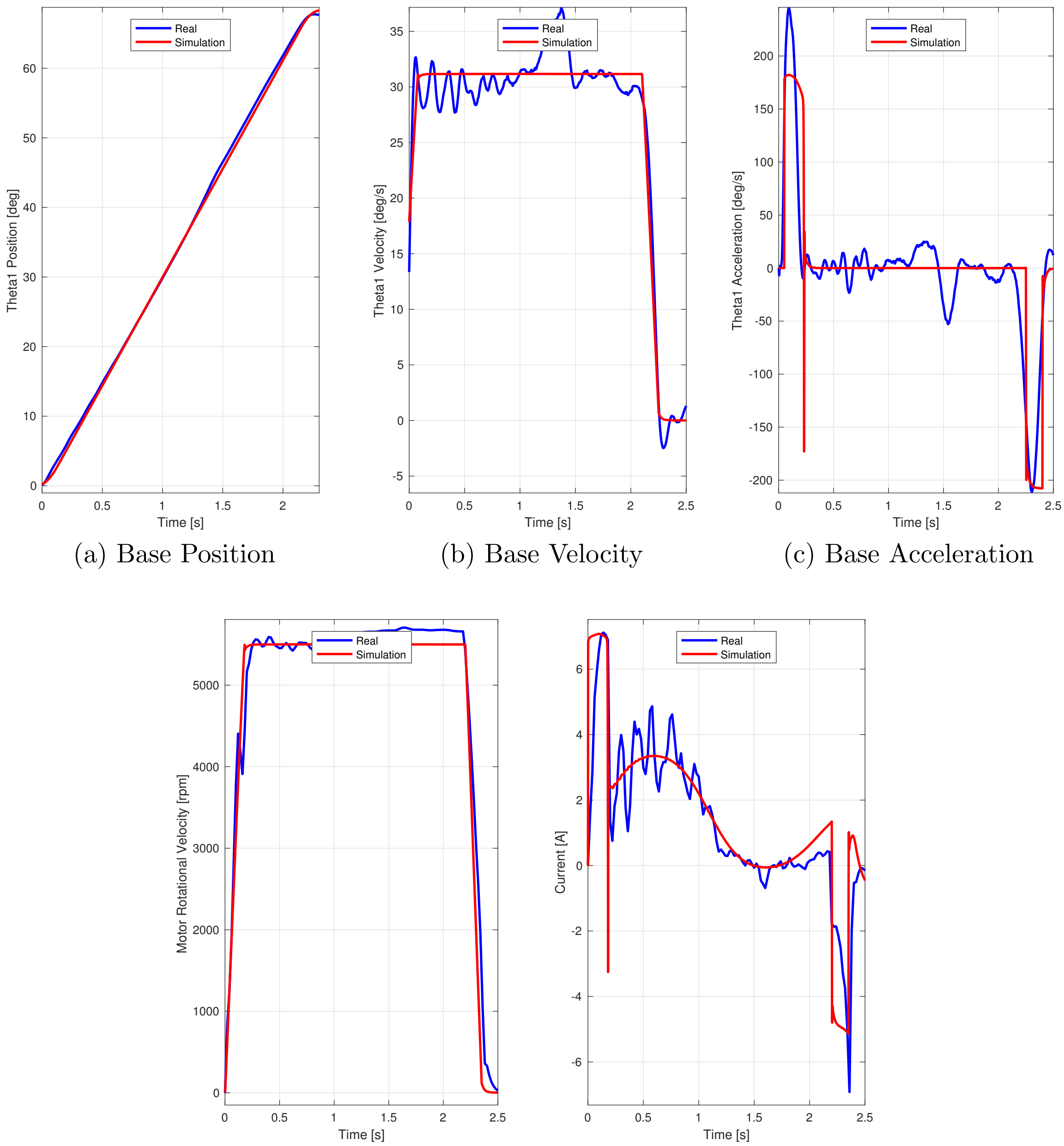
(a) Upper Arm Position (b) Upper Arm Velocity (c) Upper Arm Acceleration

(d) Motor Rotational Velocity (e) Motor Current Usage

Figure 7.2: Test2 - Lifting of Upper Arm

7.5 Test 3 - Rotating of base

This testes if performed by sending a pulse signal (from 0% to 100% power at time = 0s and 100% to 0% power at time=2.1s) at the motor controller for the base rotation of both the real crane and the simulation model . This will run the motor controlling the base rotation at full power, thus illustrating the maximal capacity of the base rotation. The stating condition of the crane is $\theta_1 = 0°$, $\theta_2 = 45°$ and $\theta_3 = -45°$. The result if the test is presented in figure 7.3



(a) Base Position

(b) Base Velocity

(c) Base Acceleration

(d) Motor Rotational Velocity

(e) Motor Current Usage

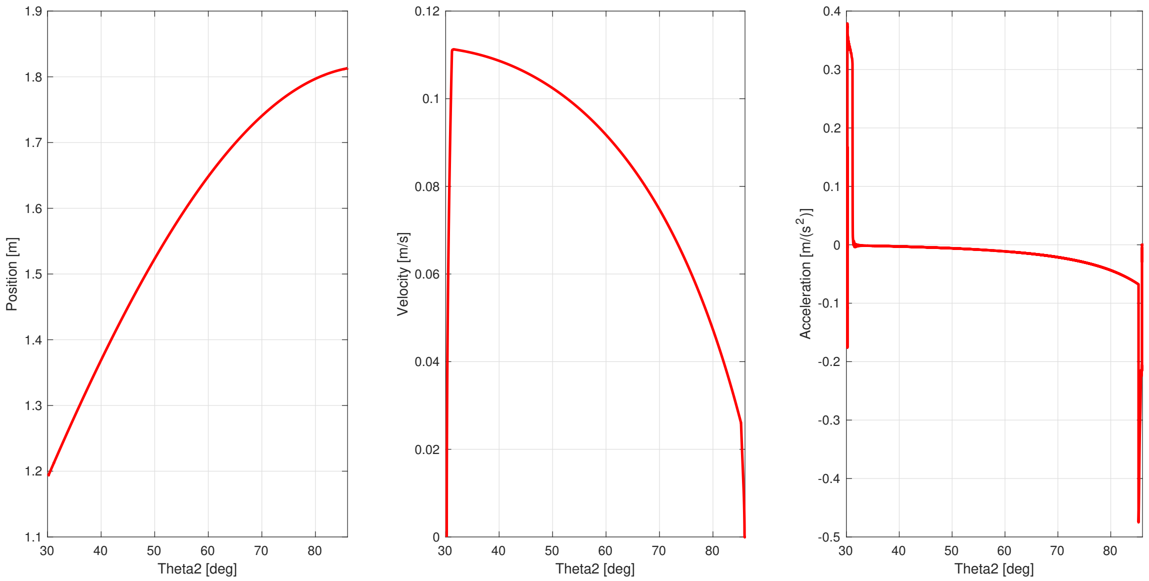
Figure 7.3: Test 3 - Base Rotation

7.6 Review of the Results

The comparative results clearly show that the simulation model is an accurate representation of the real crane when one accounts for the noice in the data from the lab tests, and the possible errors in the parameters used in the simulation model. This can be as considered a verification that the modeling technique used in this thesis produces a impressively accurate model of a system of rigid bodies in spatial motion.

7.6.1 Lower Arm

The results concerning the lower arm show that the design of the actuator geometry and the selection of the actuator components have resulted in a well performing actuation of the lower crane arm. The crane arm has a wide angular range and is able to move from the lower position to the upper position in under 7 seconds. The responsiveness is high with an angular acceleration of over 20 [deg/s^2]. This should enable the crane to be used in a wide variety of scientific testing. The position, velocity and acceleration of the tip of the lower arm in Z-direction has also been studied. The reason for that is that one of the main fields of research that the crane lab is intended for is to test control systems for heave compensating, and it is therefore interesting to show the capability of the crane for such testing. The simulation model is used to plot position, velocity and acceleration in Z-direction as functions of the angular position of the lower arm, the result is shown in figure 7.4. The upper arm is set at a position of $\theta_3 = -57°$ which is the middle position of the angular movement of the upper arm. The crane is also loaded with 16kg in the test, and the lower actuator motor is running at full power. The position of the tip of the lower arm is plotted relative to the top of the upper base plate.

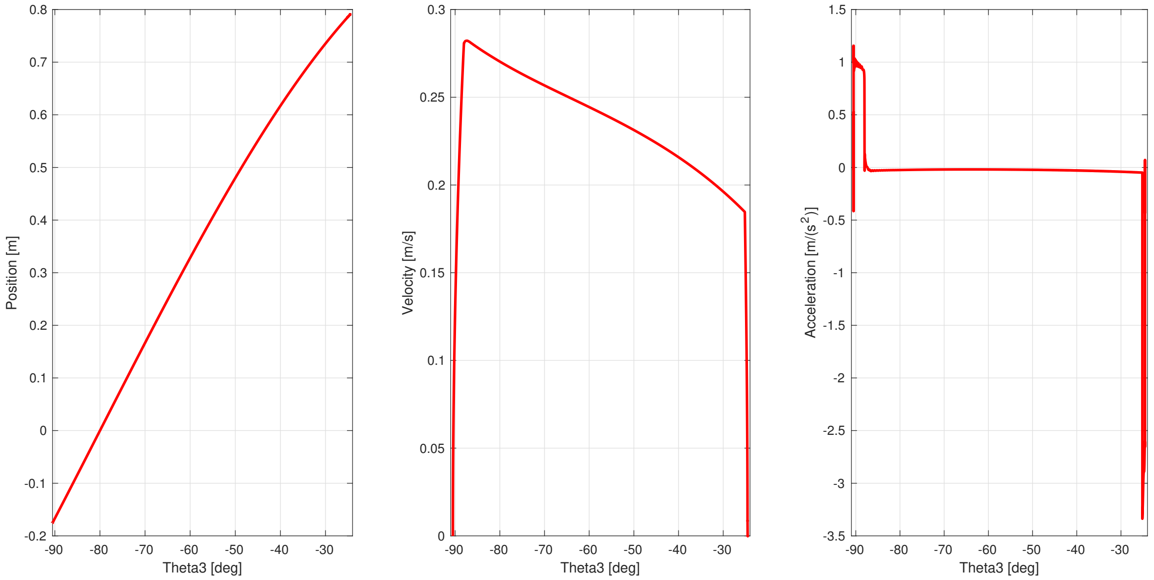


(a) Possiton (b) Velocity (c) Acceleration

Figure 7.4: Tip of Short Lower Arm Z- Position, Velocity, and Acceleration

7.6.2 Upper Arm

The results concerning the upper arm show that the design of the actuator geometry and the selection of the actuator components have resulted in a well performing actuation of the upper crane arm. The crane arm has a wide angular range and is able to move from the lower position to the upper position in under 4.5 seconds. The responsiveness is very high with an angular acceleration of over 70 [deg/s^2]. This should enable the crane to be used in a wide variety of scientific testing. The position, velocity and acceleration of the tip of the upper arm in Z-direction has also been studied. The reason for that is that one of the main fields of research that the crane lab is intended for is to test control systems for heave compensating, and it is therefore interesting to show the capability of the crane for such testing. The simulation model is used to plot position, velocity and acceleration in Z-direction as functions of the angular position of the upper arm, the result is shown in figure 7.4. The lower arm is set at a position of $\theta_2 = 80°$ which is the most natural position of the upper arm during heave compensation. The crane is also loaded with 16kg in the test, and the upper actuator motor is running at full power. The position of the tip of the upper arm is plotted relative to the connection between the upper and lower arm.



(a) Possiton (b) Velocity (c) Acceleration

Figure 7.5: Tip of Upper Arm Z-Position, Velocity, and Acceleration

Chapter 8

Conclusion and Further Work

This thesis is presenting the results and processes of designing a scaled down crane model that is intended for studying of crane behaviour and control systems.

One of the most central parts of the work done and presented in this thesis is the development of a simulation model that is able to predict the behaviour of several different configurations of the scaled down crane model. The simulation model is built up partly with mathematics that in an exact manner represents real world behaviour, and partly by models that is estimating the real behaviour. It has of course been of the highest interest to be able to model the crane as accurately as possible, thus basing the model in as great extent as possible on exact physical models. Exact models can often be complex and computational intensive to solve during simulation and it has therefore been a focus to use an approach that will yield a compact and simulation friendly model.

The kinematics of both the crane arms, actuators, and the base of the crane is for the scaled down crane model considered as rigid bodies in spatial motion. This would have been a simplification on a full scale crane as deflections would have to be accounted for. The framework developed by Rokseth (2014), which is based on a combination of Lagrange-Hamiltionian theory and regular bond graph models, is proven to be an effective method for modeling of such systems and is therefore used as a basis to create an exact kinematic model of the rigid parts of the crane. The forces acting on the crane such as gravity and the actuator forces is also implemented as exact physical models by transforming them to generalized forces and interfacing them to the kinematic model. Some simplifications compared to real world physics have had to be made when modeling the different subsystems such as the internals of the actuators and the winch and wire submodel. The simplifications made when modeling the the actuators is mostly done with regard to friction and other forms of energy loss. The same is the case for the winch model. The wire model is also a simplified model which considers the wire as a chain of mass-spring-damper systems linked together to form a chain of ”wire-elements”. The interaction between the winch and the wire is also simplified by making the winch extend and retract these ”wire-elements”.

The usefulness of the simulation model during the design process of the crane has been well documented throughout this thesis. The simulation model has been used to study how different design parameters affect the crane performance and verify that different components such as the motors are able to deliver the performance that is desired. Some safety factors have of course been applied as the accuracy of the simulation model had not been verified at the design stage.

The comparative tests between the simulation model and the real crane demonstrates the accuracy of the simulation model. The crane behaviour predicted by the simulation model is more or less identical to the behaviour of the real crane when the friction parameters are adjusted. This concludes that the methods used for modeling the exact physics of the crane is indeed exact, and the only deviations can be explained by the estimation models for the friction in the actuators and slewing ring, which are also performing accurately when the correct parameters are applied. This means that the performance predictions during the design phase can be considered correct, and the crane will be able to perform according to the design requirements.

The other aspects with regard to the design of the crane is also considered a success as the structure of the crane show no deformations or deflections during testing even with loads that exceed the design load. The modularity of the crane is also proven by using as few different components as possible and realising a design that is assembled and dissembled with ease. This is making the crane an optimal basis for further development and modification.

The control systems developed and partly tested on the real crane in this project and presented in this thesis are fairly basic and only consisting of controllers that control the speed of the motors and the initiation of the winch brake. Even though the controllers are basic and only controlling one separate motor each the level of accuracy that is possible to achieve with this type of control is impressive. The controllers will therefore provide a solid basis for further development of more sophisticated control of the crane.

The project of developing the crane lab is still ongoing at the point of writing this thesis, and the production of the long crane arm and dual lower actuator as well as the winch assembly will need managing in order for the crane to achieve the predicted performance. The controllers installed in the lab should also be installed in a more permanent manner then they are at the time of writing this thesis.

Several different approaches can be taken in order to develop the crane lab further, and the interest for the crane lab form other projects have been increasing during the completion of this thesis. Testing and development of control systems for different applications have already been scheduled for the crane, and it is expected that the crane lab will be used as a testing facility for several future research projects.

It is suggested that the simulation model is extended in order to simulate the performance of the crane when the crane is placed on a floating object, e.g. a vessel. This is proposed done as explained in Rokseth et al. (2016). This includes extending the kinematic model to account for the kinematics of the floating vessel and the interactions with the floating vessel and the crane. The system will therefore be extended to a total of nine degrees of freedom. The modeling methodology used to create the kinematic model in Rokseth et al. (2016) is similar to the method that is presented in this thesis. The subsystems acing on the crane can be adapted directly from the simulation model presented in this thesis, but the interactions between the load hanging from the crane and the water should be included ass explained in Rokseth et al. (2016). The crane design is assumed, with reference to the tests considering the wave conditions, to be ready for use in the ocean basin, but a full simulation model of the floating crane setup will bee highly useful in order to develop sophisticated controllers controlling the load hanging form the crane tip with reference to the motion of the vessel.

Bibliography

AnalogDevices (2016). *Datasheet for Gyroscope ADXRS646.* Accsessed: 23.11.16, from: http://www.analog.com/media/en/technical-documentation/data-sheets/ADXRS646.pdf.

Appleton (2016). *Marine Cranes Product Range.* Accsessed: 21.09.16, from: http://www.appletonmarine.com/marine-products/cranes.html.

Autodesk (2016). *Inventor Manual.* Accsessed: 13.10.16, from: http://docs.autodesk.com/INVPRO/2010/ENU/Autodesk%20Inventor%202010%20Help/index.html.

AvidBB5 (2016). *Disk Brake Product Page.* Accsessed: 12.10.16, from: https://bikeshop.no/sykkel/bremser/bremser/skivebremser/avid/avid-bb5-skivebremskalipper-mekanisk-mtb-160mm-m-adaptere-00.5016.166.060-p0000023216.

Bachmann (2016). *Controller System Product Page.* Accsessed: 01.12.16, from: http://www.bachmann.info/en/products/controller-system/.

Bouyer, J. and Fillon, M. (2011). Experimental measurement of the friction torque on hydrodynamic plain journal bearings during start-up. *Tribology International*, 44(7-8):772–781.

Conti (2016). *Trapezodial Screw Catalog.* Accsessed: 14.11.16, from: http://www.eiemaskin.no/media/1379/en-catalogue-2012.pdf.

Contrinex (2016). *Inductive Sensor Product Page.* . Accsessed: 09.12.16, from: https://www.elfadistrelec.no/no/inductive-sensor-mm-pnp-sluttekontakt-kabel-pvc-10-30-vdc-25-70-contrinex-dw-ad-623
p/13757501?q=*&filter_Category3=Induktive+sensorer&filter_Manufacturer=Contrinex&filter_Category4=Induktive+sensorer&filter_Buyable=1&page=3&origPageSize=50&simi=97.02 (Accsessed: 09.12.16).

Controllab (2016). *20-Sim Manual.* Accsessed: 26.10.16, from: http://www.20sim.com/webhelp/.

ESCON (2016). *ESCON 50/5HR Motor Controller Datasheet.* Accsessed: 26.10.16, from: http://www.maxonmotor.com/medias/sys_master/root/8818448859166/438725-ESCON-Module-50-5-Hardware-Reference-En.pdf (Accessed: 01.12.16).

Gyberg, F. (2015). *Development of Physical Crane Model for use in Student Lab.* (Project Thesis, NTNU). Trondheim: NTNU.

Härkegård, G. (2004). *Dimensjonering av Maskindeler.* Vigmostad and Bjørke AS, 3th edition.

Heila (2016). *Marine Cranes Product Range.* Accsessed: 21.09.16, from: `http://www.heila.com/products`.

Jackson, P. L. (2010). *Getting Design Right, A systems Approach.* Taylor and Francis Group, LLC.

Karnopp, D. C., Margolis, D. L., and Rosenberg, R. C. (2012). *System Dynamics: Modeling, Simulation, and Control of Mechatronic Systems.* John Wiley and Sons, Inc., 5th edition.

Kossiakoff, A., Sweet, W. N., Seymour, S. J., and Biemer, S. M. (2011). *Systems Engineering Principles and Practice.* John Wiley and Sons, Inc., 2nd edition.

Madenci, E. and Guven, I. (2006). *The Finite Element Mehtod and Applications In Engeneering using Ansys.* Springer Sience and Business Media, LCC.

MaxsonEC60 (2016). *Maxson Motor EC60 DC-Motor Datasheet.* Accsessed: 23.10.16, from: `http://www.maxonmotor.com/medias/sys_master/root/8821068005406/16-304-EN.pdf`.

MaxsonGP52C (2016). *Maxson Motor GP52C Gearbox Datasheet.* Accsessed: 23.10.16, from: `http://www.maxonmotor.com/medias/sys_master/root/8821070069790/16-354-355-EN.pdf` (Accssed 05.12.16).

MaxsonHEDS5540 (2016). *Maxson Motor HEDS5540 Encoder Datasheet.* Accsessed: 23.10.16, from: `http://www.maxonmotor.com/medias/sys_master/root/8821074362398/16-399-400-EN.pdf`.

MaxsonRE50 (2016). *Maxson Motor RE50 DC-Motor Datasheet.* Accsessed: 23.10.16, from: `http://www.maxonmotor.com/medias/sys_master/root/8820831092766/16-187-EN.pdf`.

National-Instruments (2016). *Data Logger USB-6210 Data Sheet.* Accsessed: 17.12.16, from: `http://www.ni.com/datasheet/pdf/en/ds-9`.

Palfinger (2016). *Marine Cranes Product Range.* Accsessed: 21.09.16, from: `https://www.palfinger.com/en/marine/products/marine-cranes`.

Pedersen, E. and Engja, H. (2010). *Mathematical Modelling and Simulations of Physical Systems.* Department of Marine Technology, Norwegian University of Science and Technology.

PowerHD-1235MG (2016). *PowerHD-1235MG Product Page.* Accsessed: 22.11.16, from: `http://www.elefun.no/p/prod.aspx?v=25426`.

Redell, F. (2008). *Understanding and Applying FEA.* KETIV Technologies of California, Inc.

RobotShop (2016). *RB-Sct-283 2 Axis Joystick Data Sheet.* Accsessed: 04.12.16, from: http://www.robotshop.com/media/files/pdf/datasheet-r204bm2.pdf.

Rokseth, B. (2014). *A Bond Graph Approach for Modelling Systems of Rigid Bodies in Spartial Motion.* (Master Thesis, NTNU), Trondheim: NTNU.

Rokseth, B., Skjong, S., and Pedersen, E. (2016). Modeling of generic offshore vessel in crane operations with focus on strong rigid body connections. *IEEE Journal of Oceanic Engineering*, PP(99):1–23.

Rollix (2016). *Rollix Slwing Ring Cataloge.* Accsessed: 24.10.16, from: http://www.rollix.com///GestionFormulaires/bin/GenererFormulaire.asp?NAMEFORM=TelechargementFichierAnglais&PARAM1=http://www.rollix.com/AUTH/Defontaine%20Group/Rollix/CATALOGUE%20PDF//Rollix%20catalogue%20English%20version.pdf.

SKF6004-2Z (2016). *Ball Bearing SKF6004-2Z Product Page.* Accsessed: 15.11.16, from: http://www.skf.com/pk/products/bearings-units-housings/ball-bearings/deep-groove-ball-bearings/single-row-deep-groove-ball-bearings/single-row/index.html?designation=6004-2Z.

SKF6005-2Z (2016). *Ball Bearing SKF6005-2Z Product Page.* Accsessed: 26.10.16, from: http://www.skf.com/my/products/bearings-units-housings/ball-bearings/deep-groove-ball-bearings/single-row-deep-groove-ball-bearings/single-row/index.html?designation=6005-2Z.

Skjong, S. (2014). Modeling hydraulic winch system. *International Conference on Bond Graph Modeling and Simulation - ICBGM'2014.*

Weisstein, E. W. (2016). *"Logistic Equation." MathWorld–A Wolfram Web Resource.* Accsessed: 14.12.16, from: http://mathworld.wolfram.com/LogisticEquation.html.

Appendix A

Electronic Attachments

This appendix gives an overview of the electronic attachments submitted with this thesis. The electronic attachments are submitted as a zip. file containing 7 folders. The 7 folders contains all the models, code scripts and data that have been created during this project. This appendix is structured in accordance with this folder structure.

3D Model

This folder contains the two 3D models that have been created in inventor. The complete assemblies are named "Crane complete.iam" for both the long and the short arm version of the crane.

Bachmann Control Code

This folder contains the complete control code created for the Bachmann controller in order to control the crane in the lab.

CraneDll

This folder contains the complete code used to create the dll. file that functions as an extension to the simulation model.

Maple Scripts

This folder contains the complete Maple script that is used to calculate necessary equations for the simulation model.

Matlab Scripts

This folder contains the Matlab scripts used in this thesis. There are in total 5 different Matlab scripts:

Amplimiter.m: This script is used to calculate parameters for and visualize the current limiting function used in the control system in the simulation model.

Friction.m: This script is used to calculate and visualize the friction models used in the simulation model.

LabCalculations.m: This script is used to set input and run the Matlab function LabPlot.m.

LabPlot.m: This script is used to visualize the test results form the lab and compare the results from the lab with the results in the simulation model.

Plot.m: This script is used to create plots from various data saved in csv. files.

Simulation Model

This folder contains the simulation model including the dll. file. The necessary parts for the 3D simulation is also included, and a short video of the 3D simulation is included as well.

Test Data From Lab

This folder contains all the raw data recorded in the lab. A short description of the different tests that have been performed is also included.

Appendix B

Crane Lab User Manual

This user manual is written for the lab setup that is documented in chapter 6. Future changes to the crane lab will possibly affect the procedures presented in this manual, thus making this manual outdated. It is recommended that everyone that are to use the crane lab has a thorough understanding of how the crane is built and how the crane is functioning. This thesis is written so that it provides the necessary information to gain this understanding. It is also important to understand the equipment used in the crane lab, see chapter 6 to get an overview of the equipment and references for detailed information about the equipment. Several of the steps in the procedures of this user manual involves using the software SolutionCenter from Bachman. This software has a comprehensive user manual that will serve as a reference to the user manual for the crane lab presented here. The SolutionCenter user manual can be accessed through the help menu within the SolutionCenter software.

B.1 General Startup of the Crane Lab

This procedure explains how to start up and operate the crane with the control program that is presented in appendix F.

1. Start up the Pc and log into the user Crane Lab with the password CraneLab.
2. Power up the equipment by plugging in the contacts in the power outlets.
3. Open Bachmann software SoultionCenter.
4. Select Device manager.
5. Select Open Connection in the navigator on the left side. This will, if the Bachmann controller has booted correctly, connect the Bachman controller to the pc.
6. Select Variables View in SoultionCenter (My Devices - M200 - MIO - 5) to view the output and input variables for the Bachmann controller.

7. Select C/C++ view in SoultionCenter. Open the folder ControllerTest in the navigator window, then the source files folder and open the program controll_app.c. This is the control program presented in appendix F.

8. Run the program by selecting: Run - Run - Bachman Software Module. The program is now running on the Bachmann controller.

9. Select the device manager view in SoultionCenter. Scroll down in the variables list and find variable 1. Edit the value of the variable to 1 to enable the Escon motor controllers.

10. The crane can now be controlled by the joysticks.

B.2 Writing a New Control Program

This procedure explains how to write a new control program and implementing it in the Bachman controller.

1. Perform steps 1-5 in the general startup procedure.

2. Select C/C++ view in SoultionCenter.

3. Select: file - new - C Project.

4. Select C Template of the Bachman module in the dialog that appears, then select the Bachman 4.1.2 C Toolchain

5. Give the program a name.

6. Select finish to create the program.

7. A template for the new program is now created, it is placed under ”ProgramName” - Source Files - ”programname”_app.c in the file naviagtor.

8. Open the program to edit it to your specifications.

B.3 Changing the Properties of the Escon Motor Controllers

This procedure explains how to assess and change the properties of the Escon motor controllers.

1. Conect the desired Escon Motor controller with USB cable to the computer.

2. Open the program Escon Studio.

3. Folow the startup wizzard. Do not change anything if you do not have a reason to do so. The default values are correct for the crane configuration presented in this thesis.
4. The controller parameters can now be altered in the controller monitor.
5. Escon Studio software can also with the built in data recorder be used to record data for the controller.

B.4 Connecting Bachman Controller to a New PC

The PC installed in the lab is already set up for the Bachmann controller. However, if a new PC is to be used the following procedure can be used in order to connect with the Bachmann controller.

1. Connect power to the Bachmann controller and let it boot.
2. Connect the computer to the Bachmann controller with a crossed ethernet cable.
3. Make sure the IPv4 address of the computer is: 192.0.1.[1-250]
4. Make sure the IPv4 subnet mask is: 255.255.0.0
5. Make sure that IPv6 is not enabled.
6. Open Bachmann software SoultionCenter.
7. Select Device manager.
8. Select Open Connection in the navigator on the left side. This will, if the Bachmann controller has booted correctly and the PC is set up correctly, connect the Bachman controller to the PC.
9. If the controller is locked a dialog box will appear prompting for a username and password. The default user name and password can then be used in order to gain access. Username: M1, Password: bachmann

B.5 Reading and Recording Data from the Gyroscopes

.

This procedure explains the data from the gyroscopes can be accessed and saved to a file.

1. Locate the folder "KranTest" located on the C-drive of the computer.

2. Open the program ”GyroerKran”. The program will open a window in LabVeiw displaying the data from the gyroscopes in real time.

3. Use the folder icon in the LabVeiw window to select a location to save the data.

4. Start the data recorder by selecting the green ”play” button in the LabVeiw window.

Appendix C

Lab Controller Setup

A presentation of the control setup in the lab is presented in this appendix. The wiring of the control setup is presented in figure C.1. A description of the connections at the Bachmann controller is given in table C.1, and a description of the connections at the motor controllers are given in table refBaseMotorCon, C.3, and C.4.

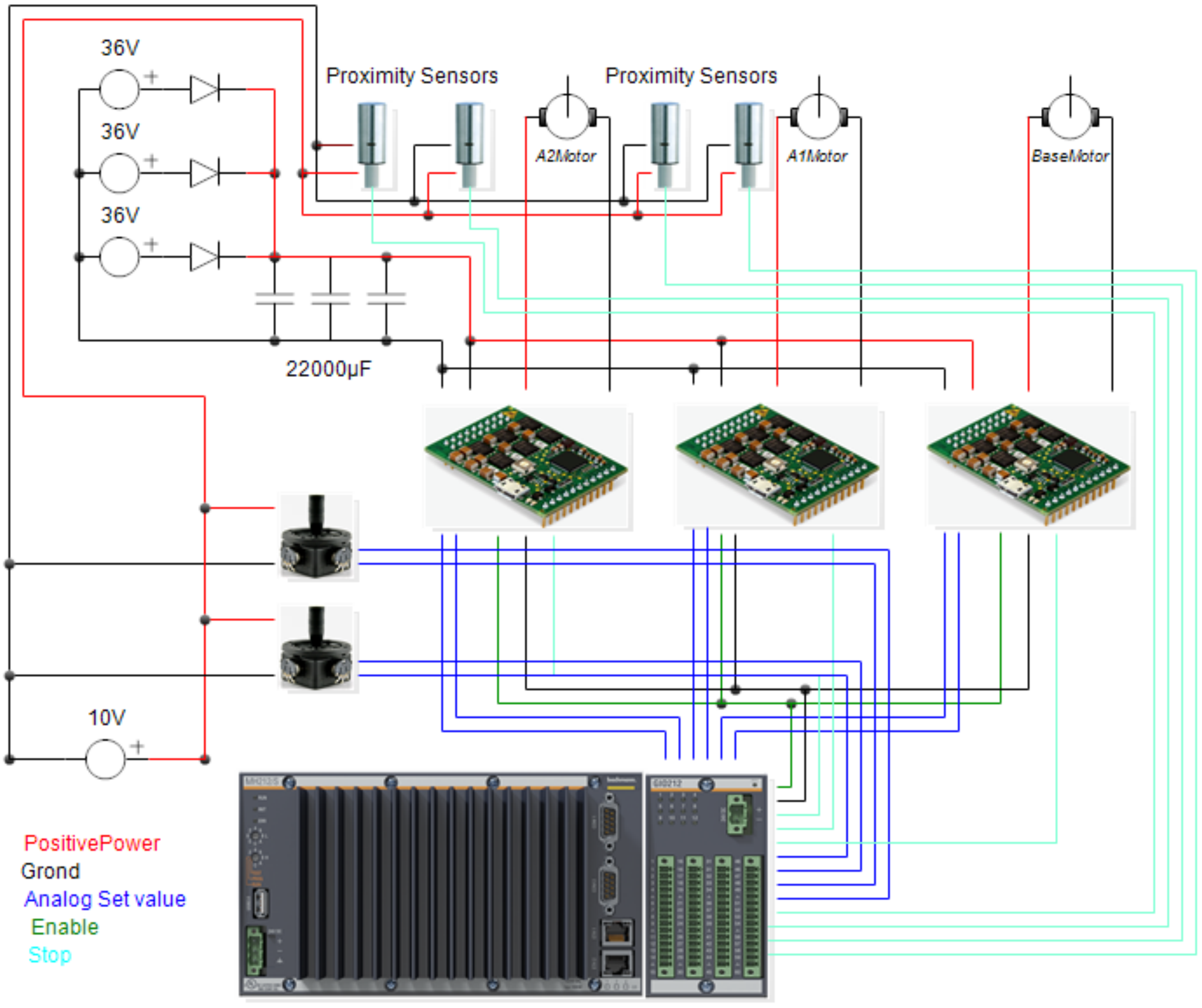


Figure C.1: Wire Diagram Controll System

Table C.1: Conections Bachman Controller

Bachman Controller		
Chanel	Name	Connected to
1	Enable	Escon Base, A1 and A2 motor module
2	SpeedOutA1	Escon A1 module
20	SpeedOutA2	Escon A2 module
22	SpeedOutB1	Escon Base module
18	StopOutA1	Escon A1 module
5	StopOutA2	Escon A2 module
3	SpeedInA1	Joystic 1
21	SpeedInA2	Joystic 2
23	SpeedInB1	Joystic1
4	StopInA11	Proximity Sensor 1
19	StopInA12	Proximity Sensor 2
6	StopInA21	Proximity Sensor 3
7	StopInA22	Proximity Sensor 4

Table C.2: Conections Escon Controller Base Motor

Escon base motor module			
Chanel	Name	Connected to	Name, ref. appendix F
1/2	DC Motor +	Base Motor	+
3/4	DC Motor -	Base Motor	-
7/8	36V Positive	Power Supply	+
9/10	36V Negative	Power Supply	-
21	Enable	Bachman Ch: 1	Enable +
23	Grond	Bachman CH: 1	Enable -
28	SetValue -	Bachman CH: 22	SpeedOutB1 -
29	SetValue +	Bachman CH: 22	SpeedOutB1 +

Table C.3: Conections Escon Controller Actuator1 Motor

Escon Actuator1 motor module			
Chanel	Name	Connected to	Name, ref. appendix F
1/2	DC Motor +	Base Motor	+
3/4	DC Motor -	Base Motor	-
7/8	36V Positive	Power Supply	+
9/10	36V Negative	Power Supply	-
21	Enable	Bachman Ch: 1	Enable +
22	Stop	Bachman Ch: 18	StopOutA1 +
23	Grond	Bachman CH: 1, 18	Enable -, StopOutA1 -
28	SetValue -	Bachman CH: 2	SpeedOutA1 -
29	SetValue +	Bachman CH: 2	SpeedOutA1 +

Table C.4: Conections Escon Controller Actuator2 Motor

Escon Actuator2 motor module			
Chanel	Name	Connected to	Name, ref. appendix F
1/2	DC Motor +	Base Motor	+
3/4	DC Motor -	Base Motor	-
7/8	36V Positive	Power Supply	+
9/10	36V Negative	Power Supply	-
21	Enable	Bachman Ch: 1	Enable +
22	Stop	Bachman Ch: 5	StopOutA2 +
23	Grond	Bachman CH: 1, 5	Enable -, StopOutA2 -
28	SetValue -	Bachman CH: 20	SpeedOutA2 -
29	SetValue +	Bachman CH: 20	SpeedOutA2 +

Appendix D

Maple Calculations

The code presented in this appendix is used in Maple to perform algebraic calculations necessary to develop the simulation model presented in chapter 4. The code defines all relevant positions in terms of a common reference frame and as functions of the generalized coordinates, calculates the necessary matrices and exports these as C-code. The exported C-code is used to build the Dll-file used as extension for the simulation model.

$\mathit{restart}:$
$\mathit{with}(\mathit{LinearAlgebra})$

Calculations for the Crane Kinematic Model

$\mathrm{pi} := \mathrm{Pi}:$

Defining delta1, as a function of theta2.

$\mathit{u\_x} := \sin(\mathit{theta2})\cdot \mathit{u}:$
$\mathit{b\_x} := \cos(\mathit{theta2})\cdot \mathit{b}:$
$\mathit{e\_x} := \mathit{b\_x} + \mathit{u\_x} - \mathit{a}:$
$\mathit{b\_z} := \sin(\mathit{theta2})\cdot \mathit{b}:$
$\mathit{u\_z} := \cos(\mathit{theta2})\cdot \mathit{u}:$
$\mathit{e\_z} := \mathit{h} + \mathit{b\_z} - \mathit{u\_z}:$
$\mathit{e} := \mathrm{sqrt}\left(\mathit{e\_z}^2 + \mathit{e\_x}^2\right):$
$\mathit{delta1} := \arcsin\left(\frac{\mathit{e\_z}}{\mathit{e}}\right):$

Defining delta2, as a function of theta2, and theta3.

$\mathit{c\_mark} := \mathrm{sqrt}\left(s^2 + c^2\right):$
$\mathit{beta2} := \arcsin\left(\frac{s}{\mathit{c\_mark}}\right):$
$\mathit{my2} := \mathrm{pi} + \mathit{theta3} - \mathit{beta2} - \mathit{epsilon2}:$
$v := \mathrm{sqrt}\left(r^2 + (\mathit{L1} - w)^2\right):$
$\mathit{epsilon2} := \arcsin\left(\frac{r}{v}\right):$
$d := \mathrm{sqrt}\left(v^2 + \mathit{c\_mark}^2 - 2\cdot(v\cdot \mathit{c\_mark}\cdot\cos(\mathit{my2}))\right):$
$\mathit{phi2} := \mathit{solve}\left(v = \mathrm{sqrt}\left(d^2 + \mathit{c\_mark}^2 - 2\cdot(d\cdot \mathit{c\_mark}\cdot\cos(\mathit{Phi2}))\right), \mathit{Phi2}\right):$
$\mathit{alpha2} := \mathrm{pi} - \left(\frac{\pi}{2} - \mathit{theta2}\right) + \mathit{theta3} - \mathit{beta2}:$
$\mathit{kappa2} := \mathrm{pi} - \mathit{alpha2} - \mathit{phi2}:$
$\mathit{delta2} := \frac{\mathrm{pi}}{2} - \mathit{kappa2}:$

Defining rotational matrices

#rotation of the base
$\mathit{R0\_1} := \mathit{Matrix}([[\cos(\mathit{theta1}), -\sin(\mathit{theta1}), 0], [\sin(\mathit{theta1}), \cos(\mathit{theta1}), 0], [0, 0, 1]]):$

#rotation of the lower arm
$\mathit{R1\_2} := \mathit{Matrix}([[1, 0, 0], [0, \cos(\mathit{theta2}), -\sin(\mathit{theta2})], [0, \sin(\mathit{theta2}), \cos(\mathit{theta2})]]):$

#rotation of the upper arm
$\mathit{R2\_3} := \mathit{Matrix}([[1, 0, 0], [0, \cos(\mathit{theta3}), -\sin(\mathit{theta3})], [0, \sin(\mathit{theta3}), \cos(\mathit{theta3})]]):$

```
#rotation of the lower actuator
R1_4 ≔ Matrix([[1, 0, 0], [0, cos(delta1), -sin(delta1)], [0, sin(delta1), cos(delta1)]]) :

#rotation of the lower actuator
R1_6 ≔ Matrix([[1, 0, 0], [0, cos(delta2), -sin(delta2)], [0, sin(delta2), cos(delta2)]]) :
```

Defining relevant positions as local coordinates

```
r0_1_0 ≔ Vector([0, 0, 0]) : #position of frame 1 relative to frame 0
r1_2_1 ≔ Vector([0, 0, h]) : #position of frame 2 relative to frame 1
r2_3_2 ≔ Vector([0, L1, 0]) : #position of frame 3 relative to frame 2
r1_4_1 ≔ Vector([0, a, 0]) : #position of frame 4 relative to frame 1
r2_5_2 ≔ Vector([0, b, -u]) : #position of frame 5 relative to frame 2
r2_6_2 ≔ Vector([0, w, -r]) : #position of frame 6 relative to frame 2
r3_7_3 ≔ Vector([0, c, -s]) : #position of frame 7 relative to frame 3
r3_8_3 ≔ Vector([0, L2, 0]) : #position of frame 8 relative to frame 3


r1_cm1_1 ≔ Vector([Xcm1, Ycm1, 0]) : #position of cm 1 relative to frame 1
r2_cm2_2 ≔ Vector([0, k, 0]) : #position of cm 2 relative to frame 2
r3_cm3_3 ≔ Vector([0, n, 0]) : #position of cm 3 relative to frame 3
r4_cm4_4 ≔ Vector([0, 0, p]) : #position of cm 4 relative to frame 4
r6_cm5_6 ≔ Vector([0, o, 0]) : #position of cm 5 relative to frame 6
```

Positions in terms of the 0-frame

```
r0_2_1 ≔ R0_1 r1_2_1 :
r0_3_2 ≔ R0_1 R1_2 r2_3_2 :
r0_4_1 ≔ R0_1 r1_4_1 :
r0_5_2 ≔ R0_1 R1_2 r2_5_2 :
r0_6_2 ≔ R0_1 R1_2 r2_6_2 :
r0_7_3 ≔ R0_1 R1_2 R2_3 r3_7_3 :
r0_8_3 ≔ R0_1 R1_2 R2_3 r3_8_3 :
```

Positions in terms and relative to the 0-frame

```
r0_2_0 ≔ r0_1_0 + r0_2_1 :
r0_3_0 ≔ r0_2_0 + r0_3_2 :
r0_4_0 ≔ r0_1_0 + r0_4_1 :
r0_5_0 ≔ r0_2_0 + r0_5_2 :
r0_6_0 ≔ r0_2_0 + r0_6_2 :
r0_7_0 ≔ r0_3_0 + r0_7_3 :
r0_8_0 ≔ r0_3_0 + r0_8_3 :
```

Unit vectors that each link revolve about

```
i_vect ≔ Vector([1, 0, 0]) :
```

```
j_vect := Vector([0, 1, 0]) :
k_vect := Vector([0, 0, 1]) :

e1 := R0_1 k_vect : #rotation of base
e2 := R0_1 R1_2 i_vect : #rotation of lower arm
e3 := R0_1 R1_2 R2_3 i_vect : #rotation of upper arm
```

Position for each Center of mass defined relative to and in terms of base frame

```
r0_cm1 := r0_1_0 + R0_1 r1_cm1_1 :#Mass of the platform
r0_cm2 := r0_2_0 + R0_1 R1_2 r2_cm2_2 :#Mass of lower arm
r0_cm3 := r0_3_0 + R0_1 R1_2 R2_3 r3_cm3_3 : #Mass of upper arm
r0_cm4 := r0_4_0  + R0_1 R1_4 r4_cm4_4 :#Mass of lower actuator
r0_cm5 := r0_6_0 + R0_1 R1_6 r6_cm5_6 :#Mass of upper actuator
```

Geometric Jacobian for each center of mass

```
#Mass of platform
r0_cm1_1 := r0_cm1 :
J1 := Matrix([[CrossProduct(e1, r0_cm1_1), ZeroMatrix(3, 2)], [e1, ZeroMatrix(3, 2)]]) :

#Mass of lower arm
r0_cm2_1 := r0_cm2 - r0_1_0 :
r0_cm2_2 := r0_cm2 - r0_2_0 :
J2 := Matrix([[CrossProduct(e1, r0_cm2_1), CrossProduct(e2, r0_cm2_2), ZeroMatrix(3, 1)], [e1,
    e2, ZeroMatrix(3, 1)]] :


#Mass of lower arm
r0_cm3_1 := r0_cm3 − r0_1_0 :
r0_cm3_2 := r0_cm3 - r0_2_0 :
r0_cm3_3 := r0_cm3 − r0_3_0 :
J3 := Matrix([[CrossProduct(e1, r0_cm3_1), CrossProduct(e2, r0_cm3_2), CrossProduct(e3,
    r0_cm3_3)], [e1, e2, e3]]) :


#Mass of lower actuator
r0_cm4_1 := r0_cm4 - r0_1_0 :
r0_cm4_2 := r0_cm4 − r0_2_0 :
J4 := Matrix([[CrossProduct(e1, r0_cm4_1), CrossProduct(e2, r0_cm4_2), ZeroMatrix(3, 1)],
      [e1, e2, ZeroMatrix(3, 1)]]) :


#Mass of upper actuator
r0_cm5_1 := r0_cm5 − r0_1_0 :
r0_cm5_2 := r0_cm5 − r0_2_0 :
r0_cm5_3 := r0_cm5 - r0_3_0 :
J5 := Matrix([[CrossProduct(e1, r0_cm5_1), CrossProduct(e2, r0_cm5_2), CrossProduct(e3,
    r0_cm5_3 ) ],  [e1, e2, e3 ] ]) :
```

Geometric Jacobian for each actuator conection

```
#Actuator 1 lower conection
r0_4_1 ≔ r0_4_0 -r0_1_0 :
J6 ≔ Matrix( [ [CrossProduct(e1, r0_4_1 ), ZeroMatrix(3, 2 ) ], [e1, ZeroMatrix(3, 2 ) ] ]) :

#Actuator 1 upper conection
r0_5_1 ≔ r0_5_0 - r0_1_0 :
r0_5_2 ≔ r0_5_0 − r0_2_0 :
J7 ≔ Matrix( [ [CrossProduct(e1, r0_5_1 ), CrossProduct(e2, r0_5_2 ), ZeroMatrix(3, 1 ) ],  [e1, e2,
    ZeroMatrix(3, 1 ) ] ]) :


#Actuator 2 lower conection
r0_6_1 ≔ r0_6_0 - r0_1_0 :
r0_6_2 ≔ r0_6_0 − r0_2_0 :
J8 ≔ Matrix( [ [CrossProduct(e1, r0_6_1 ), CrossProduct(e2, r0_6_2 ), ZeroMatrix(3, 1 ) ],  [e1, e2,
    ZeroMatrix(3, 1 ) ] ]) :


#Actuator 2 upper conection
r0_7_1 ≔ r0_7_0 − r0_1_0 :
r0_7_2 ≔ r0_7_0 − r0_2_0 :
r0_7_3 ≔ r0_7_0 - r0_3_0 :
J9 ≔ Matrix( [ [CrossProduct(e1, r0_7_1 ), CrossProduct(e2, r0_7_2 ), CrossProduct(e3, r0_7_3 ) ],
     [e1, e2, e3 ] ]) :
```

Jacobian for the Tip of the Crane

```
#Crane Tip
r0_8_1 ≔ r0_8_0 − r0_1_0 :
r0_8_2 ≔ r0_8_0 − r0_2_0 :
r0_8_3 ≔ r0_8_0 - r0_3_0 :
J10 ≔ Matrix( [ [CrossProduct(e1, r0_8_1 ), CrossProduct(e2, r0_8_2 ), CrossProduct(e3, r0_8_3 ) ],
     [e1, e2, e3 ] ]) :
```

Inertia Matrices in Local Coordinate Systems

```
I1_1 ≔ Matrix( [ [I_1x, 0, 0 ], [0, I_1y, 0 ], [0, 0, I_1z ] ]) : #Inertia of base
I2_2 ≔ Matrix( [ [I_2x, 0, 0 ], [0, I_2y, 0 ], [0, 0, I_2z ] ]) : #Inertia of lower arm
I3_3 ≔ Matrix( [ [I_3x, 0, 0 ], [0, I_3y, 0 ], [0, 0, I_3z ] ]) : #Inertia of upper arm
I4_4 ≔ Matrix( [ [I_4x, 0, 0 ], [0, I_4y, 0 ], [0, 0, I_4z ] ]) : #Inertia of lower actuator
```

```
I5_5 := Matrix([[I_5x, 0, 0], [0, I_5y, 0], [0, 0, I_5z]]) : #Inertia of upper actuator
```

Inertia Matrices in Terms of Base Frame

```
R1_0 := Transpose(R0_1) :
I1 := R0_1 I1_1 R1_0 :#Base

R0_2 := R0_1 R1_2 :
R2_0 := Transpose(R0_2) :
I2 := R0_2 I2_2 R2_0 : #Lower arm

R0_3 := R0_1 R1_2 R2_3 :
R3_0 := Transpose(R0_3) :
I3 := R0_3 I3_3 R3_0 :  #upper arm

R0_4 := R0_1 R1_4 :
R4_0 := Transpose(R0_4) :
I4 := R0_4 I4_4 R4_0 : #lower actuator

R0_6 := R0_1 R1_6 :
R6_0 := Transpose(R0_6) :
I5 := R0_6 I5_5 R6_0 :#Upper actuator
```

Mass Matrices

```
M1 := IdentityMatrix(3) m1 : #Base
M2 := IdentityMatrix(3) m2 : #Lower arm
M3 := IdentityMatrix(3) m3 : #Upper arm
M4 := IdentityMatrix(3) m4 : #Lower actuator
M5 := IdentityMatrix(3) m5 : #Upper actuator
```

Mass-Inertia Matrices

```
B1 := Transpose(J1) Matrix([[M1, ZeroMatrix(3, 3)], [ZeroMatrix(3, 3), I1]]) J1 :
B2 := Transpose(J2) Matrix([[M2, ZeroMatrix(3, 3)], [ZeroMatrix(3, 3), I2]]) J2 :
B3 := Transpose(J3) Matrix([[M3, ZeroMatrix(3, 3)], [ZeroMatrix(3, 3), I3]]) J3 :
B4 := Transpose(J4) Matrix([[M4, ZeroMatrix(3, 3)], [ZeroMatrix(3, 3), I4]]) J4 :
B5 := Transpose(J5) Matrix([[M5, ZeroMatrix(3, 3)], [ZeroMatrix(3, 3), I5]]) J5 :

B := simplify(B1 + B2 + B3 + B4 + B5) :
```

dBdq

```
dBdq1 := simplify(map(diff, B, theta1)) :
dBdq2 := simplify(map(diff, B, theta2)) :
dBdq3 := simplify(map(diff, B, theta3)) :
```

Code generation

```
with(CodeGeneration) :
with(codegen, optimize, makeproc) :

for i from 1 to 3 do for j from i to 3 do F := makeproc(B[i, j]); C(F, resultname = cat("B_", i, j),
    output = "bij.txt") :end do end do:

for i from 1 to 3 do for j from i to 3 do F := makeproc(dBdq1[i, j]); C(F, resultname
    = cat("dBdq1_", i, j), output = "bij.txt") :end do end do:
for i from 1 to 3 do for j from i to 3 do F := makeproc(dBdq2[i, j]); C(F, resultname
    = cat("dBdq2_", i, j), output = "bij.txt") :end do end do:
for i from 1 to 3 do for j from i to 3 do F := makeproc(dBdq3[i, j]); C(F, resultname
    = cat("dBdq3_", i, j), output = "bij.txt") :end do end do:
```

Appendix E

Simulation Model Source Code

The source code for most of the simulation model is enclosed in this appendix, the code for elements with a standard 20-Sim structure is excluded from the printed appendix, but can be found in the electronic attachment. The code for the controllers is also excluded from this printed appendix as the code is presented in section 4.3 The appendix is structured as similarly to how the simulation model is built up as possible. See chapter 4 for the presentation of the simulation model, and appendix D for how several of the equations in the code is derived.

E.1 Parameters

```
parameters

   real global g= 9.81; //Gravitational Constant

// --- Crane Configuration --- //
   real global config = 1; //1=Long Arm, All other is short arm

// --- Winch Assembly Properties ----//
   real global Dr0= 0.04; //Diameter of emptily reel [m]
   real global Wr= 0.13; //With of the reel [m]
   real global fw= 0.7; //Coefficient the packing of the wire on the reel
   real global Dw= 0.002;//Diameter of wire [m]
   real global WireOnReel= 150;//Initial amount of wire on the reel [m]
   real global Ew= 200000000000;//Wire E-module [Pa]
   real global RhoWire = 7200;//Weight of the wire kg/(m^3)
   real global sigma= 10; //Damping coefficient of wire
   real global RRI= 0.0099; //Reel Assembly Rotational Inertia. [kgm^2]
   real global BDR= 2; //Belt Drive Reduction
   real global BDE= 0.9; //Belt Drive Efficiency
   real global PMRI= 5.5e-005;// Pulley at Motor Rotational Inertia[kgm^2]
   real global NWE= 5; // Number of Wire Elements.
```

```
// --- Motor Properties --- //
   // RE50
   real global RE50TI= 0.000177;//Terminal inductance [H]
   real global RE50TR= 0.224;//Terminal resistance [ohm]
   real global RE50TC= 0.0604;//Torque constant [Nm/A]
   real global RE50SC= 158;//Speed constant [rpm/V]
   real global RE50RI= 5.6e-005;//Rotor inertia [kgm^2]
   //EC60
   real global EC60TI= 0.000188;//Terminal inductance [H]
   real global EC60TR= 0.307;//Terminal resistance [ohm]
   real global EC60TC= 0.0534;//Torque constant [Nm/A]
   real global EC60SC= 179;//Speed constant [rpm/V]
   real global EC60RI= 0.000121;//Rotor inertia [kgm^2]

// --- Gearbox Properties --- //
   real global LAGR= 26; //Lower Actuator Gearbox Reduction
   real global LAGE= 0.83; //Lower Actuator Gearbox Efficiency
   real global LAGI= 9.1e-007; //Lower Actuator Gearbox Rotational Inertia [kgm^2]
   real global UAGR= 15.17; //Upper Actuator Gearbox Reduction
   real global UAGE= 0.83; //Upper Actuator Gearbox Efficiency
   real global UAGI= 1.68e-006; //Lower Actuator Gearbox Rotational Inertia [kgm^2]
   real global BGR= 186; //Base Gearbox Reduction
   real global BGE= 0.68; //Base Gearbox Efficiency
   real global BGI= 1.77e-006; //Base Gearbox Rotational Inertia [kgm^2]
   real global WGR= 43;//Winch Gearbox Reduction
   real global WGE= 0.75;//Winch Gearbox Efficiency
   real global WGI= 1.73e-006;//Winch Gearbox Rotational Inertia [kgm^2]

// --- Trapezoidal Screw Properties --- //
   real global TSI= 0.00011; //Trapezoidal Screw Assembly Rotational Inertia [kgm^2]
   real global TSL= 0.01; //Trapezoidal Screw Lead [m/rpm]

// --- General Actuator Properties --- //
   real global MARN= 0.9; //Mass Actuator Rod and Nut [kg]

// --- Slewing ring Properties --- //
   real global SRSGI= 0.000353; //Slewing Ring Small Gear Rotational Inertia [kgm^2]
   real global SRSGT= 13; //Slewing Ring Small Gear Number of Teeth.
   real global SRLGT= 74; //Slewing Ring Large Gear Number of Teeth.

variables

// --- Geometric Parameters --- //
   real global L1,L2,b,a,h,w,c,u,r,s,k,n,o,p,Xcm1,Ycm1;

// --- Weights and Inertias --- //
   real global m0,m1,m2,m3,m4,m5;
   real global MassA1top;
   real global I_1z,I_2x,I_2z,I_3x,I_3z,I_4x,I_4z,I_5x,I_5z;

// --- Initial values ----- //
   real global theta1_init_deg,theta2_init_deg,theta3_init_deg,wire_init;

// --- Limits --- //
   real global A1Max,A1Min,A2Max,A2Min;

initialequations
```

```
  if config == 1 then
  //Long Arm

  L1= 2.954;    L2= 0.96;     b= 0.445;       a= 0.1;
  h= 0.495;     w= 2.150;     c= 0.19;        u= 0.280;
  r= 0.100;     s= 0.118;     k= 1.286;       n= 0.420;
  o= 0.324;     p= 0.429;     Xcm1 = 0;       Ycm1 = -0.023;
  m0 = 16;      m1 = 67;      m2 = 51.3;      m3 = 5.5;
  m4 = 11.8;    m5 = 6.8;     MassA1top=2.6;  I_1z = 1.8;
  I_2x = 120.1; I_2z = 120.2; I_3x = 1.6;     I_3z = 1.5;
  I_4x= 2.9;    I_4z= 3.0;    I_5x= 1.1;      I_5z= 1.1;
  theta1_init_deg = 0;
  theta2_init_deg = 38; //DesignMax=87, DesignMin=38
  theta3_init_deg = -57; //DesignMax=-28, DesignMin=-98
  wire_init= 1; //Initial Length of the wire from the crane tip [kgm^2]
  A1Max= 0.946;//DesignMax=946
  A1Min= 0.7095;//DesignMin=690
  A2Max= 0.913;//DesignMax=920
  A2Min= 0.6725;//DesignMin=664

  else
  //Short Arm

  L1= 1.251;    L2= 0.96;     b= 0.461;       a= 0.1;
  h= 0.495;     w= 0.461;     c= 0.19;        u= 0.140;
  r= 0.140;     s= 0.118;     k= 0.533;       n= 0.420;
  o= 0.324;     p= 0.440;     Xcm1 = 0;       Ycm1 = -0.023;
  m0 = 16;      m1 = 67;      m2 = 23.3;      m3 = 5.5;
  m4 = 8.5;     m5 = 6.8;     MassA1top=2.6;  I_1z = 1.8;
  I_2x = 9.2;   I_2z = 9.4;   I_3x = 1.6;     I_3z = 1.5;
  I_4x= 2.3;    I_4z= 2.3;    I_5x= 1.1;      I_5z= 1.1;
  theta1_init_deg = 0;
  theta2_init_deg = 80; //DesignMax=87, DesignMin=28//RealMax=84.5, RealMin=30.2
  theta3_init_deg = -90.5; //DesignMax=-23, DesignMin=-93//RealMax=-24.4,
      RealMin=-90.5
  wire_init= 1; //Initial Length of the wire from the crane tip [kgm^2]
  A1Max= 0.946;//DesignMax=951mm//RealMax=946mm
  A1Min= 0.7095;//DesignMin=695mm//RealMin=709.5mm
  A2Max= 0.913;//DesignMax=920//RealMax=913mm
  A2Min= 0.6725;//DesignMin=664//RealMin=672.5mm

  end;
```

E.2 IC-Element

```
variables
  real global m0,m1,m2,m3,m4,m5;
  real global theta1_init_deg,theta2_init_deg,theta3_init_deg;
  real global I_1z,I_2x,I_2z,I_3x,I_3z,I_4x,I_4z,I_5x,I_5z,Xcm1,Ycm1;
  real global L1,L2,b,a,h,w,c,u,r,s,k,n,o,p;
  real theta1_init,theta2_init,theta3_init;
```

```
   real theta1,theta2,theta3;
   real inarr[39,1],outarr[24,1];
   real B[3,3],dBdq1[3,3],dBdq2[3,3],dBdq3[3,3],C[3,3];
   real f_in[3,1],f_out[3,1],e_in[3,1],e_out[3,1];
initialequations
//setting initial values for the generalized coordinates
   theta1_init=theta1_init_deg*pi/180;
   theta2_init=theta2_init_deg*pi/180;
   theta3_init=theta3_init_deg*pi/180;
equations
//flow inn
   f_in[1,1]=p2.f;       f_in[2,1]=p4.f;       f_in[3,1]=p6.f;
//effort inn
   e_in[1,1]=p1.e;       e_in[2,1]=p3.e;       e_in[3,1]=p5.e;
//integrating to find generalized coordinates
   theta1=int(f_in[1,1],theta1_init);
   theta2=int(f_in[2,1],theta2_init);
   theta3=int(f_in[3,1],theta3_init);
//Building input vector
   inarr[1,1]=m1;        inarr[2,1]=m2;        inarr[3,1]=m3;
   inarr[4,1]=m4;        inarr[5,1]=m5;
   inarr[6,1]=0;"I_1x"   inarr[7,1]=0;"I_1y"   inarr[8,1]=I_1z;
   inarr[9,1]=I_2x;      inarr[10,1]=0;"I_2y"  inarr[11,1]=I_2z;
   inarr[12,1]=I_3x;     inarr[13,1]=0;"I_3y"  inarr[14,1]=I_3z;
   inarr[15,1]=I_4x;     inarr[16,1]=0;"I_4y"  inarr[17,1]=I_4z;
   inarr[18,1]=I_5x;     inarr[19,1]=0;"I_4y"  inarr[20,1]=I_5z;
   inarr[21,1]=theta1;   inarr[22,1]=theta2;   inarr[23,1]=theta3;
   inarr[24,1]=Xcm1;     inarr[25,1]=Ycm1;
   inarr[26,1]=L1;       inarr[27,1]=L2;       inarr[28,1]=a; inarr[29,1]=h;
   inarr[30,1]=b;        inarr[31,1]=k;        inarr[32,1]=w; inarr[33,1]=c;
   inarr[34,1]=n;        inarr[35,1]=u;        inarr[36,1]=r; inarr[37,1]=s;
   inarr[38,1]=p;        inarr[39,1]=o;
//Running dll file
   outarr = dll('Crane1.dll', 'Dynamics', inarr);
//building B matrix
   B[1,1]=outarr[1];     B[1,2]=outarr[2];     B[1,3]=outarr[3];
   B[2,1]=B[1,2];        B[2,2]=outarr[4];     B[2,3]=outarr[5];
   B[3,1]=B[1,3];        B[3,2]=B[2,3];        B[3,3]=outarr[6];
//building dbq1 matrix
   dBdq1[1,1]=outarr[7]; dBdq1[1,2]=outarr[8]; dBdq1[1,3]=outarr[9];
   dBdq1[2,1]=dBdq1[1,2]; dBdq1[2,2]=outarr[10]; dBdq1[2,3]=outarr[11];
   dBdq1[3,1]=dBdq1[1,3]; dBdq1[3,2]=dBdq1[2,3]; dBdq1[3,3]=outarr[12];
//building dBdq2 matrix
   dBdq2[1,1]=outarr[13]; dBdq2[1,2]=outarr[14]; dBdq2[1,3]=outarr[15];
   dBdq2[2,1]=dBdq2[1,2]; dBdq2[2,2]=outarr[16]; dBdq2[2,3]=outarr[17];
   dBdq2[3,1]=dBdq2[1,3]; dBdq2[3,2]=dBdq2[2,3]; dBdq2[3,3]=outarr[18];
//building dBdq3 matrix
   dBdq3[1,1]=outarr[19]; dBdq3[1,2]=outarr[20]; dBdq3[1,3]=outarr[21];
   dBdq3[2,1]=dBdq3[1,2]; dBdq3[2,2]=outarr[22]; dBdq3[2,3]=outarr[23];
   dBdq3[3,1]=dBdq3[1,3]; dBdq3[3,2]=dBdq3[2,3]; dBdq3[3,3]=outarr[24];
//setting effort and flow
   f_out=inverse(B)*int(e_in);
   C[1,1:3] = 0.5* transpose(f_in)*dBdq1 ;
   C[2,1:3] = 0.5* transpose(f_in)*dBdq2 ;
   C[3,1:3] = 0.5* transpose(f_in)*dBdq3 ;
   e_out=C*f_in;
   p1.f=f_out[1,1];      p2.e=e_out[1,1];
   p3.f=f_out[2,1];      p4.e=e_out[2,1];
   p5.f=f_out[3,1];      p6.e=e_out[3,1];
```

E.3 Calculation_of_Angles

```
variables
   real global L1,L2,b,a,h,w,c,u,r,s;
   real global theta1_init_deg,theta2_init_deg,theta3_init_deg;
   real theta1_init,theta2_init,theta3_init;
   real theta1,theta2,theta3,delta1,delta2,delta2_3D_animation;
   real theta1_deg,theta2_deg,theta3_deg,delta1_deg,delta2_deg;
   real theta1dott,theta2dott,theta3dott;
   real theta1dottdott,theta2dottdott,theta3dottdott;
   real theta1dottdott_rad,theta2dottdott_rad,theta3dottdott_rad;
   real LowerArmZpos,LowerArmZvelocity,LowerArmZacceleration;
   real UpperArmZpos,UpperArmZvelocity,UpperArmZacceleration;
initialequations
   theta1_init=theta1_init_deg*pi/180;
   theta2_init=theta2_init_deg*pi/180;
   theta3_init=theta3_init_deg*pi/180;
equations
// --- Calculating The Angular Positions --- //
   theta1 = int(q1)+theta1_init; theta2 = int(q2)+theta2_init;
   theta3 = int(q3)+theta3_init;
   delta1=arcsin((h+sin(theta2)*b-cos(theta2)*u)/((cos(theta2)*b+sin(theta2)*u-a)^2+
          (h+sin(theta2)*b-cos(theta2)*u)^2)^(1/2));
   delta2=theta2+theta3-arcsin(s/(c^2+s^2)^(1/2))+
          arccos(((L1^2-2*L1*w+r^2+w^2)^(1/2)*(c^2+s^2)^(1/2)*cos(theta3-arcsin(s/
          (c^2+s^2)^(1/2))-arcsin(r/(L1^2-2*L1*w+r^2+w^2)^(1/2)))+c^2+s^2)/
          (2*(L1^2-2*L1*w+r^2+w^2)^(1/2)*(c^2+s^2)^(1/2)*cos(theta3-arcsin(s/(c^2+s^2)^(1/2))-
          arcsin(r/(L1^2-2*L1*w+r^2+w^2)^(1/2)))+L1^2-2*L1*w+c^2+r^2+s^2+w^2)^(1/2)/
          (c^2+s^2)^(1/2));
   out[1,1] = theta1; out[2,1] = theta2; out[3,1] = theta3;
   out[4,1] = delta1; out[5,1] = delta2;

// --- Calculations For Plotting and 3D animation --- //
   theta1_deg=(theta1*180)/pi; theta2_deg=(theta2*180)/pi; theta3_deg=(theta3*180)/pi;
   delta1_deg=(delta1*180)/pi; delta2_deg=(delta2*180)/pi;
   delta2_3D_animation=(delta2-theta2);
   theta1dott=(q1*180)/(pi); theta2dott=(q2*180)/(pi); theta3dott=(q3*180)/(pi);
   theta1dottdott=ddt(theta1dott); theta2dottdott=ddt(theta2dott);
       theta3dottdott=ddt(theta3dott);
   theta1dottdott_rad=ddt(q1); theta2dottdott_rad=ddt(q2); theta3dottdott_rad=ddt(q3);
   LowerArmZpos=L1*sin(theta2)+h+0.07;
   LowerArmZvelocity=L1*cos(theta2)*q2;
   LowerArmZacceleration=L1*cos(theta2)*theta2dottdott_rad-L1*sin(theta2)*q2^2;
   UpperArmZpos=L2*sin(theta3+theta2);
   UpperArmZvelocity=L2*cos(theta3+theta2)*q3;
   UpperArmZacceleration=L2*cos(theta3+theta2)*theta3dottdott_rad-L2*sin(theta3+theta2)*q3^2;
```

E.4 Gravity

E.4.1 MTF-Element - $J1$

```
variables
  real global Xcm1,Ycm1;
  real theta1,J[3,3];
equations
  theta1=angles[1,1];
  J[1,1]=-sin(theta1) * Xcm1 - cos(theta1) * Ycm1; J[1,2]=0; J[1,3]=0;
  J[2,1]=cos(theta1) * Xcm1 - sin(theta1) * Ycm1;  J[2,2]=0; J[2,3]=0;
  J[3,1]=0;                                        J[3,2]=0; J[3,3]=0;
  p1.e = transpose(J) * p2.e;   p2.f = J * p1.f;
```

E.4.2 MTF-Element - $J2$

```
variables
  real global k;
  real theta1,theta2,J[3,3];
equations
  theta1=angles[1,1]; theta2=angles[2,1];
  J[1,1]=-cos(theta1)*(cos(theta2)*k);  J[1,2]=sin(theta1)*(sin(theta2)*k); J[1,3]=0;
  J[2,1]=-sin(theta1)*(cos(theta2)*k ); J[2,2]=-sin(theta1)*k;             J[2,3]=0;
  J[3,1]=0;                             J[3,2]=cos(theta2)*k;              J[3,3]=0;
  p1.e = transpose(J) * p2.e; p2.f = J * p1.f;
```

E.4.3 MTF-Element - $J3$

```
variables
  real global n,L1;
  real theta1,theta2,theta3,J[3,3];
equations
  theta1=angles[1,1]; theta2=angles[2,1]; theta3=angles[3,1];
  J[1,1]=-cos(theta1)*((n*cos(theta3)+L1)*cos(theta2)-sin(theta2)*(n*sin(theta3)));
  J[1,2]=((n*cos(theta3) +L1)*sin(theta2)+cos(theta2)*(n*sin(theta3)))*sin(theta1);
  J[1,3]=sin(theta1)*((n*sin(theta2))*cos(theta3)+sin(theta3)*(n*cos(theta2)));
  J[2,1]=-((n*cos(theta3)+L1)*cos(theta2)-sin(theta2)*(n*sin(theta3)))*sin(theta1);
  J[2,2]=-cos(theta1)*((n*cos(theta3)+L1)*sin(theta2)+cos(theta2)*(n*sin(theta3)));
  J[2,3]=-cos(theta1)*((n*sin(theta2))*cos(theta3)+sin(theta3)*(n*cos(theta2)));
  J[3,1]=0;
  J[3,2]=(n*cos(theta3)+L1)*cos(theta2)-sin(theta2)*(n*sin(theta3));
  J[3,3]=(-n*sin(theta2))*sin(theta3)+cos(theta3)*(n*cos(theta2));
  p1.e = transpose(J) * p2.e; p2.f = J * p1.f;
```

E.4.4 MTF-Element - $J4$

```
variables
   real global a,p,h;
   real theta1,delta1,J[3,3];
equations
   theta1=angles[1,1]; delta1=angles[4,1];
   J[1,1]=-cos(theta1)*(-sin(delta1)*p+a);
   J[1,2]=-sin(theta1)*(-cos(delta1)*p+h);
   J[1,3]=0;
   J[2,1]=-sin(theta1)*(-sin(delta1)*p+a);
   J[2,2]=cos(theta1)*(-cos(delta1)*p+h);
   J[2,3]=0;
   J[3,1]=0;
   J[3,2]=-sin(delta1) * p + a;
   J[3,3]=0;
   p1.e = -transpose(J) * p2.e; p2.f = -J * p1.f;
```

E.4.5 MTF-Element - $J5$

```
variables
   real global w,r,o,L1;
   real theta1,theta2,delta2,J[3,3];
equations
   theta1=angles[1,1]; theta2=angles[2,1]; delta2=angles[5,1];
   J[1,1]=-cos(theta1)*(cos(delta2)*o+cos(theta2)*w+sin(theta2)*r);
   J[1,2]=sin(theta1)*(sin(theta2)*w-cos(theta2)*r+sin(delta2)*o);
   J[1,3]=sin(theta1)*((w-L1)*sin(theta2)+sin(delta2)*o-cos(theta2)*r);
   J[2,1]=-sin(theta1)*(cos(delta2)*o+cos(theta2)*w+sin(theta2)*r);
   J[2,2]=-cos(theta1)*(sin(theta2)*w-cos(theta2)*r+sin(delta2)*o);
   J[2,3]=cos(theta1)*((w-L1)*sin(theta2)+sin(delta2)*o-cos(theta2)*r);
   J[3,1]=0;
   J[3,2]=cos(delta2)*o+cos(theta2)*w+sin(theta2)*r;
   J[3,3]=(w - L1)*cos(theta2)+cos(delta2)*o+sin(theta2)*r;
   p1.e = transpose(J) * p2.e; p2.f = J * p1.f;
```

E.5 Actuator_Conection

E.5.1 MTF-Element - $J6$

```
variables
   real global a;
   real theta1,J[3,3];
equations
   theta1=angles[1,1];
   J[1,1]=-cos(theta1)*a; J[1,2]=0; J[1,3]=0;
   J[2,1]=-sin(theta1)*a; J[2,2]=0; J[2,3]=0;
   J[3,1]=0;              J[3,2]=0; J[3,3]=0;
   p1.e = transpose(J) * p2.e; p2.f = J * p1.f;
```

E.5.2 MTF-Element - *J7*

```
variables
   real global b,u;
   real theta1,theta2,J[3,3];
equations
   theta1=angles[1,1]; theta2=angles[2,1];
   J[1,1]= -cos(theta1)*cos(theta2)*b-cos(theta1)*sin(theta2)*u;
   J[1,2]=sin(theta1)*(sin(theta2)*b-cos(theta2)*u);
   J[1,3]=0;
   J[2,1]=-sin(theta1)*cos(theta2)*b-sin(theta1)*sin(theta2)*u;
   J[2,2]=-cos(theta1)*(sin(theta2)*b-cos(theta2)*u);
   J[2,3]=0;
   J[3,1]=0;
   J[3,2]=cos(theta1)*(cos(theta1)*cos(theta2)*b+cos(theta1)*sin(theta2)*u)-
          sin(theta1)*(-sin(theta1)*cos(theta2)*b-sin(theta1)*sin(theta2)*u);
   J[3,3]=0;
   p1.e = transpose(J) * p2.e; p2.f = J * p1.f;
```

E.5.3 MTF-Element - *J8*

```
variables
   real global w,r;
   real theta1,theta2,J[3,3];
equations
   theta1=angles[1,1]; theta2=angles[2,1];
   J[1,1]=-cos(theta1)*cos(theta2)*w-cos(theta1)*sin(theta2)*r;
   J[1,2]=sin(theta1)*(sin(theta2)*w-cos(theta2)*r);
   J[1,3]=0;
   J[2,1]=-sin(theta1)*cos(theta2)*w-sin(theta1)*sin(theta2)*r;
   J[2,2]=-cos(theta1)*(sin(theta2)*w-cos(theta2)*r);
   J[2,3]=0;
   J[3,1]=0;
   J[3,2]=cos(theta1)*(cos(theta1)*cos(theta2)*w+cos(theta1)*sin(theta2)*r)-
          sin(theta1)*(-sin(theta1)*cos(theta2)*w-sin(theta1)*sin(theta2)*r);
   J[3,3]=0;
   p1.e = transpose(J) * p2.e; p2.f = J * p1.f;
```

E.5.4 MTF-Element - *J9*

```
variables
```

```
   real global L1,a,c,s;
   real theta1,theta2,theta3,J[3,3];
equations
   theta1=angles[1,1]; theta2=angles[2,1]; theta3=angles[3,1];
   J[1,1]=-cos(theta1)*cos(theta2)*L1-(cos(theta1)*cos(theta2)*cos(theta3)-cos(theta1)*
          sin(theta2)*sin(theta3))*c+(-cos(theta1)*cos(theta2)*sin(theta3)-cos(theta1)*
          sin(theta2)*cos(theta3))*s;
   J[1,2]=sin(theta1)*(sin(theta2)*L1+(sin(theta2)*cos(theta3)+cos(theta2)*sin(theta3))*c-
          (-sin(theta2)*sin(theta3)+cos(theta2)*cos(theta3))*s);
   J[1,3]=sin(theta1)*((sin(theta2)*cos(theta3)+cos(theta2)*sin(theta3))*c-(-sin(theta2)*
          sin(theta3)+cos(theta2)*cos(theta3))*s);
   J[2,1]=-sin(theta1)*cos(theta2)*L1+(-sin(theta1)*cos(theta2)*cos(theta3)+sin(theta1)*
          sin(theta2)*sin(theta3))*c-(sin(theta1)*cos(theta2)*sin(theta3)+sin(theta1)*
          sin(theta2)*cos(theta3))*s;
   J[2,2]=-cos(theta1)*(sin(theta2)*L1+(sin(theta2)*cos(theta3)+cos(theta2)*sin(theta3))*c-
          (-sin(theta2)*sin(theta3)+cos(theta2)*cos(theta3))*s);
   J[2,3]=-cos(theta1)*((sin(theta2)*cos(theta3)+cos(theta2)*sin(theta3))*c-(-sin(theta2)*
          sin(theta3)+cos(theta2)*cos(theta3))*s);
   J[3,1]=0;
   J[3,2]=cos(theta1)*(cos(theta1)*cos(theta2)*L1+(cos(theta1)*cos(theta2)*cos(theta3)-
          cos(theta1)*sin(theta2)*sin(theta3))*c-(-cos(theta1)*cos(theta2)*sin(theta3)-
          cos(theta1)*sin(theta2)*cos(theta3))*s)-sin(theta1)*(-sin(theta1)*cos(theta2)*L1+
          (-sin(theta1)*cos(theta2)*cos(theta3)+sin(theta1)*sin(theta2)*sin(theta3))*c-
          (sin(theta1)*cos(theta2)*sin(theta3)+sin(theta1)*sin(theta2)*cos(theta3))*s);
   J[3,3]=cos(theta1)*((cos(theta1)*cos(theta2)*cos(theta3)-cos(theta1)*sin(theta2)*
          sin(theta3))*c-(-cos(theta1)*cos(theta2)*sin(theta3)-cos(theta1)*sin(theta2)*
          cos(theta3))*s)-sin(theta1)*((-sin(theta1)*cos(theta2)*cos(theta3)+sin(theta1)*
          sin(theta2)*sin(theta3))*c-(sin(theta1)*cos(theta2)*sin(theta3)+sin(theta1)*
          sin(theta2)*cos(theta3))*s);
   p1.e = transpose(J) * p2.e; p2.f = J * p1.f;
```

E.5.5 MTF-Element - *A1_Local_To_Global*

```
variables
   real global b,a,h,u,theta1_init_deg,theta2_init_deg;
   real theta1,theta2,delta1,ex,ey,ez;
equations
   theta1=angles[1,1]; theta2=angles[2,1]; delta1=angles[4,1];
   ex=-sin(theta1)*(cos(theta2)*b+sin(theta2)*u-a);
   ey=cos(theta1)*(cos(theta2)*b+sin(theta2)*u-a);
   ez=h+sin(theta2)*b-cos(theta2)*u;
   p2.e[1,1] = -sin(theta1)*cos(delta1)*p1.e;
   p2.e[2,1] = cos(theta1)*cos(delta1)*p1.e;
   p2.e[3,1] = sin(delta1)*p1.e;
   p1.f =((ex * p2.f[1,1]) + (ey * p2.f[2,1]) + (ez *
      p2.f[3,1]))/sqrt((ex^2)+(ey^2)+(ez^2));
```

E.5.6 MTF-Element - *A2_Local_To_Global*

```
variables
   real global L1,w,c,r,s;
   real theta1,theta2,theta3,delta2,dx,dy,dz;
equations
   theta1=angles[1,1]; theta2=angles[2,1]; theta3=angles[3,1]; delta2=angles[5,1];
   dx=-((c*cos(theta3)+sin(theta3)*s-w+L1)*cos(theta2)-sin(theta2)*(c*sin(theta3)-
      s*cos(theta3)+r))*sin(theta1);
   dy=((c*cos(theta3)+sin(theta3)*s-w+L1)*cos(theta2)-sin(theta2)*(c*sin(theta3)-
      s*cos(theta3)+r))*cos(theta1);
   dz=(c*cos(theta3)+sin(theta3)*s-w+L1)*sin(theta2)+cos(theta2)*(c*sin(theta3)-
      s*cos(theta3)+r);
   p2.e[1,1] = -sin(theta1)*cos(delta2)*p1.e;
   p2.e[2,1] = cos(theta1)*cos(delta2)*p1.e;
   p2.e[3,1] = sin(delta2)*p1.e;
   p1.f =((dx*p2.f[1,1])+(dy*p2.f[2,1])+(dz*p2.f[3,1]))/sqrt((dx^2)+(dy^2)+(dz^2));
```

E.5.7 Integrator - *Actuator_pos_calculations*

```
variables
   real global L1,b,a,h,w,c,u,r,s,theta1_init_deg,theta2_init_deg,theta3_init_deg;
   real theta1_init,theta2_init,theta3_init;
   real A11_Xpos_init,A11_Ypos_init,A11_Zpos_init;
   real A12_Xpos_init,A12_Ypos_init,A12_Zpos_init;
   real A21_Xpos_init,A21_Ypos_init,A21_Zpos_init;
   real A22_Xpos_init,A22_Ypos_init,A22_Zpos_init;
   real e_init,d_init;
   real A11_Xpos,A11_Ypos,A11_Zpos;
   real A12_Xpos,A12_Ypos,A12_Zpos;
   real A21_Xpos,A21_Ypos,A21_Zpos;
   real A22_Xpos,A22_Ypos,A22_Zpos;
   real e,e_3D_animation,d,d_3D_animation;
initialequations
   theta1_init=theta1_init_deg*pi/180;
   theta2_init=theta2_init_deg*pi/180;
   theta3_init=theta3_init_deg*pi/180;
   A11_Xpos_init=-sin(theta1_init) * a;
   A11_Ypos_init=cos(theta1_init) * a;
   A11_Zpos_init=0;
   A12_Xpos_init=-sin(theta1_init) * cos(theta2_init) * b - sin(theta1_init) *
       sin(theta2_init) * u;
   A12_Ypos_init=cos(theta1_init) * cos(theta2_init) * b + cos(theta1_init) *
       sin(theta2_init) * u;
   A12_Zpos_init=h + sin(theta2_init) * b - cos(theta2_init) * u;
   A21_Xpos_init=-sin(theta1_init) * cos(theta2_init) * w - sin(theta1_init) *
       sin(theta2_init) * r;
   A21_Ypos_init=cos(theta1_init) * cos(theta2_init) * w + cos(theta1_init) *
       sin(theta2_init) * r;
   A21_Zpos_init=h + sin(theta2_init) * w - cos(theta2_init) * r;
   A22_Xpos_init=-sin(theta1_init) * cos(theta2_init) * L1 + (-sin(theta1_init) *
       cos(theta2_init) * cos(theta3_init) + sin(theta1_init) * sin(theta2_init) *
       sin(theta3_init)) * c - (sin(theta1_init) * cos(theta2_init) * sin(theta3_init)
       + sin(theta1_init) * sin(theta2_init) * cos(theta3_init)) * s;
   A22_Ypos_init=cos(theta1_init) * cos(theta2_init) * L1 + (cos(theta1_init) *
       cos(theta2_init) * cos(theta3_init) - cos(theta1_init) * sin(theta2_init) *
       sin(theta3_init)) * c - (-cos(theta1_init) * cos(theta2_init) *
       sin(theta3_init) - cos(theta1_init) * sin(theta2_init) * cos(theta3_init)) * s;
   A22_Zpos_init=h + sin(theta2_init) * L1 + (sin(theta2_init) * cos(theta3_init) +
       cos(theta2_init) * sin(theta3_init)) * c - (-sin(theta2_init) *
       sin(theta3_init) + cos(theta2_init) * cos(theta3_init)) * s;
   e_init=((cos(theta2_init)*b+sin(theta2_init)*u-a)^2+(h+sin(theta2_init)*b-
           cos(theta2_init)*u)^2)^(1/2);
   d_init=(u^2+(L1-w)^2+c^2+s^2+2*(u^2+(L1-w)^2)^(1/2)*(c^2+s^2)^(1/2)*cos(theta3_init-
           arcsin(s/(c^2+s^2)^(1/2))-arcsin(u/(u^2+(L1-w)^2)^(1/2))))^(1/2);
equations
   A11_Xpos = int(A11[1,1])+A11_Xpos_init;
   A11_Ypos = int(A11[2,1])+A11_Ypos_init;
   A11_Zpos = int(A11[3,1])+A11_Zpos_init;
   A12_Xpos = int(A12[1,1])+A12_Xpos_init;
   A12_Ypos = int(A12[2,1])+A12_Ypos_init;
   A12_Zpos = int(A12[3,1])+A12_Zpos_init;
   A21_Xpos = int(A21[1,1])+A21_Xpos_init;
   A21_Ypos = int(A21[2,1])+A21_Ypos_init;
   A21_Zpos = int(A21[3,1])+A21_Zpos_init;
   A22_Xpos = int(A22[1,1])+A22_Xpos_init;
   A22_Ypos = int(A22[2,1])+A22_Ypos_init;
   A22_Zpos = int(A22[3,1])+A22_Zpos_init;
   e=int(e_dott)+e_init; d=int(d_dott)+d_init;
   out[1,1]=e_init;      out[2,1]=e;
   out[3,1]=d_init;      out[4,1]=d;

// --- Calculations for 3D animation --- //
   e_3D_animation=e*100;
   d_3D_animation=d*100;
```

E.6 Actuator_Limit_Swhiches

```
variables
   real global A1Max,A1Min,A2Max,A2Min;
   real stopA11, stopA12, stopA21, stopA22,A1,A2;
initialequations
stopA11=0; stopA12=0; stopA21=0; stopA22=0;
equations
   A1=input[2,1]; A2=input[4,1];
   if (A1>= A1Max) then
      stopA12 = 1;
   else
      stopA12 = 0;
   end;
   if (A1<=A1Min) then
      stopA11 = 1;
   else
      stopA11 = 0;
   end;
   if (A2>=A2Max) then
```

```
      stopA22 = 1;
   else
      stopA22 = 0;
   end;
   if (A2<=A2Min) then
      stopA21 = 1;
   else
      stopA21 = 0;
   end;
   output[1,1]=stopA11; output[2,1]=stopA12;
   output[3,1]=stopA21; output[4,1]=stopA22;
```

E.7 Wire_and_Winch

E.7.1 MTF-Element $J10$

```
variables
   real global L1,L2;
   real theta1,theta2,theta3,J[3,3];
equations
   theta1=angles[1,1]; theta2=angles[2,1]; theta3=angles[3,1];
   J[1,1]=-cos(theta1) * cos(theta2) * L1 - (cos(theta1) * cos(theta2) * cos(theta3) -
       cos(theta1) * sin(theta2) * sin(theta3)) * L2;
   J[1,2]=sin(theta1) * (sin(theta2) * L1 + (sin(theta2) * cos(theta3) + cos(theta2) *
       sin(theta3)) * L2);
   J[1,3]=sin(theta1) * (sin(theta2) * cos(theta3) + cos(theta2) * sin(theta3)) * L2;
   J[2,1]=-sin(theta1) * cos(theta2) * L1 + (-sin(theta1) * cos(theta2) * cos(theta3)
       + sin(theta1) * sin(theta2) * sin(theta3)) * L2;
   J[2,2]=-cos(theta1) * (sin(theta2) * L1 + (sin(theta2) * cos(theta3) + cos(theta2)
       * sin(theta3)) * L2);
   J[2,3]=-cos(theta1) * (sin(theta2) * cos(theta3) + cos(theta2) * sin(theta3)) * L2;
   J[3,1]=0;
   J[3,2]=cos(theta1) * (cos(theta1) * cos(theta2) * L1 + (cos(theta1) * cos(theta2) *
       cos(theta3) - cos(theta1) * sin(theta2) * sin(theta3)) * L2) - sin(theta1) *
       (-sin(theta1) * cos(theta2) * L1 + (-sin(theta1) * cos(theta2) * cos(theta3) +
       sin(theta1) * sin(theta2) * sin(theta3)) * L2);
   J[3,3]=cos(theta1) * (cos(theta1) * cos(theta2) * cos(theta3) - cos(theta1) *
       sin(theta2) * sin(theta3)) * L2 - sin(theta1) * (-sin(theta1) * cos(theta2) *
       cos(theta3) + sin(theta1) * sin(theta2) * sin(theta3)) * L2;
   p2.e = transpose(J) * p1.e; p1.f = J * p2.f;
```

E.8 WireElement

E.8.1 MTF-Element $MTF1$

```
variables
  real X0_dott, Y0_dott, Z0_dott;
  real X0, Y0, Z0, X1, Y1, Z1;
  real rt1, rt2, rt3;
equations
  X0_dott=p1.f[1,1]; Y0_dott=p1.f[2,1]; Z0_dott=p1.f[3,1];
  X0=input[1,1];     Y0=input[2,1];     Z0=input[3,1];
  X1=input[4,1];     Y1=input[5,1];     Z1=input[6,1];
  rt1=(X1-X0)/(sqrt((X1-X0)^2+(Y1-Y0)^2+(Z1-Z0)^2)+0.0001);
  rt2=(Y1-Y0)/(sqrt((X1-X0)^2+(Y1-Y0)^2+(Z1-Z0)^2)+0.0001);
  rt3=(Z1-Z0)/(sqrt((X1-X0)^2+(Y1-Y0)^2+(Z1-Z0)^2)+0.0001);
  p1.e[1,1]=rt1*p2.e; p1.e[2,1]=rt2*p2.e; p1.e[3,1]=rt3*p2.e;
  p2.f = rt1 * X0_dott + rt2 * Y0_dott + rt3 * Z0_dott;
```

E.8.2 MTF-Element $MTF2$

```
variables
  real X1_dott, Y1_dott, Z1_dott;
  real X0, Y0, Z0, X1, Y1, Z1;
  real rt1, rt2, rt3;
equations
  X1_dott=p1.f[1,1]; Y1_dott=p1.f[2,1]; Z1_dott=p1.f[3,1];
  X0=input[1,1];     Y0=input[2,1];     Z0=input[3,1];
  X1=input[4,1];     Y1=input[5,1];     Z1=input[6,1];
  rt1=(X1-X0)/(sqrt((X1-X0)^2+(Y1-Y0)^2+(Z1-Z0)^2)+0.0001);
  rt2=(Y1-Y0)/(sqrt((X1-X0)^2+(Y1-Y0)^2+(Z1-Z0)^2)+0.0001);
  rt3=(Z1-Z0)/(sqrt((X1-X0)^2+(Y1-Y0)^2+(Z1-Z0)^2)+0.0001);
  p1.e[1,1]=rt1*p2.e; p1.e[2,1]=rt2*p2.e; p1.e[3,1]=rt3*p2.e;
  p2.f = rt1 * X1_dott + rt2 * Y1_dott + rt3 * Z1_dott;
```

E.8.3 Pos

```
variables
  real global L1,L2,h,theta1_init_deg,theta2_init_deg,theta3_init_deg,wire_init;
  real theta1_init, theta2_init, theta3_init;
  real X0init, Y0init, Z0init;
  real X0,Y0,Z0,X1,Y1,Z1;
  real theta_Wzy1, theta_Wzx1;
initialequations
  theta1_init=(theta1_init_deg*pi)/180;
  theta2_init=(theta2_init_deg*pi)/180;
  theta3_init=(theta3_init_deg*pi)/180;
  X0init=-sin(theta1_init) * cos(theta2_init) * L1 + (-sin(theta1_init) *
     cos(theta2_init) * cos(theta3_init) + sin(theta1_init) * sin(theta2_init) *
     sin(theta3_init)) * L2;
  Y0init=cos(theta1_init) * cos(theta2_init) * L1 + (cos(theta1_init) *
     cos(theta2_init) * cos(theta3_init) - cos(theta1_init) * sin(theta2_init) *
     sin(theta3_init)) * L2;
```

```
   Z0init=h + sin(theta2_init) * L1 + (sin(theta2_init) * cos(theta3_init) +
       cos(theta2_init) * sin(theta3_init)) * L2;
equations
   X0=int(XYZ0in[1,1])+X0init;
   Y0=int(XYZ0in[2,1])+Y0init;
   Z0=int(XYZ0in[3,1])+Z0init;
   X1=int(XYZ1in[1,1])+X0init;
   Y1=int(XYZ1in[2,1])+Y0init;
   Z1=int(XYZ1in[3,1])+Z0init-(wire_init/5);
   Out1=[X0;Y0;Z0;X1;Y1;Z1];
   theta_Wzy1=arctan((Y1-Y0)/(Z1-Z0));
   theta_Wzx1=arctan((X1-X0)/(Z1-Z0));
```

Appendix F

Controll Code for Bachmann Controller

The sequences of code added to the control code template for the Bachmann controller are enclosed in the sections below. See appendix B.2 to understand how the control program is constructed. The complete control code is considered to be too large to include in the printed appendix the complete code is therefore only included in the electronic attachments submitted with this thesis.

F.1 Variables

```
MLOCAL UINT32 ControllerCycleCount = 0;

MLOCAL SVI_ADDR EnableAddr;
MLOCAL SVI_ADDR SpeedOutA1Addr;
MLOCAL SVI_ADDR SpeedOutA2Addr;
MLOCAL SVI_ADDR SpeedOutB1Addr;
MLOCAL SVI_ADDR SpeedInA1Addr;
MLOCAL SVI_ADDR SpeedInA2Addr;
MLOCAL SVI_ADDR SpeedInB1Addr;
MLOCAL SVI_ADDR StopOutA1Addr;
MLOCAL SVI_ADDR StopOutA2Addr;
MLOCAL SVI_ADDR StopInA11Addr;
MLOCAL SVI_ADDR StopInA12Addr;
MLOCAL SVI_ADDR StopInA21Addr;
MLOCAL SVI_ADDR StopInA22Addr;

MLOCAL SINT32 StopA1 = 0;
MLOCAL SINT32 StopA2 = 0;
MLOCAL UINT32 SpeedInA1 = 0;
MLOCAL UINT32 SpeedInA2 = 0;
MLOCAL UINT32 SpeedInB1 = 0;
MLOCAL REAL32 SpeedOutA1 = 0;
MLOCAL REAL32 SpeedInA1Corr = 0;
MLOCAL REAL32 SpeedOutA2 = 0;
MLOCAL REAL32 SpeedInA2Corr = 0;
MLOCAL REAL32 SpeedOutB1 =0;
MLOCAL REAL32 SpeedInB1Corr = 0;
MLOCAL UINT32 StopInA11 = 0;
MLOCAL UINT32 StopInA12 = 0;
MLOCAL UINT32 StopInA21 = 0;
MLOCAL UINT32 StopInA22 = 0;
```

F.2 Controll_Cycle

```
MLOCAL VOID Control_Cycle(VOID){
   if (ControllerCycleCount >= 1) {
      //Updating variables.
            if (svi_GetVal(pSviLib, SpeedInA1Addr, &SpeedInA1) < 0) {
                       log_Err("Failed to write value");}

            if (svi_GetVal(pSviLib, SpeedInA2Addr, &SpeedInA2) < 0) {
                       log_Err("Failed to write value");}

            if (svi_GetVal(pSviLib, SpeedInB1Addr, &SpeedInB1) < 0) {
                       log_Err("Failed to write value");}

            if (svi_GetVal(pSviLib, StopInA11Addr, &StopInA11) < 0) {
                       log_Err("Failed to write value");}

            if (svi_GetVal(pSviLib, StopInA12Addr, &StopInA12) < 0) {
                       log_Err("Failed to write value");}

            if (svi_GetVal(pSviLib, StopInA21Addr, &StopInA21) < 0) {
                       log_Err("Failed to write value");}

            if (svi_GetVal(pSviLib, StopInA22Addr, &StopInA22) < 0) {
                       log_Err("Failed to write value");}

            // Setting "dead-zone" for A1.
            SpeedInA1Corr=0;
            if (SpeedInA1>19000 || SpeedInA1<15400){
              SpeedInA1Corr=(SpeedInA1*2)-(34405);}

            // Setting "dead-zone" for A2.
            SpeedInA2=SpeedInA2+20;
            SpeedInA2Corr=0;
            if (SpeedInA2>19000 || SpeedInA2<15400){
              SpeedInA2Corr=(SpeedInA2*2)-(34405);}

            // Setting "dead-zone" for B1.
            SpeedInB1Corr=0;
            if (SpeedInB1>19000 || SpeedInB1<15400){
              SpeedInB1Corr=(SpeedInB1*2)-(34405);}
            // Setting Speed the motors
            SpeedOutA1=-SpeedInA1Corr;
            SpeedOutA2=-SpeedInA2Corr;
            SpeedOutB1=SpeedInB1Corr;

            // Checking if the actuators are at the boundaries.
            StopA1=0;
            StopA2=0;

           if (StopInA11==1 && SpeedInA1>16400){
              StopA1=1;}

            if (StopInA12==1 && SpeedInA1<18000){
              StopA1=1;}

            if (StopInA21==1 && SpeedInA2>16400){
              StopA2=1;}

            if (StopInA22==1 && SpeedInA2<18000){
              StopA2=1;}

            // Updating output variables.
            SetValue(StopOutA1Addr, StopA1);
            SetValue(SpeedOutA1Addr, SpeedOutA1);
            SetValue(StopOutA2Addr, StopA2);
            SetValue(SpeedOutA2Addr, SpeedOutA2);
            SetValue(SpeedOutB1Addr, SpeedOutB1);
   }
}
```

F.3 Control_CycleEnd

```
MLOCAL VOID Control_CycleEnd(TASK_PROPERTIES * pTaskData){
{ ... }
    ControllerCycleCount++;
}
```

F.4 Controller_Init

```
MLOCAL SINT32 Controller_Init(VOID) {
{ ... }
    /* Convert symbolic address "5/1" to binary SVI address. */
    if (svi_GetAddr(pSviLib, "5/1", &EnableAddr, &SviFormat) != SVI_E_OK) {
        log_Wrn("Lights_Init: Can't get address of value '5/1'!");
        return (ERROR);}
    /* Convert symbolic address "5/2" to binary SVI address. */
    if (svi_GetAddr(pSviLib, "5/2", &SpeedOutA1Addr, &SviFormat) != SVI_E_OK) {
        log_Wrn("Lights_Init: Can’t get address of value ’5/2’!");
        return (ERROR);}
    /* Convert symbolic address "5/20" to binary SVI address. */
    if (svi_GetAddr(pSviLib, "5/20", &SpeedOutA2Addr, &SviFormat) != SVI_E_OK) {
        log_Wrn("Lights_Init: Can’t get address of value ’5/20’!");
        return (ERROR);}
    /* Convert symbolic address "5/22" to binary SVI address. */
    if (svi_GetAddr(pSviLib, "5/22", &SpeedOutB1Addr, &SviFormat) != SVI_E_OK) {
        log_Wrn("Lights_Init: Can’t get address of value ’5/22’!");
        return (ERROR);}
    /* Convert symbolic address "5/18" to binary SVI address. */
    if (svi_GetAddr(pSviLib, "5/18", &StopOutA1Addr, &SviFormat) != SVI_E_OK) {
        log_Wrn("Lights_Init: Can’t get address of value ’5/18’!");
        return (ERROR);}
    /* Convert symbolic address "5/5" to binary SVI address. */
    if (svi_GetAddr(pSviLib, "5/5", &StopOutA2Addr, &SviFormat) != SVI_E_OK) {
        log_Wrn("Lights_Init: Can’t get address of value ’5/5’!");
        return (ERROR);}
    /* Convert symbolic address "5/3" to binary SVI address. */
    if (svi_GetAddr(pSviLib, "5/3", &SpeedInA1Addr, &SviFormat) != SVI_E_OK) {
        log_Wrn("Lights_Init: Can’t get address of value ’5/3’!");
        return (ERROR); }
    /* Convert symbolic address "5/21" to binary SVI address. */
    if (svi_GetAddr(pSviLib, "5/21", &SpeedInA2Addr, &SviFormat) != SVI_E_OK) {
        log_Wrn("Lights_Init: Can’t get address of value ’5/3’!");
        return (ERROR);}
    /* Convert symbolic address "5/23" to binary SVI address. */
    if (svi_GetAddr(pSviLib, "5/23", &SpeedInB1Addr, &SviFormat) != SVI_E_OK) {
        log_Wrn("Lights_Init: Can’t get address of value ’5/23’!");
        return (ERROR);}
    /* Convert symbolic address "5/19" to binary SVI address. */
    if (svi_GetAddr(pSviLib, "5/19", &StopInA12Addr, &SviFormat) != SVI_E_OK) {
        log_Wrn("Lights_Init: Can’t get address of value ’5/19’!");
        return (ERROR); }
    /* Convert symbolic address "5/4" to binary SVI address. */
    if (svi_GetAddr(pSviLib, "5/4", &StopInA11Addr, &SviFormat) != SVI_E_OK) {
        log_Wrn("Lights_Init: Can’t get address of value ’5/4’!");
        return (ERROR);}
    /* Convert symbolic address "5/6" to binary SVI address. */
    if (svi_GetAddr(pSviLib, "5/6", &StopInA21Addr, &SviFormat) != SVI_E_OK) {
        log_Wrn("Lights_Init: Can’t get address of value ’5/6’!");
        return (ERROR);}
    /* Convert symbolic address "5/7" to binary SVI address. */
    if (svi_GetAddr(pSviLib, "5/7", &StopInA22Addr, &SviFormat) != SVI_E_OK) {
        log_Wrn("Lights_Init: Can’t get address of value ’5/7’!");
        return (ERROR);}
    return (OK);
}
```

F.5 Controll_AppEOI

```
SINT32 controll_AppEOI(VOID){
{ ... }
       /* init controller */
                     if (Controller_Init() < 0)
                         break;
{ ... }
```

F.6 SetValue

```
MLOCAL VOID SetValue(SVI_ADDR SviAddr, UINT32 value) {
    if (svi_SetVal(pSviLib, SviAddr, value) < 0) {
        log_Err("Failed to write value");
    }
}
```

Appendix G

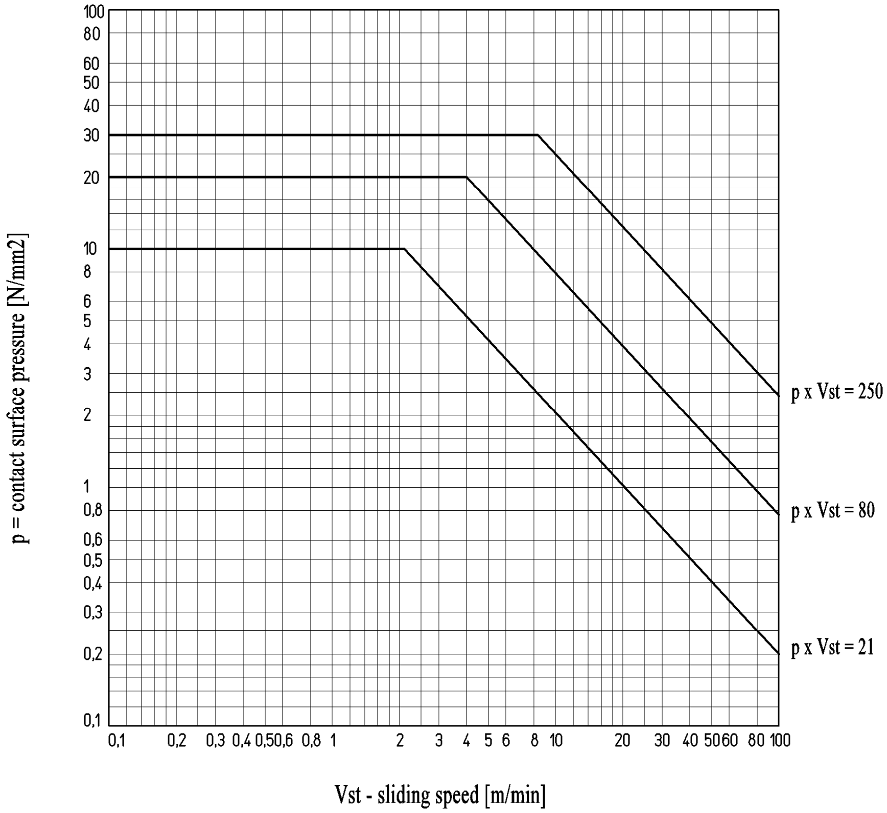
Trapezodial Screw and Nut Sizing Criteria

CONTI

Bronze nut sizing

As per the bronze nut, the study of the product p•Vst allows you to plot the graph N.1 where three areas are highlighted, each of which is characterized by certain working conditions that in terms of smoothness of the surfaces in contact allow us to make evaluations based on experimental results previously obtained. A good lubrication is always necessary, possibly with oil. With little or no lubrication working condition may vary greatly.

Graph N° 1 – Sliding condition for Bronze



Area A : area A is enclosed by the limit p•Vst = 21 [N/mm^2 • m/min]
In this area, the operation is in the best conditions.
"Continuous service" is possible as the amount of heat produced within these limits p • Vst is pretty low. Therefore the life of the nut is very good.

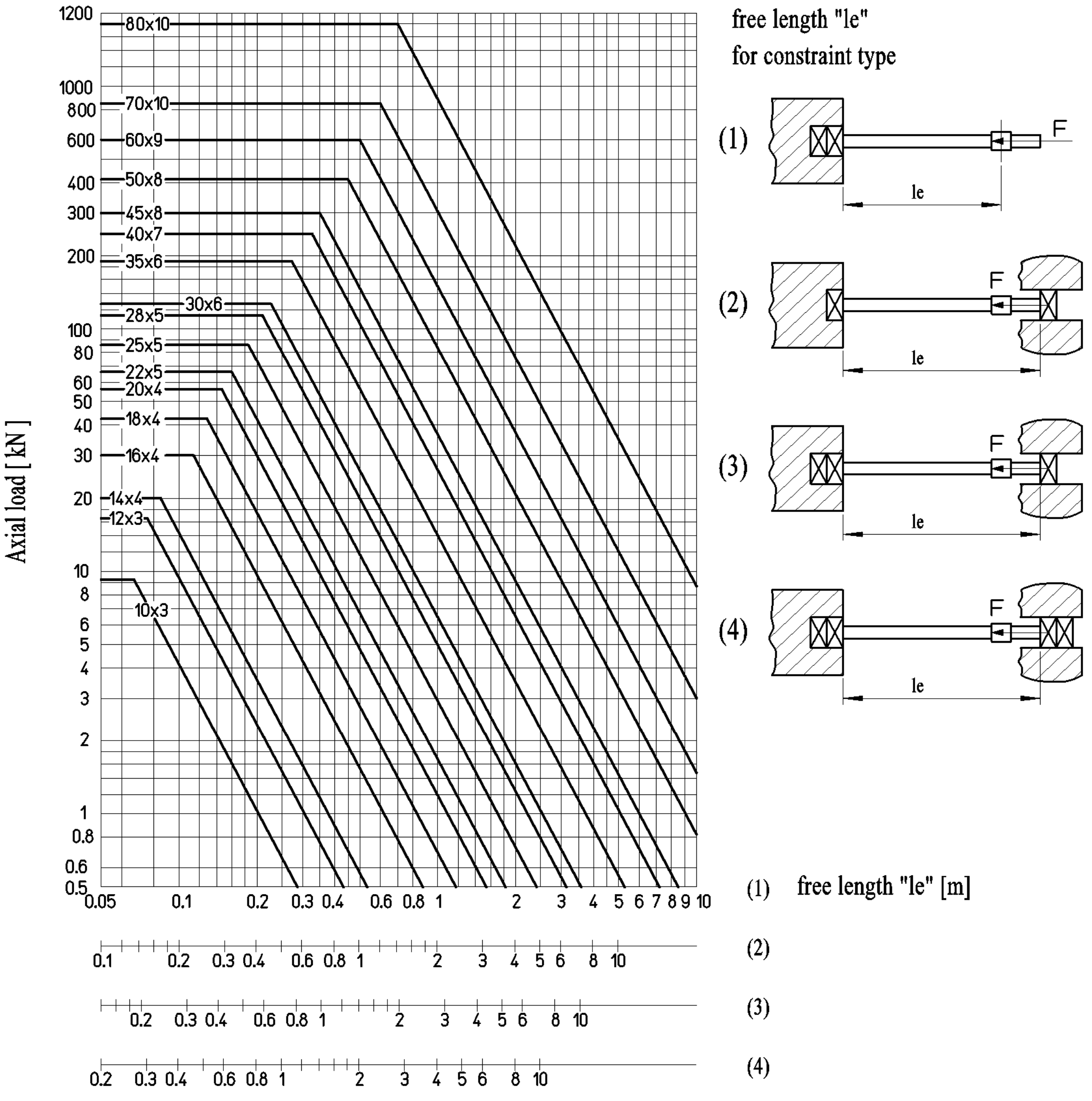
Area B : area B is enclosed by the limit p•Vst = 80 [N/mm^2 • m/min]
In this area, the operation is in more severe conditions.
Sliding conditions are as such that a steady lubrication is required to contain the erosion of the bronze so you still have good values of the life of the nut. “Continuous service” is possible for limited period only as the amount of heat produced is as such to produce an overheating of the nut, also depending from the real amount of oil used as this, lubrication apart, helps reducing heat.
Life of the nut is however limited.

Area C : area C is enclosed by the limit p•Vst = 250 [N/mm^2 • m/min]
In this area, the operation is in very heavy conditions.
With these values of p•Vst “continuous service” is certainly not possible. Even with good lubrication we face to a great overheating and a very quick nut wear off because the fiction between the surfaces in contact is as such to cause a rapid corrosion of the nut.

Critical Axial Load (Peak Load)

When there are compression loaded screws allowance must be made for limitations due to peak load to avoid screw bending due to excessive axial compression load. Admissible axial load depends on the core diameter (d3) of the screw, end constraints (bearings) and free length 'le'.
Regarding the values given in graph no. 6, allow a minimum safety factor ≥ 2.

Graph no. 6 - Peak Load



Example: find the admissible axial load of a Tr 30x6 screw 3000 mm long with constraint conditions as in drawing 4.
From graph 6 Take Fmax = 11 kN with safety factor of 2 and assume Fadm = 11/2 = 5.5 kN.

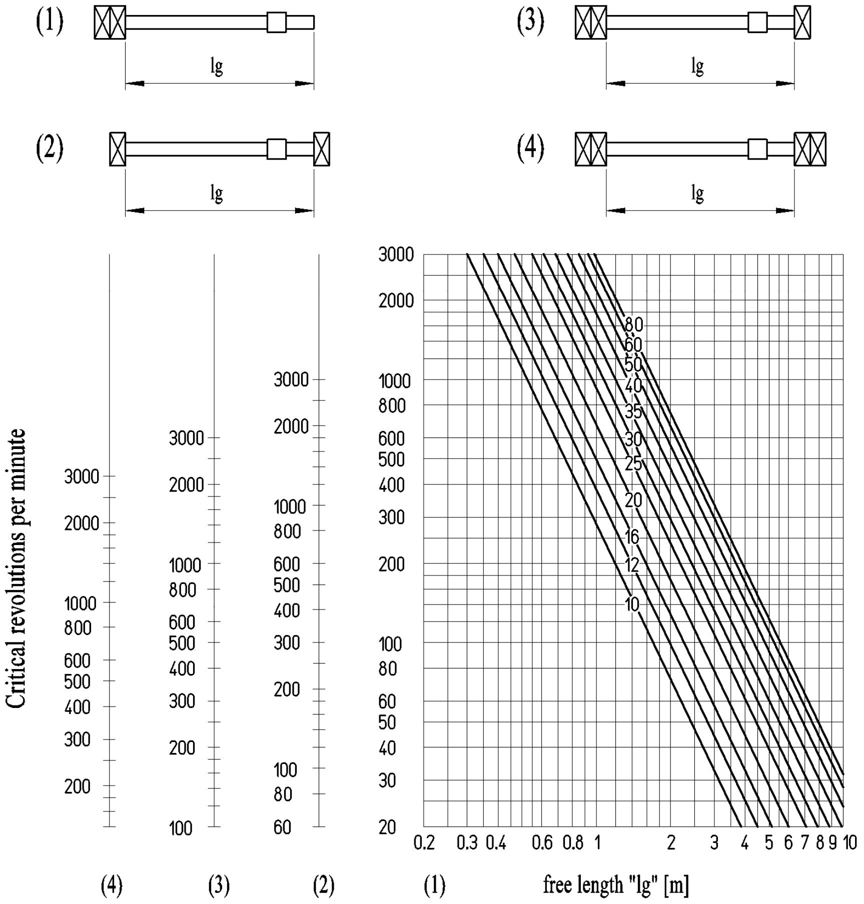
CONTI

Critical revolutions per minute

The critical revolutions per minute is the rotating speed at which screw vibrations appear. This rotation speed must never be reached because the vibrations cause serious operating irregularities. Critical rpm depend on screw diameter, end constraints (bearings), free length "lg" and from the assembly accuracy.
For values shown in Graph 7 assume a minimum safety factor related to the assembly accuracy as per the following chart:

Chart n°3 Assembly accuracy coefficient:		
Assembly accuracy	**Conditions**	**Safety coefficient**
Good assembly accuracy: - Nut alignment to screw within 0.05mm	Bearing and nut seats obtained from CNC lathe onto an already finished structure.	1.3 – 1.6
Average assembly accuracy: - Nut alignment to screw within 0.10mm	Bearing and nut seats processed on parts which are then assembled together. Alignments are checked by comparators with estreme care after mounting.	1.7 – 2.5
Low assembly accuracy: - Nut alignment to screw within 0.25mm	Bearing and nut seats processed on parts which are then assembled or welded together. Alignments are checked by comparators after mounting.	2.6 – 4.5

Graph no. 7 – Critical rpm



Example: find the critical rpm of a screw Tr 40x7 length 3000 mm with constraint conditions as in drawing 3 with average assembly accuracy. Graph 7 gives critical rotation speed ≅ 1000 rpm
From chart n°3 we calcolate the Safety coefficient = 2.2.
We can reach the working speed at a maximum round speed of: n. max = 1000/2.2 = 454 rpm.