Design and Production of a Knuckle Boom Crane

Henrik Borge

Henrik Veholmen Erdal

Benjamin Aardal Færevaag

Bachelor’s thesis in Mechanical engineering

Bergen, Norway 2021

Høgskulen på Vestlandet

Western Norway University of Applied Sciences

Henrik Borge

Henrik Veholmen Erdal

Benjamin Aardal Færevaag

Department of Mechanical- and Marine Engineering

Western Norway University of Applied Sciences

NO-5063 Bergen, Norway

IMM 2021-M04

Høgskulen på Vestlandet

Fakultet for Ingeniør- og Naturvitenskap

Institutt for maskin- og marinfag

Inndalsveien 28

NO-5063 Bergen, Norge

Norsk tittel:	Design og produksjon av en “Knuckle Boom” Kran
Author(s), student number:	Henrik Borge, h578112
	Henrik Veholmen Erdal, h578138
	Benjamin Aardal Færevaag, h578133
Study program:	Mechanical engineering
Date:	June 2021
Report number:	IMM 2021-M04
Supervisor at HVL:	Thorstein R. Rykkje
Assigned by:	Høgskulen på Vestlandet
Contact person:	Thorstein R. Rykkje
Antall filer levert digitalt:	1

Preface

This bachelor's thesis marks the end of the bachelor's program of mechanical engineering at the Department of Mechanical and Marine Engineering at Western Norway University of Applied Sciences (WNUAS) in Bergen. The project has been conducted under the supervision of doctoral student Thorstein R. Rykkje, to whom we would like to thank for supporting us throughout the project.

We would also like to show our gratitude to over engineer Harald Moen and to engineer Frode. W. Jansen for supervising the assemble of the crane and helping with tackling various obstacles that came up on the way.

Henrik Borge
Henrik Veholmen Erdal
Benjamin Aardal Færevaag
Bergen, 25.05.2021

Abstract

In this thesis, an automatic, scaled knuckle boom crane has been designed and produced. The crane is able to operate from a scaled model boat. This project is made in collaboration with doctoral student Thorstein R. Rykkje. The goal was to produce a crane that can be remotely controlled and able to be automated so that the crane counteracts the dynamic forces it is subjected to, such that the load is at rest. The crane must have the possibility to be used in further testing and research on Smart Ships and cranes at WNUAS.

The crane has been developed in Creo [1], 3D-printed and then built and constructed with needed equipment manually. Different designs have been developed, all with different functionalities and looks to achieve the best fitted product. The requirements that were needed led to a lot of adjustments during the process. The 3D-printer’s limitations in both size and margin of error needed to be taken into count. Subsequently the crane consists of multiple parts, printed with polylactic acid (PLA).

The product is equipped with three servo motors that controls its movement, and a data acquisition system to capture the behavior off the crane due to physical forces, called (Arduino UNO Wi-Fi REV2). The arduino circuit card consists of a built-in gyro that tracks angular velocity. An encoder is mounted to each motor controlling the base rotation to keep track of the angles. Lastly ta winch with a hook is mounted to the crane.

Sammendrag

I denne bacheloroppgaven ble en “knuckle-boom” kran utviklet. Denne kranen er laget i samarbeid med doktorgradsstudent Thorstein R. Rykkje. Målet var å produsere en kran som kan fjernstyres og automatiseres, slik at kranen motvirker de dynamiske kreftene den blir utsatt for, med den hensikt at lasten står i ro. Kranen skal også kunne brukes i videre testing og forskning på Smarte Skip og kraner ved HVL.

Kranen ble utviklet i Creo [1], 3D-printet og så bygget og konstruert med nødvendig utstyr. Flere forskjellige design ble produsert, alle med ulik funksjonalitet og utseende. Til slutt ble det best egnete designet valgt. 3D-printeren hadde noen begrensninger med tanke på størrelse og feilmargin som måtte bli tatt hensyn til. Designet som ble valgt består av flere deler som alle ble printet i materialet polymelkesyre (PLA).

Kranen er utstyrt med tre servomotorer som står for kranens bevegelser og ett datasamlingssystem (Arduino UNO Wi-Fi REV2) som fanger opp kranens bevegelse i henhold til de fysiske kreftene den er utsatt for. En vinsj med krok er montert på for å løfte lasten. Arduino kortet har en innebygget gyro som måler vinkelhastighet. I tillegg er det montert rotasjons enkodere i alle tre ledd for å holde kontroll på vinklene til de forskjellige delene til enhver tid.

Table of contents

Nomenclature

Θ_T	Angle of Tower.
Θ_{MB}	Angle of Main Boom.
Θ_B	Angle of Boom.
m	Mass
L	Length of joint
$\dot{\omega}$	Angular velocity matrix
ω	Angular velocity vector
s	Time
H_c	Angular momentum
M_c	Moment of inertia
$T_{Motor\ Torque}$	Torque moment
J_c	3x3 Mass moment of inertia matrix

1 Introduction

1.1 Engineering Background

Knuckle boom cranes are often used on fixed platforms and on vessels with big deck areas. They are suitable to operate at close distances. The knuckle boom crane is a pedestal-mounted, slew-bearing crane with a joint in the middle of the boom.

Vessels are used in offshore operations where harsh weather conditions often occur. This means the knuckle boom crane must be able to maneuver in the worst weather scenarios. As well as being solid and strong, the crane must be able to keep the load stable despite the motion of the ship. A control system would sense motion in the ship due to waves and wind, before the crane moves accordingly to its given position.

A scaled model of a knuckle boom crane will be designed and constructed. The project is building on earlier calculations, and hopefully this project will help future engineers. This could prove beneficial in many scenarios at sea for examples the deployment of a load close to the platform during high waves or when loading from one vessel to another.

1.2 Problem Description

The goal of this thesis is to make a remote-controlled knuckle boom crane. The crane should have the opportunity to get automated and further used in testing and research on Smart Ships and cranes at WNUAS. The goal is to develop a crane that will be able to counteract the dynamic forces which it is exposed to, with the purpose of keeping the load at rest.

The crane needs to be calculated and tested in Ansys (Solidworks) to know its exact strength and dimensions. With that information it can is possible to calculate the maximum load and torque of the crane, as well as where the motors and sensors can be placed.

Together with Thorstein Rykkje, an automized model of a scaled knuckle boom crane will be developed. This contains 3D-modeling, 3D-printing, mounting, programming, and testing. When the crane is built, motors and sensors will be fitted such that the crane can react and respond to its given signals.

1.3 Approach

The execution of this project has mainly followed the current approach:

1. Calculations and early roughly drafts
2. 3D-modeling of the crane
3. Buying all necessary equipment like motors, sensors, bearings etc.…
4. 3D-printing, building and testing the crane
5. Continual Improvement of the crane after each result

The first months were difficult considering only one student having access to Creo [1] and Ansys. The Covid-19 restrictions made it challenging to meet and work together at school as well as receiving help with the software. Most of the work got done from home using Teams-meetings in the first months. When the restrictions were eased, we met more often to make the goal possible to reach.

Figure 1 - Knuckle Boom Crane

2 Method

It was important to strictly follow a well-thought-out plan of execution to reach the finish line. This contained operating within both theoretical and experimental operations. With this being the third bachelor group working on the knuckle boom crane project, a lot of secondary data was used within theoretical dynamic calculations. [2] [3] [4] [5] This year's bachelor group is the first to practically build a crane, which means a more experimental project.

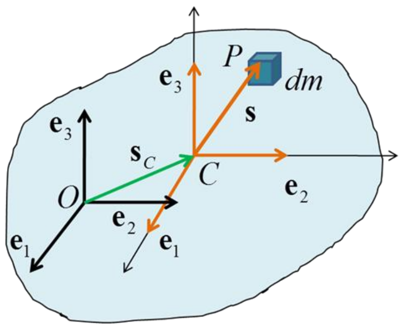
2.1 Theoretical approach

Since former groups already have done similar calculations and mathematical procedures, only specific experiments concerning this exact thesis would be necessary.

The most significant issue that needed to be addressed was the strength-calculations. To get familiar with the crane´s required strength, the biggest moment and torque needed to be revealed. An early draft of the crane was drawn for the purpose of having approximations of the body's dimensions. With these approximated lengths, widths and thicknesses, the strength-calculations could be completed at an early stage.

Dynamic theory was used to calculate the moment of the knuckle boom crane. There were three bodies that needed to be included in the prognosis. The formula was calculated using cylindrical approximations throughout the parallel axis theorem.

$$J_O = m\begin{bmatrix} h^2 & 0 & 0 \\ 0 & h^2 & 0 \\ + & 0 & 0 \end{bmatrix} + \begin{bmatrix} \frac{1}{12}m\left(3r^2+h\right) & 0 & 0 \\ 0 & \frac{1}{12}m\left(3r^2+h\right) & 0 \\ 0 & 0 & \frac{mr^2}{2} \end{bmatrix}$$



J_o is known.

We know.

$$\dot{\mathbf{H}}_C = \mathbf{M}_C(t)$$

To get $M_c(t)$, $H_c(t)$ has to be known.

$$\mathbf{H}_C(t) = \mathbf{e}(t) J_C \, \omega(t)$$

Derivate it and get.

$$\dot{\mathbf{H}}_C(t) = \dot{\mathbf{e}}(t) J_C \, \omega(t) + \mathbf{e}(t) J_C \, \dot{\omega}(t)$$

Which makes.

$$\dot{\mathbf{H}}_C(t) = \mathbf{e}(t)\tilde{\omega}(t)J_C\,\omega(t) + \mathbf{e}(t)J_C\,\dot{\omega}(t)$$

Which then again makes.

$$\dot{\mathbf{H}}_C(t) = \mathbf{e}(t)\left(\tilde{\omega}(t)J_C\,\omega(t) + J_C\,\dot{\omega}(t)\right)$$

Which means the moment is.

$$\mathbf{e}(t)\left(\tilde{\omega}(t)J_C\,\omega(t) + J_C\,\dot{\omega}(t)\right) = \mathbf{M}_C(t)$$

Put into Euler's equation.

$$\mathbf{e}(t)\begin{pmatrix} M_{O1} \\ M_{O1} \\ M_{O1} \end{pmatrix} = \mathbf{e}(t)\left(\begin{bmatrix} 0 & -\omega_3(t) & \omega_2(t) \\ \omega_3(t) & 0 & -\omega_1(t) \\ \omega_2(t) & \omega_1(t) & 0 \end{bmatrix}\begin{bmatrix} J_{11} & 0 & 0 \\ 0 & J_{22} & 0 \\ 0 & 0 & J_{33} \end{bmatrix}\begin{pmatrix} \omega_1(t) \\ \omega_2(t) \\ \omega_3(t) \end{pmatrix} + \begin{bmatrix} J_{11} & 0 & 0 \\ 0 & J_{22} & 0 \\ 0 & 0 & J_{33} \end{bmatrix}\begin{pmatrix} \dot{\omega}_1(t) \\ \dot{\omega}_2(t) \\ \dot{\omega}_3(t) \end{pmatrix}\right)$$

Assume there is only rotation about the first axis.

$$\mathbf{e}(t)\begin{pmatrix} M_{O1} \\ M_{O1} \\ M_{O1} \end{pmatrix} = \mathbf{e}(t)\left(\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\omega_1(t) \\ 0 & \omega_1(t) & 0 \end{bmatrix}\begin{bmatrix} J_{11} & 0 & 0 \\ 0 & J_{22} & 0 \\ 0 & 0 & J_{33} \end{bmatrix}\begin{pmatrix} \omega_1(t) \\ 0 \\ 0 \end{pmatrix} + \begin{bmatrix} J_{11} & 0 & 0 \\ 0 & J_{22} & 0 \\ 0 & 0 & J_{33} \end{bmatrix}\begin{pmatrix} \dot{\omega}_1(t) \\ 0 \\ 0 \end{pmatrix}\right)$$

Multiply.

$$\mathbf{e}(t)\begin{pmatrix} M_{O1} \\ M_{O1} \\ M_{O1} \end{pmatrix} = \mathbf{e}(t)\left(\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\omega_1(t) \\ 0 & \omega_1(t) & 0 \end{bmatrix}\begin{pmatrix} J_{11}\omega_1(t) \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} J_{11}\dot{\omega}_1(t) \\ 0 \\ 0 \end{pmatrix}\right)$$

One more time.

$$\mathbf{e}(t)\begin{pmatrix} M_{O1} \\ M_{O1} \\ M_{O1} \end{pmatrix} = \mathbf{e}(t)\left(\begin{pmatrix} J_{11}\dot{\omega}_1(t) \\ 0 \\ 0 \end{pmatrix}\right)$$

Throw away the frame.

$$M_{O1} = J_{11}\dot{\omega}_1(t)$$

Which equals.

$$T_{Motor\_Torque} = \frac{1}{3}mL^2\dot{\omega}_1(t)$$

This formula was used on each joint to calculate their needed motor torque. To know the motor torque, angular velocity had to be found. Doing an experiment in the wave pool, helped us figure out the angular velocity during worst case scenarios. This means setting a frequency where the waves are bigger than real life scenarios. This assured that the safety factor was big enough, as well as the crane was guaranteed to be strong- and agile enough. Calculations with an overemphasized safety factor made sure that the largest moment was substantial. This minimized the risk of failure while simultaneously ensuring suitable motors.

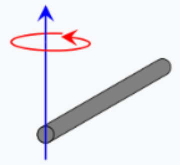
Rod of length *L* and mass *m*, rotating about one end. This expression assumes that the rod is an infinitely thin (but rigid) wire. This is also a special case of the thin rectangular plate with axis of rotation at the end of the plate, with *h* = *L* and *w* = 0.		$I_{end} = \frac{1}{3}mL^2$ [1]

Figure 2 - *Moment of inertia*

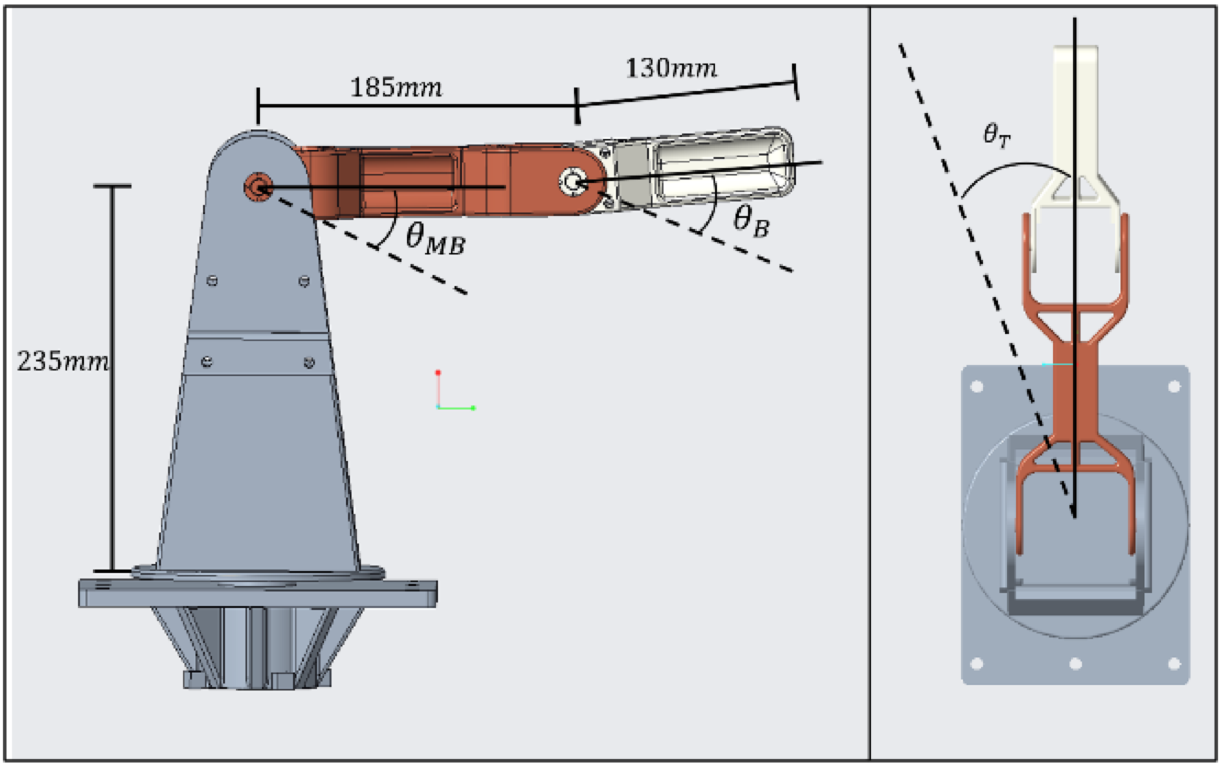


Figure 3 - Dynamic Modeling of the Crane

2.2 Experimental Approach

This thesis was mainly practical and contained different experiments where lab-exercises was influential. This chapter consist of various experiments listed in several subchapters below. To build a knuckle boom crane, experiments throughout the whole project was essential. Marine-labs, 3D-modeling, programming, 3D-printing and testing were the main experiments to reach the goal of this project.

2.2.1 Angular Velocity Experiment

First exercise was to find the maximum angular velocity to convert the crane's required force, in order to buy the correct motors. This experiment took place in the wave pool at the marine lab at WNUAS. An already built scaled boat was used as well as an app called "Physics toolbox suite".

The boat needed to contain enough mass so the water surface aligned with the waterline on the boat (as you can see in figure 4). 37kg of mass was put inside the boat. The smartphone was put into the boat which led to the app calculating all angular velocities. It is important that the smartphone is steady throughout the boat´s displacements to get reliable and correct numbers.

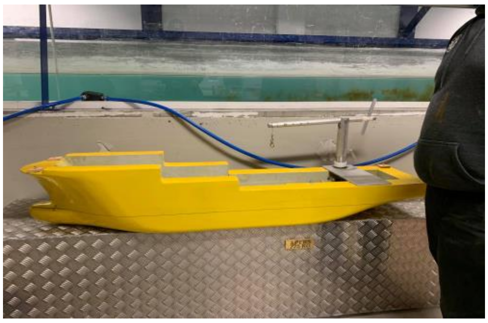


Figure 4 - Waterline on Ship

The wave pool was set to different frequencies to know the boats reactions. An exaggerated frequency of the waves was set to know the biggest angular velocity. As you can see in figure 5, the boat was exposed to "rough weather".

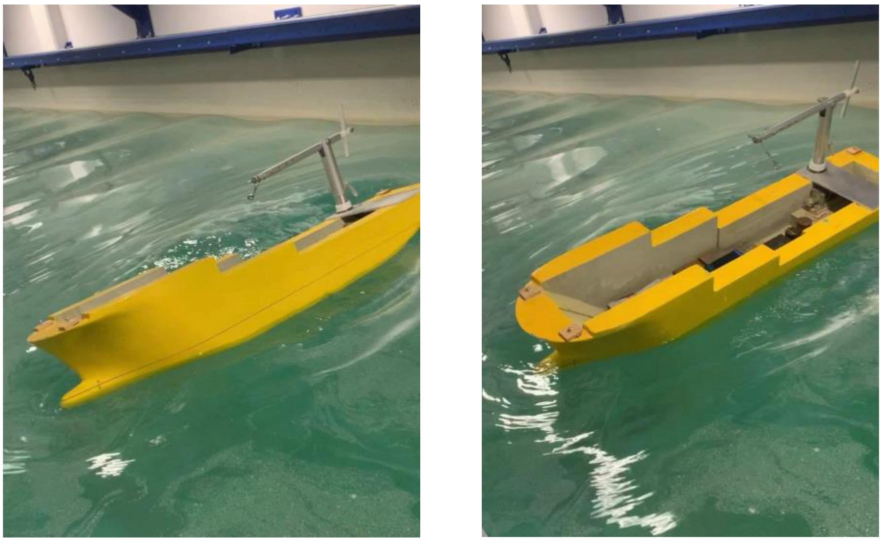


Figure 5 - *Miniaturized Ship in Marine Lab*

2.2.2 Design

The design process was a time-consuming part of the project throughout the first months. During the design process a lot of different drafts was considered. After an examination of multiple drafts, the most qualified version was kept.

An early draft got completed so the strength and dimensions could be estimated. This made it easier to order motors and other needed equipment as early as possible. With the global pandemic and the risk of long delivery time with such distinct equipment, an effective start was crucial.

Furthermore, an idea of how the crane would look was fundamental. A lot of consideration had to be done to design it as strong as possible and to make it able to fit all added attachments. There was a lot of requirements to review to make this crane a reality.

These requirements consisted of:

- The tower having to be able to turn (150°).
- Both added booms needed to be capable of moving at least 120° degrees.
- Three servomotors must fit into the crane.
- One winch must be fitted.
- Sensors must be able to be attached.
- Ball bearings, bolts and nuts need to fit.

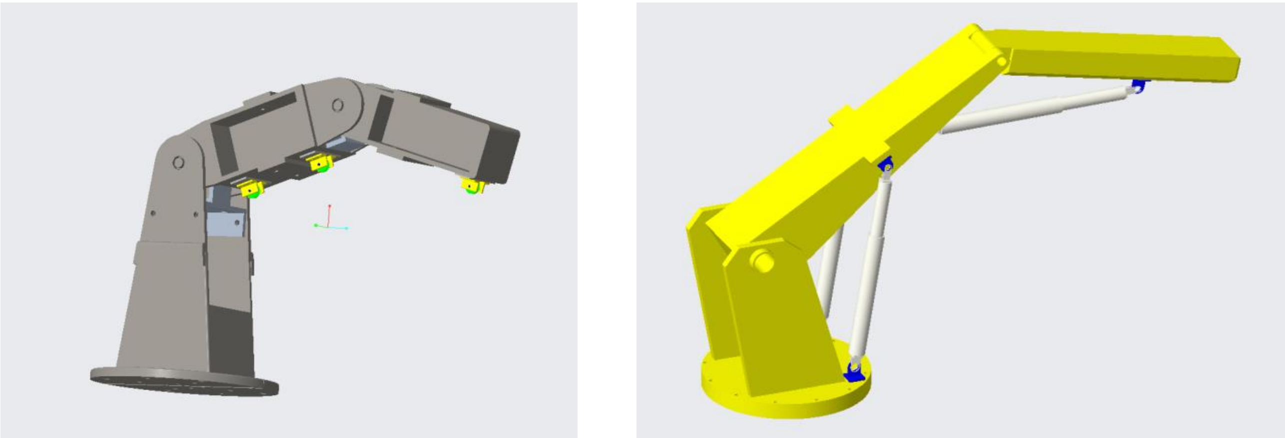


Figure 6 - 3D-modeling Drafts

2.2.3 Ansys Solidworks

Ansys Solidworks [6] is an engineering software that enables students and engineers to calculate more accurate stresses, strains, forces, and other design parameters. Students and engineers consider this to be a valuable tool for the design process, as it visualizes strengths and weaknesses. One of the most essential benefits this software provide is cost reduction. This software will be used as one of the tools for the project.

2.2.4 3D-printing

To 3D-print a product is an advanced procedure that takes a huge amount of time. One part can take up to several days to finish, which means that it is important to avoid unnecessary errors. Although the production material PLA is relatively cheap, it can be time-consuming to slip up. Therefore, it is of great importance that everything is thought of before using the printer.

The parts were printed with a material called polylactic acid (PLA). With the goal of this thesis being to build a scaled crane that can be automated and able to lift small loads, it turned out that PLA performed perfectly.

The 3D-printer's that were used is called Flashforge [7] [8]. This is an advanced machine that is extremely exact and precise. These machines print upwards, which means that printable parts are a bit limited. The 3D-printer can only add support material to a certain degree to help the construction stand. There will be uncertainties, especially in connection points and tolerances, therefore starting with only the tolerances can be smart considering time and money. In figure 7, the test prints were finished, and the margin of errors was known. With that information, bigger, more long-lasting prints could take place.

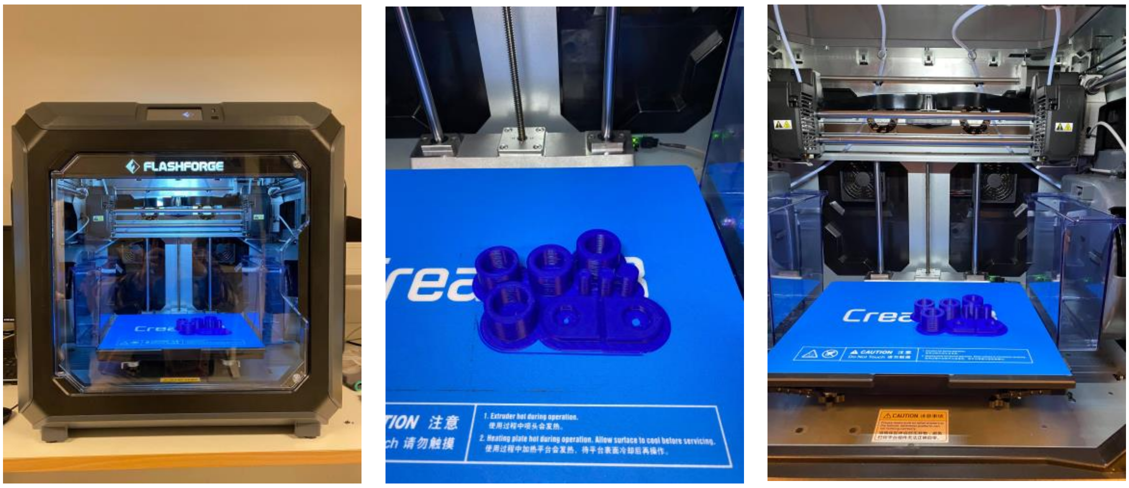


Figure 7 - 3D-Printer

2.2.4 Programming and control-system

At WUNAS, the machine engineering curriculum is heavily focused on physics and computational solid mechanics. Since programming is not included in the bachelor's degree, doctoral degree student Thorstein helped with all programming software to make the crane functional. Thorstein has experience within programming language C/C++ and Arduino and will be helpful in this procedure.

2.2.5 Testing and future work

The goal for the future is that the crane can operate totally independently. It is expected that this goal is a bit optimistic to achieve in six months, although work will be continuously done to try make it happen. A slight problem is that the lab is being reserved for marine students. Hopefully, there will come an opportunity to make it happen so that at least some kind of testing will be possible.

However, if this is not possible, another solution for testing during these six months is to design a jig that is driven by a fourth servo motor. The jig will be driven independently from the crane. This way the waves from the pool can be simulated, and testing can be done on the crane to counteract the movement from the jig.

While the final goal may take a bit longer to achieve, finishing the crane and running it by a controller device will be very helpful for future work. This would make it easier for future groups and Thorstein to reach the main goal where the crane counteracts all dynamic forces it is exposed to, without using a controller.

2.3 Source of errors

Unexpected errors will always occur during big projects, that is why it is important to start the process as early as possible. Small errors in advanced procedures like 3D-modeling and 3D-printing can cause big problems. This is usually a result of human errors where calculations are not performed correctly.

Since PLA is not an isentropic material, it became challenging to compute the strengths-calculations in Ansys. This led to using a material with similar properties to PLA. To solve this uncertainty, a material with weaker properties were used during the Ansys simulations.

Other errors like uncertainties on the 3D-printer must be taken into consideration. Therefore, concentration during the design process, is essential. This problem got solved by testing uncertainties on small parts early in the process. These tests showed that the 3D-printer had a margin of error up to 2mm as well as showing how the support material behaved. When this information was gathered, the uncertainties and how to position the parts were known.

3 Results

Throughout this project, a variety of test results has been gathered to achieve the best possible outcome. Both theoretical and experimental results is presented here.

3.1 Results from angular velocity experiment

First action that was performed in the thesis was to place a smartphone into a model boat at WNUAS marine lab. The app Physics toolbox suite was turned on and was able to calculate the angular velocity of the boat in all three axes. With the angular velocity known, the angular acceleration is possible to calculate. The goal was to compute the angular momentum, which was found from the acceleration. By knowing this, suitable servo motors were bought. The goal with this experiment was to find servo motors that could endure the angular momentum.

The moment formula from the dynamic theory, made it possible to calculate the crane´s largest momentum. The mass (m), lengths (L), radius (r) and the angular velocities (ῶ) was computed from the first draft. The first draft was constructed as solid cylindrical joints. Solid structure approximation was used to ensure that the servomotors was more than strong enough. From the approximated dimensions, the largest moment was calculated. The strengths needed were now known, thus competent servo motors were ordered.

$$T_{Motor\_Torque} = \frac{1}{3} mL^2 \dot{\omega}_1(t)$$

Calculation results:

	Length (m)	Mass (kg)	Angular Velocity (rad/s^2)	Radius (m)
Tower	0,23	0,7	5	0,028
Main Boom	0,28	0,7	5	0,025
Boom	0,3	0,5	5	0,021

Tabell 1 - Properties of joints

Angular momentum:

$$\dot{\omega} = \frac{\omega}{s} = \frac{0{,}01\,\frac{rad}{s}}{0{,}002s} = 5\,\frac{rad}{s^2}$$

Tower:

$$T_{Motor\ Torque} = \frac{1}{3} * 0{,}7kg * 0{,}23^2 m * 5\,\frac{rad}{s^2} = 0{,}08\ Nm$$

Boom:

$$T_{Motor\ Torque} = \frac{1}{3} * 0{,}5kg * 0{,}3^2 m * 5\,\frac{rad}{s^2} = 0{,}075\text{Nm}$$

Main boom + boom:

$T_{Motor\ Torque} = \frac{1}{3} * 1{,}2kg * 0{,}58^2 m * 5\frac{rad}{s^2} = 0{,}67\ Nm$

From the approximations, the largest moment was 0,67 Nm. With a safety factor set to two, the servo motors had to be strong enough for at least 1,34 Nm.

3.2 Design result

As Mechanical Engineer students the main goal was to design a functional knuckle boom crane using prior educational experience. After a lot of back and forth, the final design was finished, and the result is shown in figure 8.

The knuckle boom crane mainly consists of three bodies (tower, main boom, and knuckle boom). The construction also consists of a base which is the connection point between the crane and ship. Each of the bodies are driven by three electrical 6V servo motors with an operation range of 180 degrees and can take up to 0,96 Nm. Considering that the servo motors were too weak, adjustments on the design had to be completed. They are mounted at different joints between the bodies. One between the Boom and main boom, second one between tower and main boom, and the third is fitted on the bottom of the tower. The motors are powered by a rechargeable 6V, 3.3Ah battery pack. An electrical winch is also mounted in the middle of the tower with the hook pulled through two pulleys. The winch is powered by a 7.4V 1.1Ah battery pack.

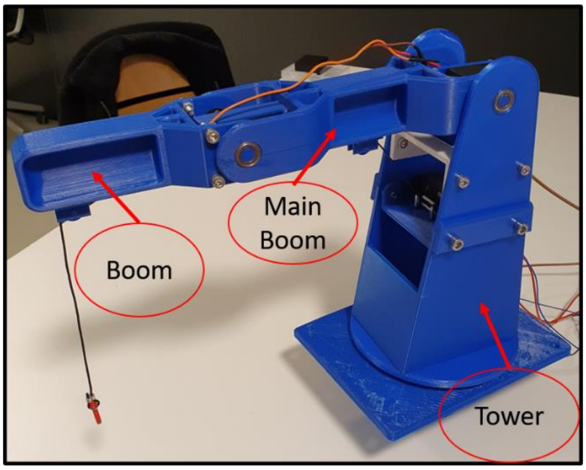


Figure 8 - Final Product

The crane is designed such that each body is restricted to one known axis of rotation. The motor placed at the bottom of the tower rotates the crane around the relative vertical axis of the ship, and the two remaining motors rotates both main boom and knuckle boom around a horizontal axis relative to the tower. This gives the crane full 3D-range of motion. The direction of the rotations and dimensions of the bodies are shown in figure 3.

The bodies where 3D-printed using polylactic acid (PLA and PLA+) to reduce weight and to encourage fast and easy production. A great advantage with 3D-printing is that mistakes are easily fixed compared to steel, and with PLA being more than strong enough having a tensile strength up to 66 Mpa [9], this was an obvious choice. A great advantage with 3D-printing is that mistakes are easily fixed compared to metal components. To establish low friction and smooth relative movement between the bodies, all joints are equipped with two ball bearings each on either side.

To facilitate remote operation of the motors, the crane is fitted with Arduino UNO WI-FI rev2 [7]. This card sends angular orientation signals in the form of PWM to the motors. The arduino card is described later in the equipment section.

3.3 Equipment

To make the crane functional different equipment was added. This section describes which decisions that was chosen in order to achieve a functional knuckle boom crane.

3.3.1 Battery pack

As a power option, a battery pack is added. The first battery pack purchased is a 6V 3,3Ah battery pack. 6V is sufficient for the Arduino card and the three servo motors, but the winch needed a voltage above 6.5V. The solution was to buy a 7.4V 1,1Ah battery pack, such that the winch was satisfied. However, the servo motors operating voltage was between 4,5-6V. To sum up, the motors are using a 6V battery pack and the winch are using a 7.4V battery pack.

3.3.2 Bearings

The decision fell on using ball bearings which were mounted to each joint to make frictionless rotations possible. As a result of uncertainties in the 3D-printer, the decisions on how to connect the bearings was hard. The students decided to use interference fit to a certain extent. To ensure that any bearings was not well-attached enough, glue were used.

Two bearings used at the bottom of the tower to create a frictionless rotation around the relative vertical axis: [10]

- 2x Deep groove ball bearings 628/9-2Z

Four bearings are attached at the booms to create a frictionless rotation around the horizontal axis relative to the tower: [10]

- 4x Deep groove ball bearings 61804-2RS1

Figure 9 - Bearing in the foot of the crane

3.3.3 Bolts and nuts

The preferred connection methods are bolts and nuts, supplied with washers where it is needed. Strength was never an issue on this project as stainless steel is so much stronger than PLA. Therefor accessibility, cost and design were the main reasons behind bolt and nut choices.

Bolt dimensions (everything in stainless steel):

- M2: Connect winch to crane

- M4: All crane parts connections

- M5: Connect tower to foot

- M8: Servomotor

- Woodscrew 2,5mm: Servomotors

Figure 10 - Bolts connecting the parts

3.3.4 Sensors

Broadcom Incremental Encoder 500 ppr, AEDB-9140-A13 is an optical encoder sensor used for feedback. The continuous information received by the sensor tells where the rotational links are located. In the picture below we can see a circular disc with small holes in the outer diameter. This disk is located around a shaft, such that they rotate together. The counter is mounted on the rigid wall of the crane, such that the disc rotates inside the gap of the counter. The counter than sends light towards the disk from one side. As the light only reach the other side of the gap when going through one of the holes in the disk, it can count the number of light signals and calculate how many degrees the disk has moved from its zero position.

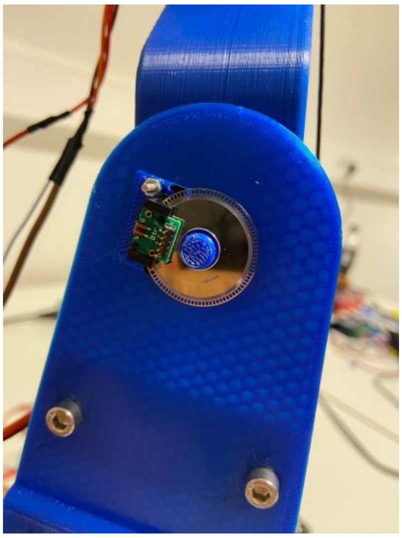


Figure 11 - Sensor connected to the crane

3.3.5 Winch

To make it possible for the crane to lift loads, a winch was attached. This is a small but powerful 1/10 scaled winch intended for model cars. With its impressive lifting capacity of 2-3 kg, the students found this one perfect for the crane:

- Yeah racing winch 0527

It has an operating voltage from 7 to 13 volts and runs on 1,1 amperes.

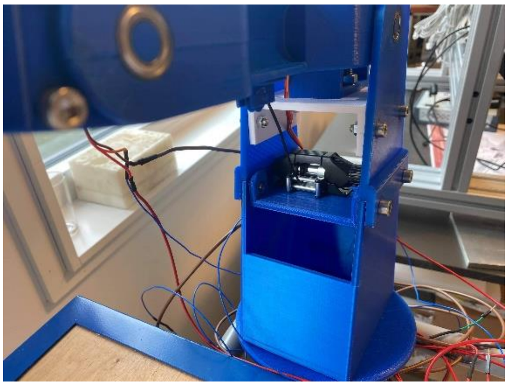


Figure 12 - Winch on crane

3.3.6 Arduino

To control the crane, an Arduino UNO Wi-Fi REV2 [11] circuit card was attached. This enables the controller to control every servo motor as well as the winch from one place. It will also make the system of connected wires more organized, such that a cleaner design is achieved. The circuit card is connected to a 3D-printed design connected to the lower part of the crane tower. With downloading the programmed codes into the card, enables it to control the crane with an Xbox One controller that sends signals to the Arduino circuit card via Wi-Fi.

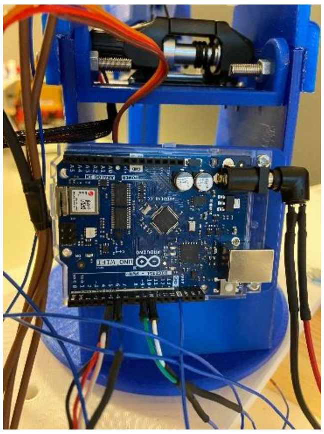


Figure 13 - Arduino card

3.3.7 Speed controller

This is a relay used to regulate direction and speed for the winch. It has a built-in system that cuts the power when the voltage is too low to drive the winch motor. Because of this application, it was discovered that the first battery was not fit for the operation of this crane. The speed controller that was decided to be added to the control-system was:

- HOBBYWING QUICRUN WP 1060 BRUSHED ESC

Figure 14 - Speed Controller

3.3.8 Workshop

A variety of tools was used in the campus workshop. The most regularly used tools for this project are wrenches, allen keys, torx, pliers, knives, files, sandpaper amongst others. Soldering was used to connect wires for the winch, servo motors and batteries. All these tools helped us making the crane more functional and easier to run.

3.4 Ansys simulations

Ansys was used regularly throughout the design process to make sure every part exceeded the minimum strength limits. Initially a safety factor of two or higher was desired. However, because PLA does not have isentropic properties, a much larger safety factor was preferred for most parts. Since it was not possible to find all properties needed for PLA, the most similar material to PLA was chosen in Ansys and used for approximating calculations. The material chosen was ABS, which is used on the same 3d-printers and are weaker than PLA, to ensure the cranes components are strong enough. [6]

All parts were separately simulated in Ansys to get approximated calculations of each joint. This made it possible to locate the weakest spots. The simulations were set to calculate the deformation and stress on each joint. The results of the Ansys simulations are presented in figure 15 to figure 23.

In figure 15 and 16 the force points downwards from where the initial pulley was thought to be. Each of the four holes are set to fixed supports.

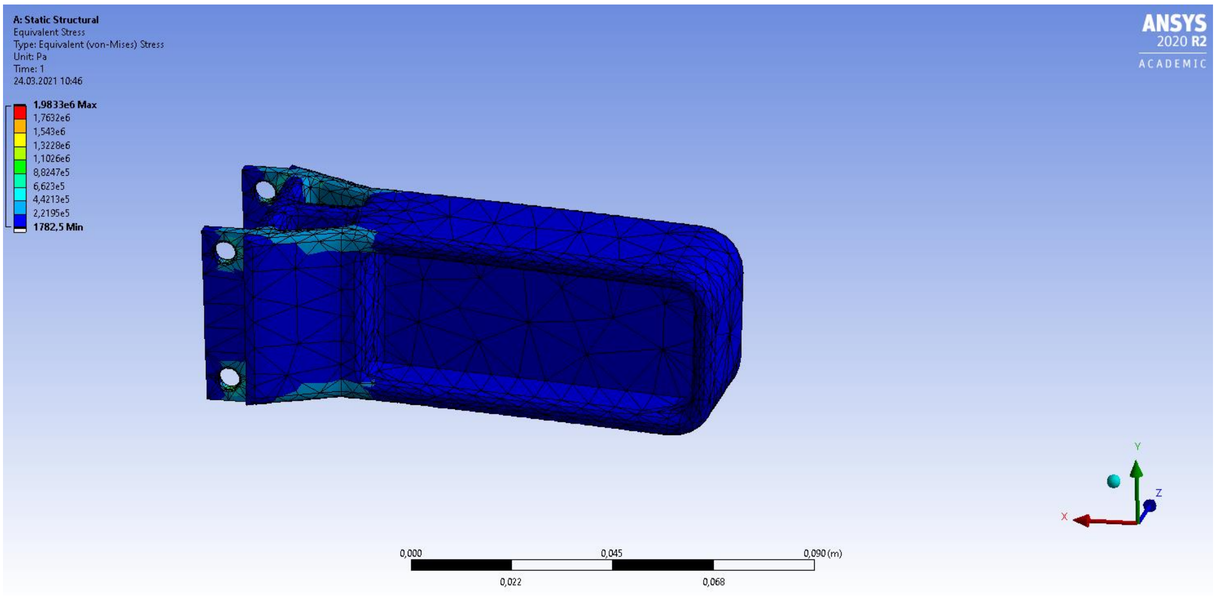


Figure 15 - Von Misses Stress, boom

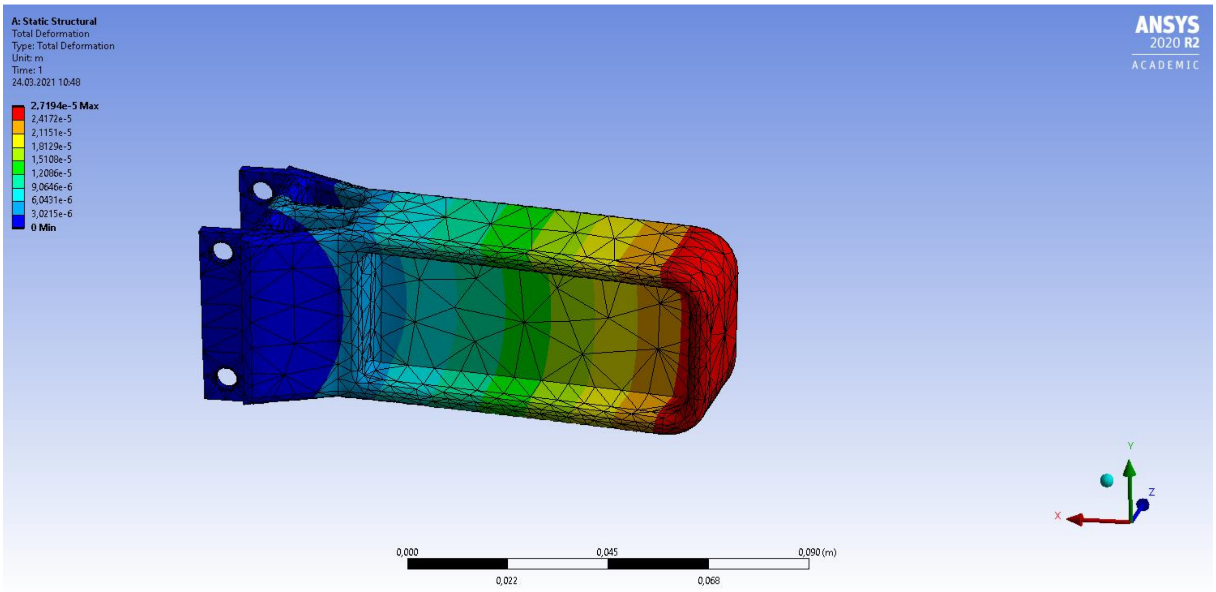


Figure 16 - Deformation, boom

The forces in figures 17 and 18 are placed inside each of the small holes around the shaft where the servo motor is connected to create a moment.

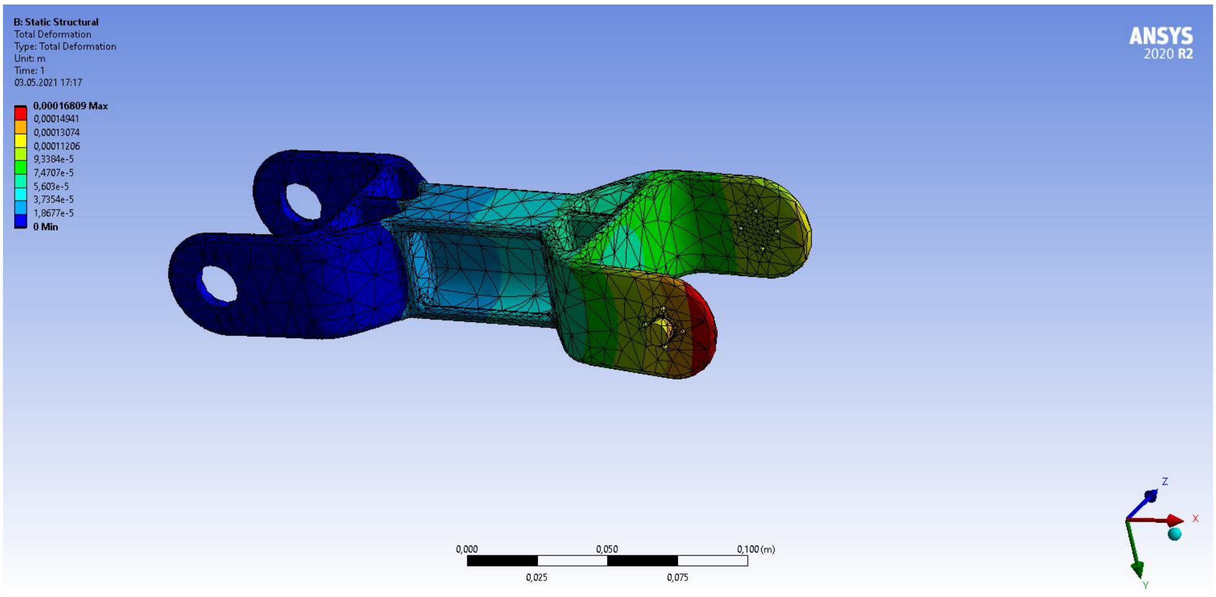


Figure 17 - Deformation, main boom

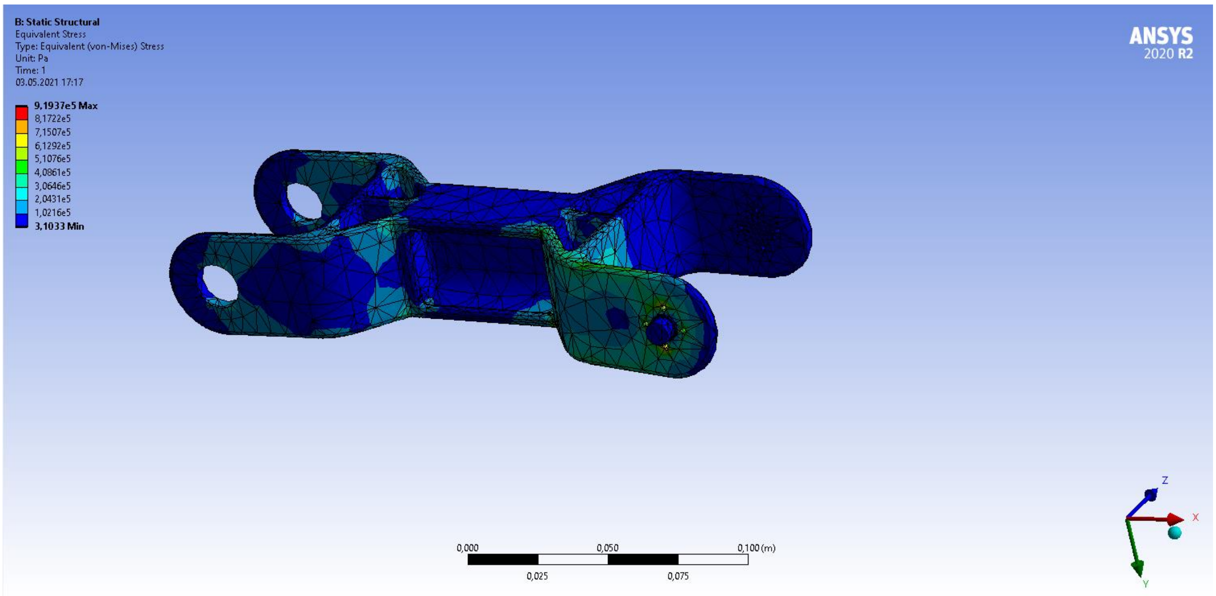


Figure 18 - Von misses stress, main boom

In figure 19 and 20 the moment exceeded from the weight of the arms and motors are set as shown in figure 19.

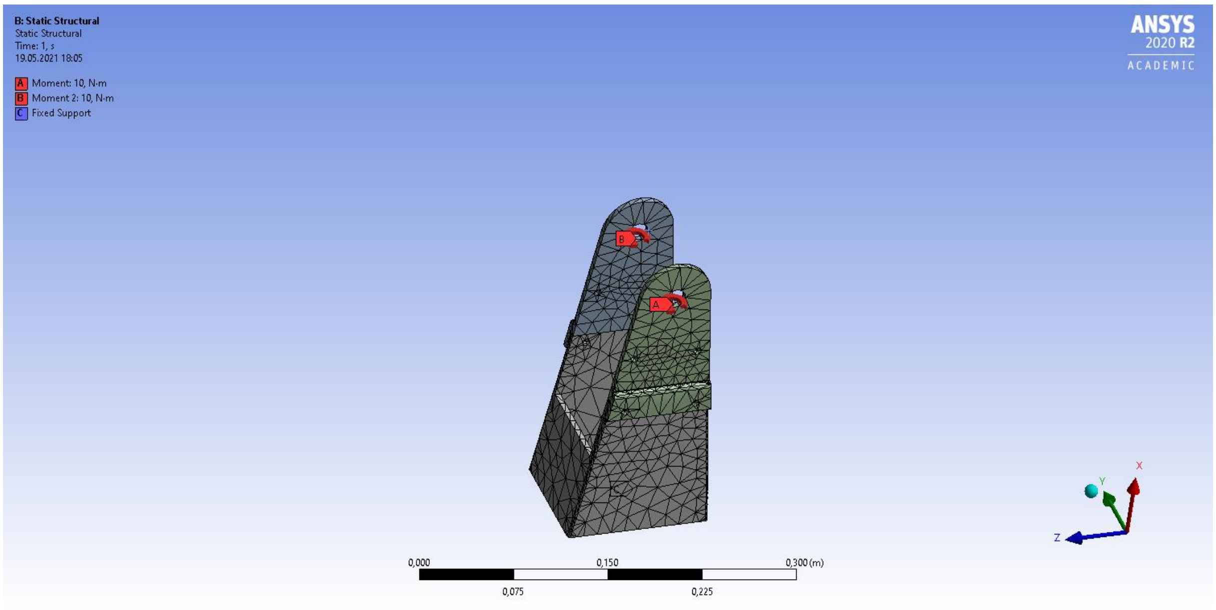


Figure 19 – Moment working on tower.

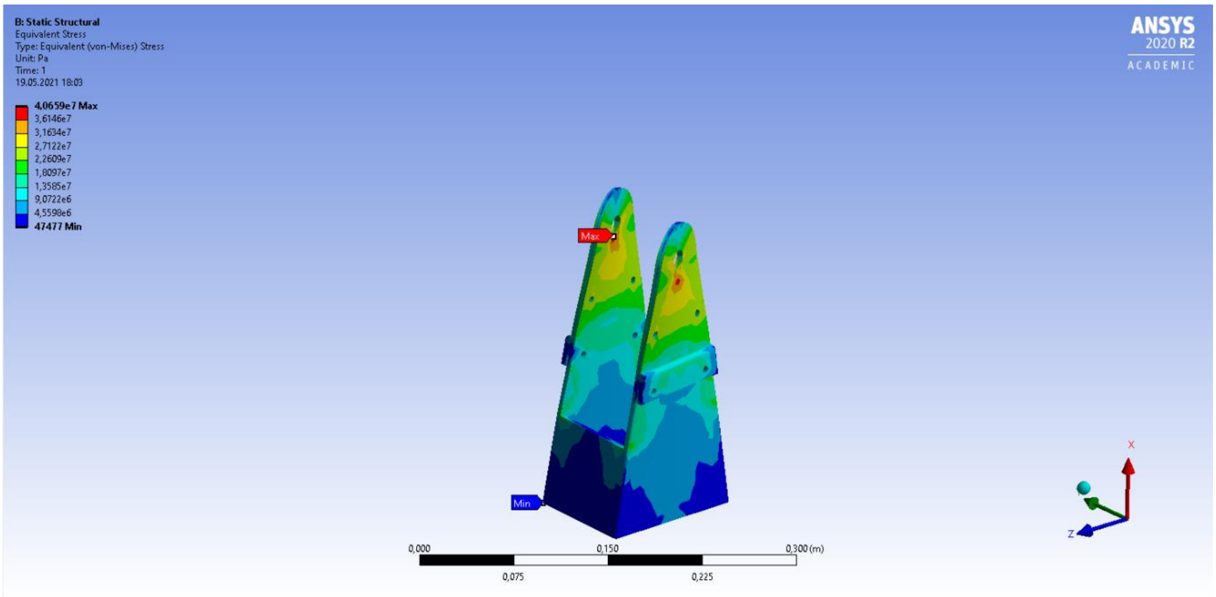


Figure 20 - Von misses stress on Tower

Figures 21-23 shows how the bottom shaft is exceeded to bending moment from the weight and velocity of the crane. However, this is very low, and the shaft is supported by two bearings.

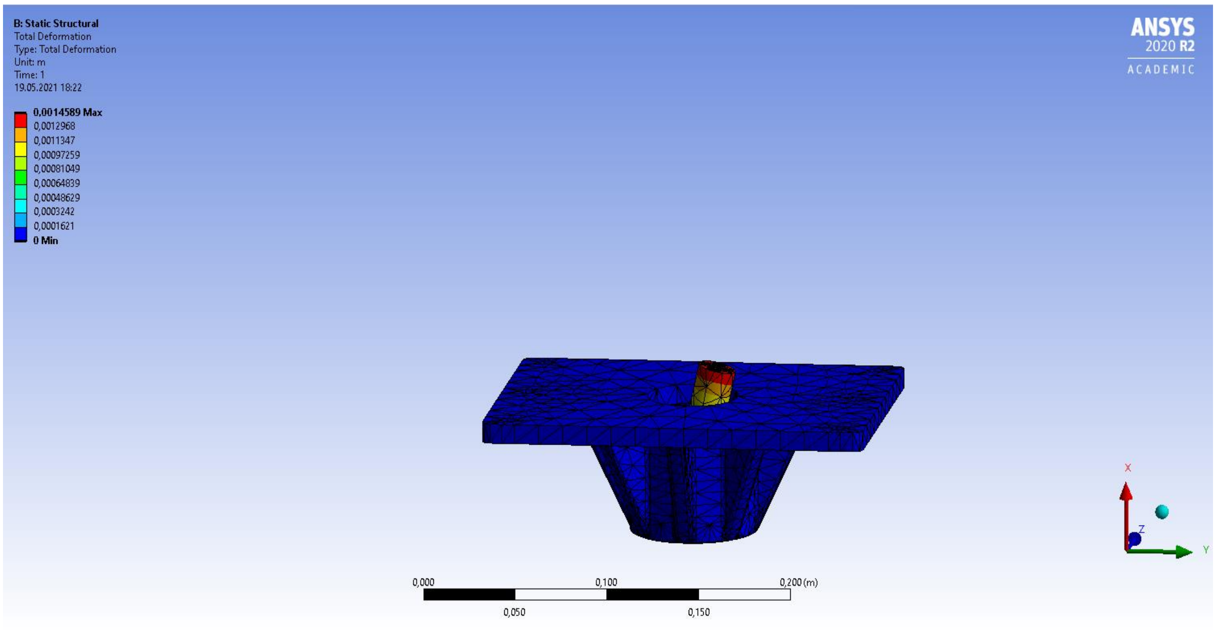


Figure 21 - Deformation on Base

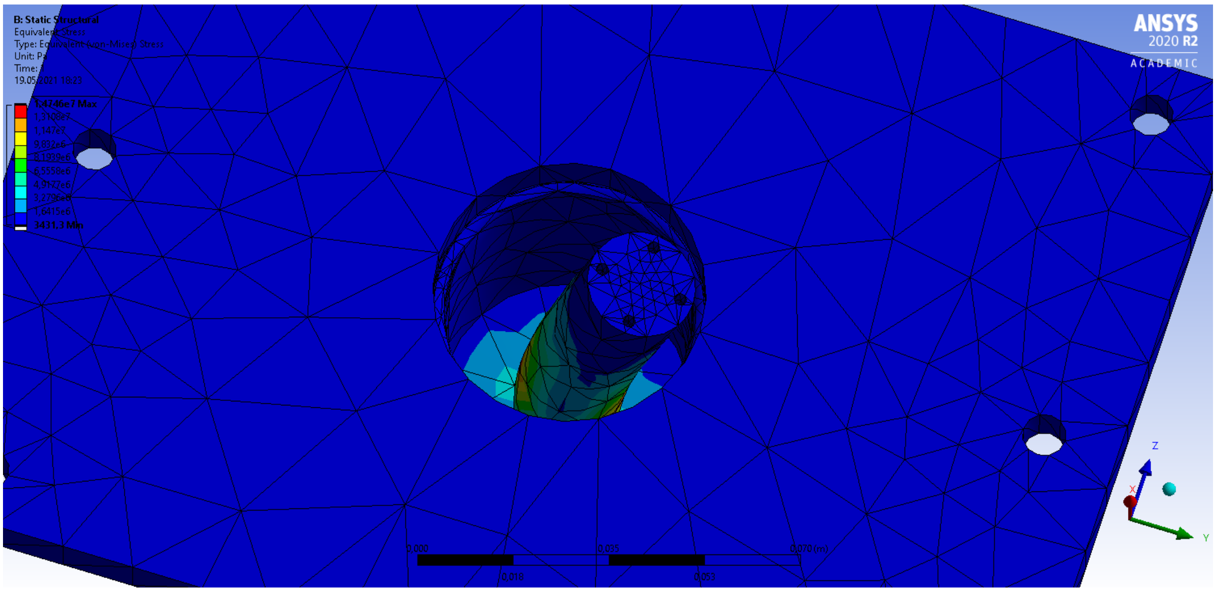


Figure 22 - Von misses stress on Base

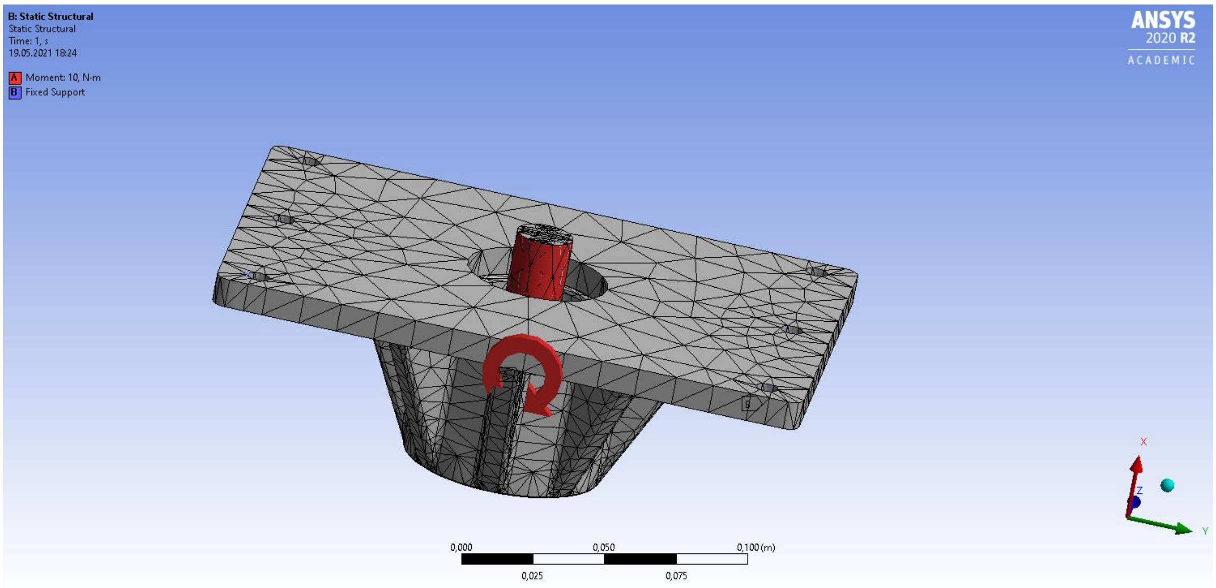


Figure 23 - Forces on Base

As seen in the figures, the numbers are not very high. This is a result of small parts that are made of plastic. Because of the light weight and all other constructive properties of the PLA, shows it that it is more than strong enough for this crane.

3.5 Software and control system

The crane is programmed to be controlled by an Xbox one controller. To make this possible the crane will be remotely operated through Wi-Fi. A laptop running Ubuntu reads user input data from the two joysticks on an Xbox one controller. The data received from the controller is of the short int type and as such is in the range of +-32,768. The value read is directly proportional to how much the user pushes on the joysticks.

The value read from the controller is used to incrementally move the end point of the crane by use of spherical coordinates. The three Cartesian coordinates of the point $[x, y, z]$, are then sent to the Arduino board and are assumed to be relative to the top joint of the tower. This system, opposed to just translating the values of the controller directly to each joint, is in anticipation of more advanced control systems that will be deployed. The needed angle for each joint is then calculated using trigonometry:

$$\theta_T = \sin^{-1}\left(\frac{x}{L_T}\right) \tag{1}$$

In equation (1), LT is the absolute distance of the point in the XY plane relative to the tower. The angle of the main boom is calculated in two steps.

$$\theta_{MB} = \sin^{-1}\left(\frac{z}{L_{BS}}\right) + \cos^{-1}\left(\frac{(L_{MB}^{z} + L_{BS}^2 - L_B^2)}{2L_{MB}L_{BS}}\right) \tag{2}$$

LBS is the distance from the top of the tower to the point in the plane spanned by LT and the z coordinate.

$$\theta_B = \cos^{-1}\left(\frac{(L_{MB}^2 + L_B^2 - L_{BS}^2)}{2L_{MB}L_B}\right) \tag{3}$$

Using pulse width modulation, the bodies are moved to their desired orientation. As of this moment, the crane is functioning. [12]

4 Discussion

A traditional knuckle boom crane is set as the most desired outcome, but the result could vary due to certain factors throughout the project. Some of the most influential discussion points are listed below.

4.1 Design

The design process is where all preferred characteristics about the crane´s look and functionality are chosen. With a huge amount of different looks on knuckle boom cranes, a lot of different outcomes will be considered. The design will be tested until the desired outcome is finished.

4.1.1 Beam shape

One factor is whether cylindrical or H-beam shaped joints will be used. To get a visual representation of the different ideas, a couple of drafts will be considered in the design process. H-beam shaped joints will allow more room for the servo motors to be mounted inside each link. Choosing the h-beam results in severe reduction of material usage for printing because this option reduces the volume and the need for infill material at each part.

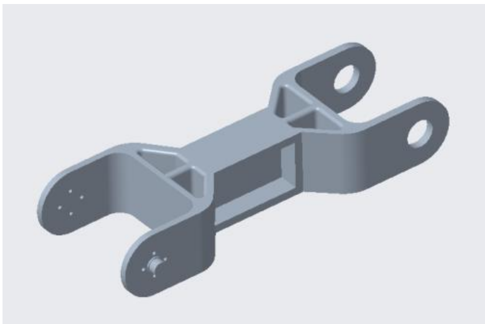


Figure 24 - Main boom

4.1.2 Telescopic joint

It was considered whether a telescopic joint should be used at the end of the boom or not. This would include a fourth servo motor which leads to more weight. Different drafts will be designed and considered. However, a telescopic link is not a necessity for this project, and it might overcomplicate an already advanced project. Whether this option will be used or not, it might be something for the next group of students to develop.

Figure 25 - Crane with telescope function

4.1.3 The base

After connecting the base to the rest of the crane it showed that the base is too weak. When the crane moved, the whole crane wiggled. In conclusion, a stronger, less fragile base must be designed. In Figure 26 the difference between two base examples is shown, where the one to the left is the first draft.

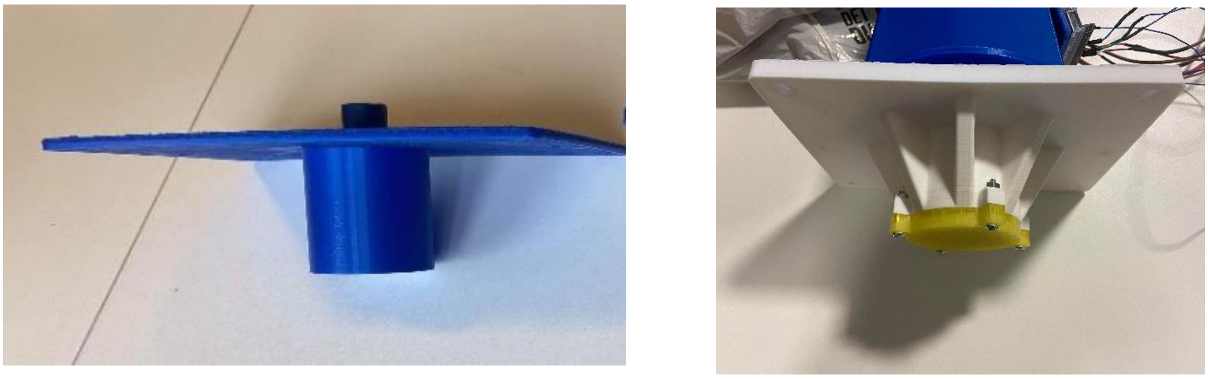


Figure 26 - Different base products

4.1.4 Main boom and knuckle boom

The first crane that was 3D-printed was the one to your left in figure 27. This was made as a result of only having the smaller 3D-printer where more parts needed to be printed and connected. This led to, to big booms relative to the tower. This was a design that did not look like a real-life knuckle boom crane because of its unlikely dimensions.

Therefor both main boom and knuckle boom is replaced by two elegant and smaller bodies. These new bodies are lighter weighted, yet more than robust enough for lifting its own weight and load. This change is completed due to the lifting capacity of the servo motors. It is important to reduce unnecessary weight considering the maintenance and efficiency of the servo motors.

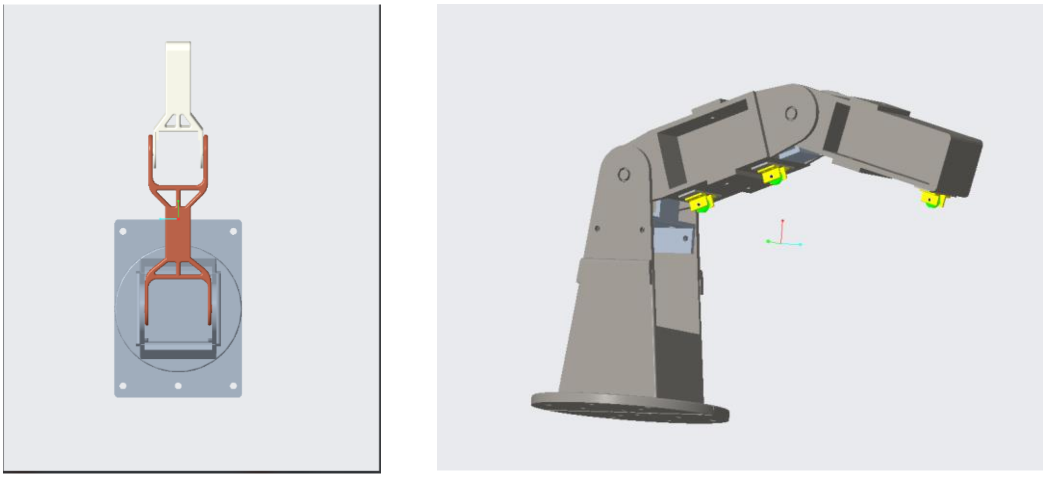


Figure 27 - Difference between last and first draft

4.1.5 Power supply option

Another factor is whether the design should include traditional hydraulic cylinders and pumps as in a full-scale model. Considering that the project regards a scaled model, makes it too expensive. For this reason, it is in favoring that electrical servo motors, rotary sensors and an Arduino circuit card will be used.

4.1.6 Servo motors

For the process of choosing servo motors there is a couple of different solutions to discuss. One of the solutions is whether the linked arms should be connected with the servo motors through wires. This originally came up as an idea because the students were worried the weight of the motors would have significant effect on the boat's stability in the water. A downside is that wires would stretch and cause time delay. Therefore, it would be difficult to make the crane react accurately enough. Also, mass calculations revealed that the servo motors are of low mass compared to the crane´s mass.

If servo motors are the ideal solution, placing them will be a challenge. One solution is to mount the motors such that the shaft is connected to the rotating part, and the motor mount is attached to the rigid part of the crane. For the rotating link at the bottom of the crane, it was discussed whether to use a regular electrical motor or a servo motor. It turned out that the 180 degrees rotating angle for the servo motor is sufficient.

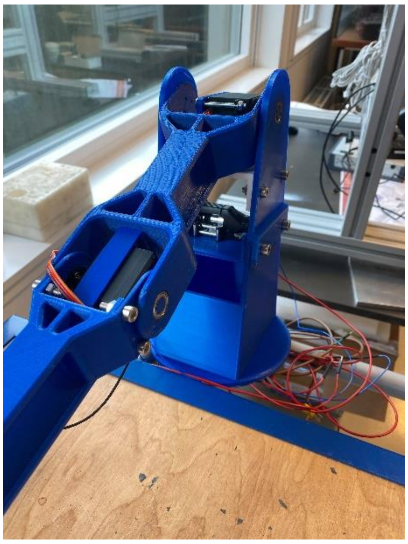


Figure 28 - Servo Motors

4.2 3D-printer

At the origin of this project the students where restricted from making parts longer than 150 mm in each direction of a XYZ-coordinate system. This complicated the design process as the bodies had to be divided into smaller pieces. For this reason, multiple connection methods were discussed. However, the final solution ended up being bolts and nuts to hold the parts in place, as well as using well suited glue. Glue made for connecting plastic to metal was used for connecting the bearings in each link.

Later in the project, the students were given access to a 3D-printer with larger building area and two extruders. The new printer was the Flashforge creator 3, a valued upgrade from the Flashforge Adventurer 3 the students originally had access to. This increased the possibilities regarding the size of the 3d-printed components. As a result of this, new joints were constructed for the "crane arms" which enabled us to make the whole construction in fewer parts.

5 Conclusion

As a final conclusion, a scaled knuckle boom crane was designed which is shown in figure 29. The crane is driven by a programmed Xbox controller. The crane can turn more than 150 degrees around the vertical axis and the booms can operate in angles up to 120 degrees. Three servo motors and one winch are powered trough battery packs that can run up to 3 hours. The winch that is connected can take loads up to 3kg. When all is said and done the crane successfully handles all applied torque and forces.

With all this in mind, all the requirements were performed and achieved. The crane is fully functional and can be used in future testing and research on Smart Ships and cranes. Hopefully, this project will be looked at as inspiration and can be used by other engineers to achieve the goal, singlehandedly counteracting to all dynamic forces. All in all, a very instructive, enjoyable and challenging task.

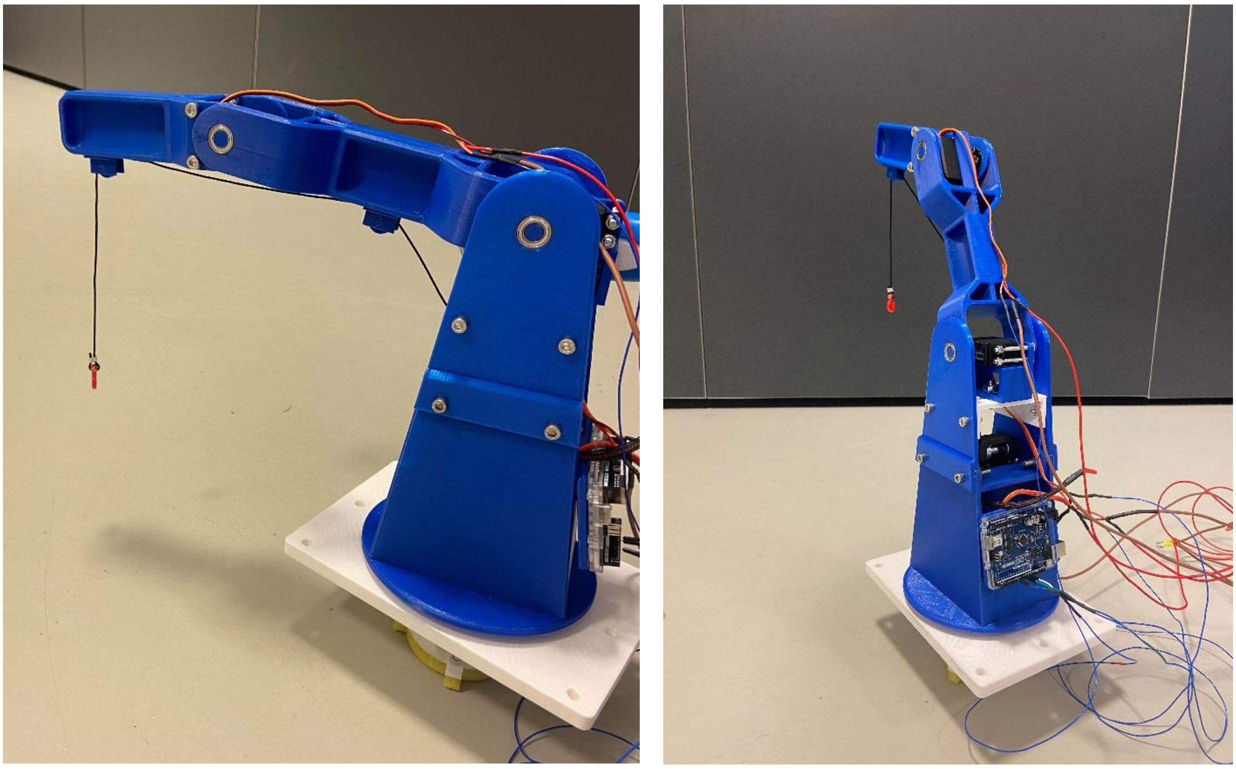


Figure 29 - Final Product

References

[1] PSI-USA, Inc. dba Domain Robot, "PTC," PTC inc., 9 Desember 1993. [Online]. Available: https://www.ptc.com/en/products/creo/parametric.

[2] P. T. R. J. H. Ø. R. T. R. a. I. T. J. Alexander Jacobsen Jardim, "Modeling Crane-Induced Ship Motion Using the Moving Frame Method," *ASME. J. Offshore Mech. Arct. Eng,* October 2019.

[3] M. V. E. &. R. T. Eia, ""Modeling a Knuckle-Boom Crane Control to Reduce Pendulum Motion Using the Moving Frame Method," *Proceedings of the ASME 2019 International Mechanical Engineering Congressand Exposition. Volume 4: Dynamics, Vibration and control,* 11-14 November 2019.

[4] T. G. E. H. A. K. M. G. D. T. T. A. Rykkje, "DV. "Production and Analytics of a Multi-Linked Robotic System." Proceedings of he ASME 2019 International Mechanical Engineering Congress and Exposition. Volume 4: Dynamics, Vibration, and Control. Salt Lake City, Utah, USA. November 11–14, 2019.," 11-14 November 2019.

[5] I. TJ, "The moving frame method in dynamics: Reforming a curriculum and assessment," *International Journal of Mechanical Engineering Education. 2018;46(2):158-191.doi:10.1177/0306419017730633,* 2018.

[6] CSC CORPORATE DOMAINS, INC, "Ansys," 15 November 1994. [Online]. Available: https://www.ansys.com/startup-program?utm_adposition=1t1&utm_adposition=1t1.

[7] "Clas Ohlson," [Online]. Available: https://www.clasohlson.com/no/FlashForge-Adventurer-3,-3D-skriver/p/38-8870.

[8] "Clas Ohlson," [Online]. Available: https://www.clasohlson.com/no/FlashForge-Guider-II-3D-skriver/p/38-8466.

[9] T. Rogers, "Creative Mechanisms," 7 October 2015. [Online]. Available: https://www.creativemechanisms.com/blog/learn-about-polylactic-acid-pla-prototypes.

[10] SKF, "SKF," 2018. [Online]. Available: https://www.skf.com/binaries/pub12/Images/0901d196802809de-Rolling-bearings---17000_1-EN_tcm_12-121486.pdf.

[11] "Arduino UNOWi-FiRev2," Arduino, [Online]. Available: https://store.arduino.cc/arduino-uno-wiFi-rev2.

[12] A. B. E. H. B. H. Rykkje. Thorstein "Pending publication", "Production of a Knuckle-boom crane model for testing in a wave tank: An assessment into mechatronics," *IMECE2021-72677,* 1-5 November 2021.

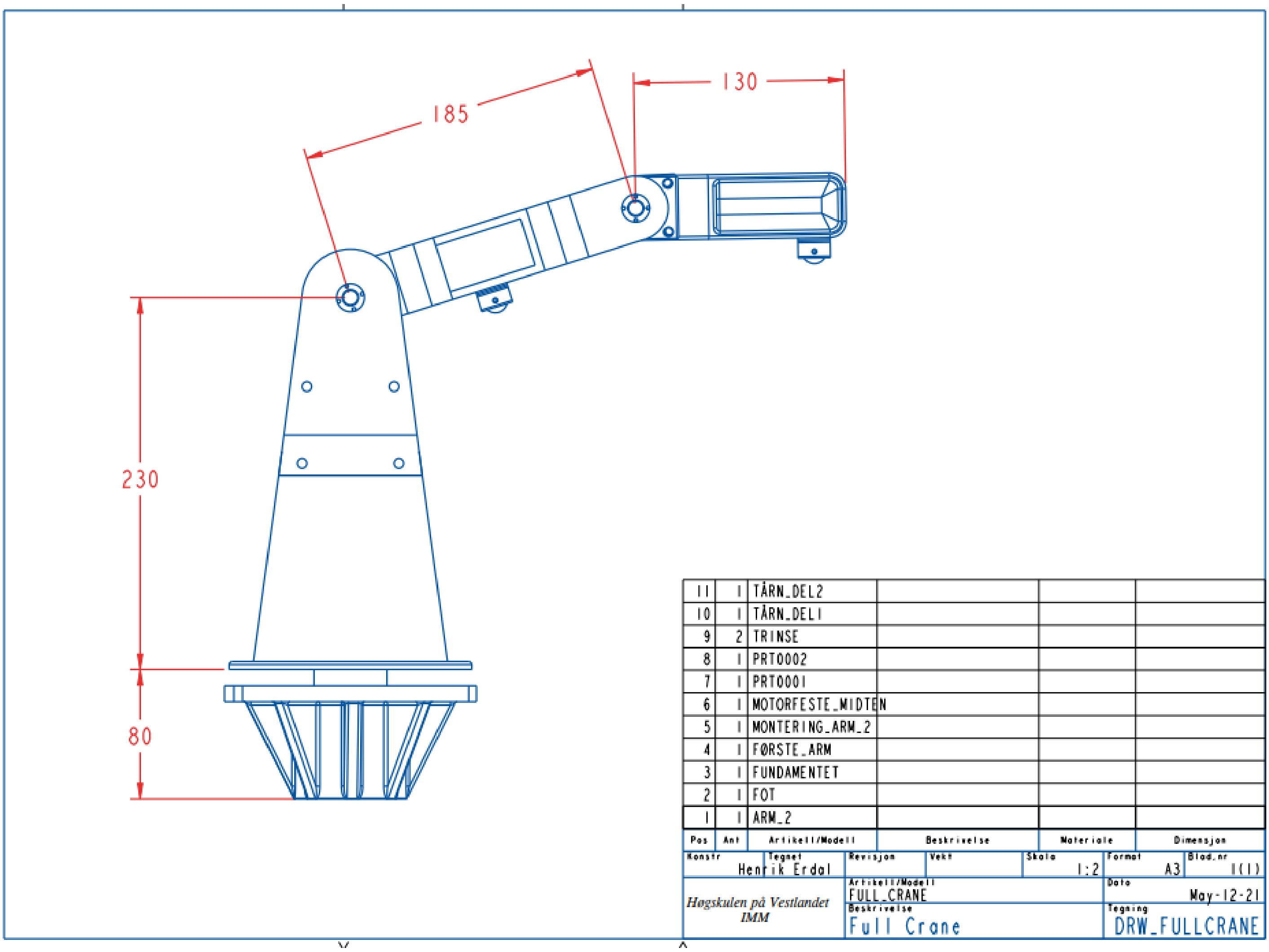
[13] PDR Ltd. d/b/a PublicDomainRegistry.com, "Heila," Heila, 13 July 1999. [Online]. Available: https://www.heila.com/services/knuckle-boom-marine-cranes/.

Table of Figures

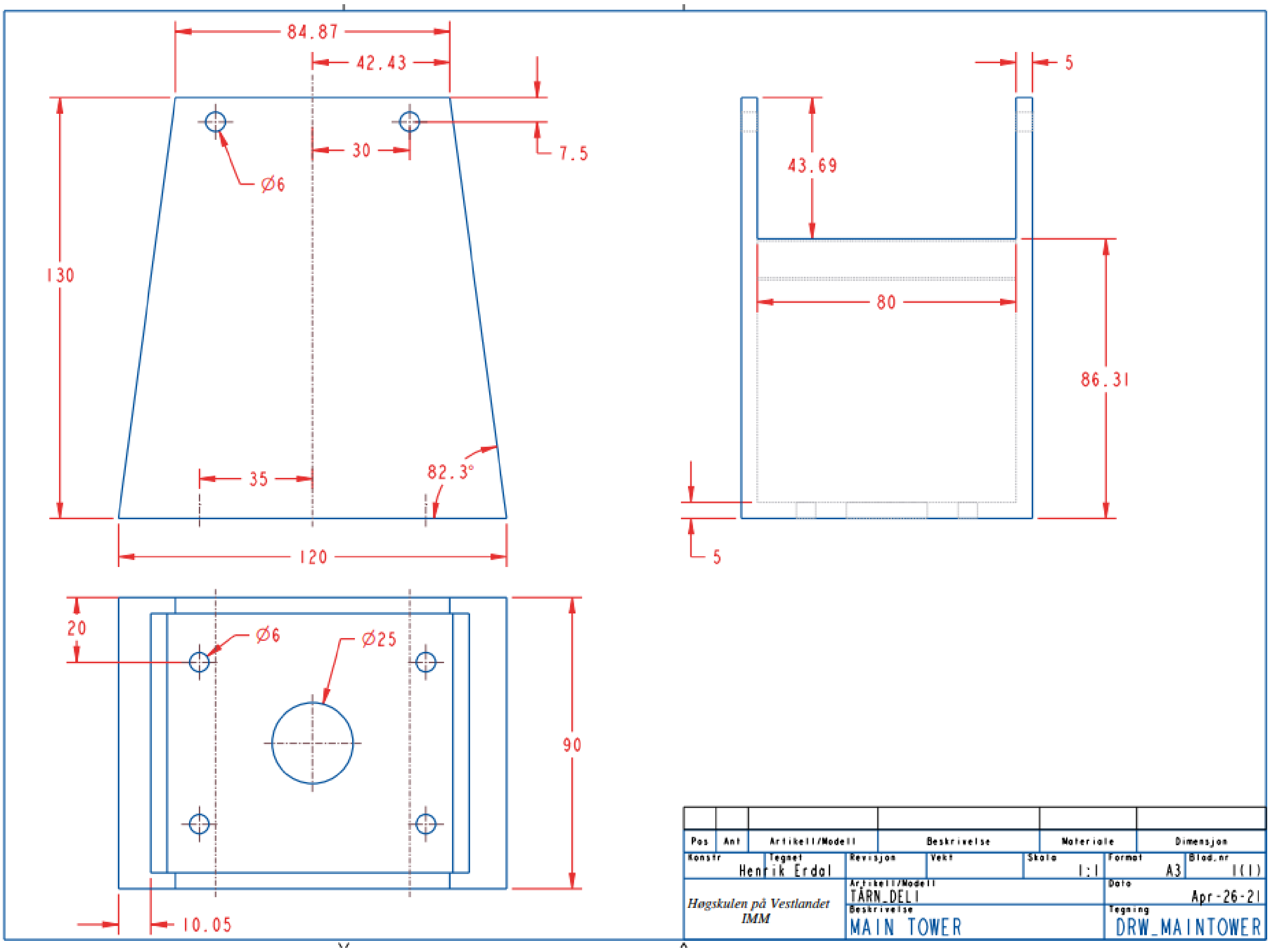
Table of tables

Appendix A - Drawings

Full Crane:

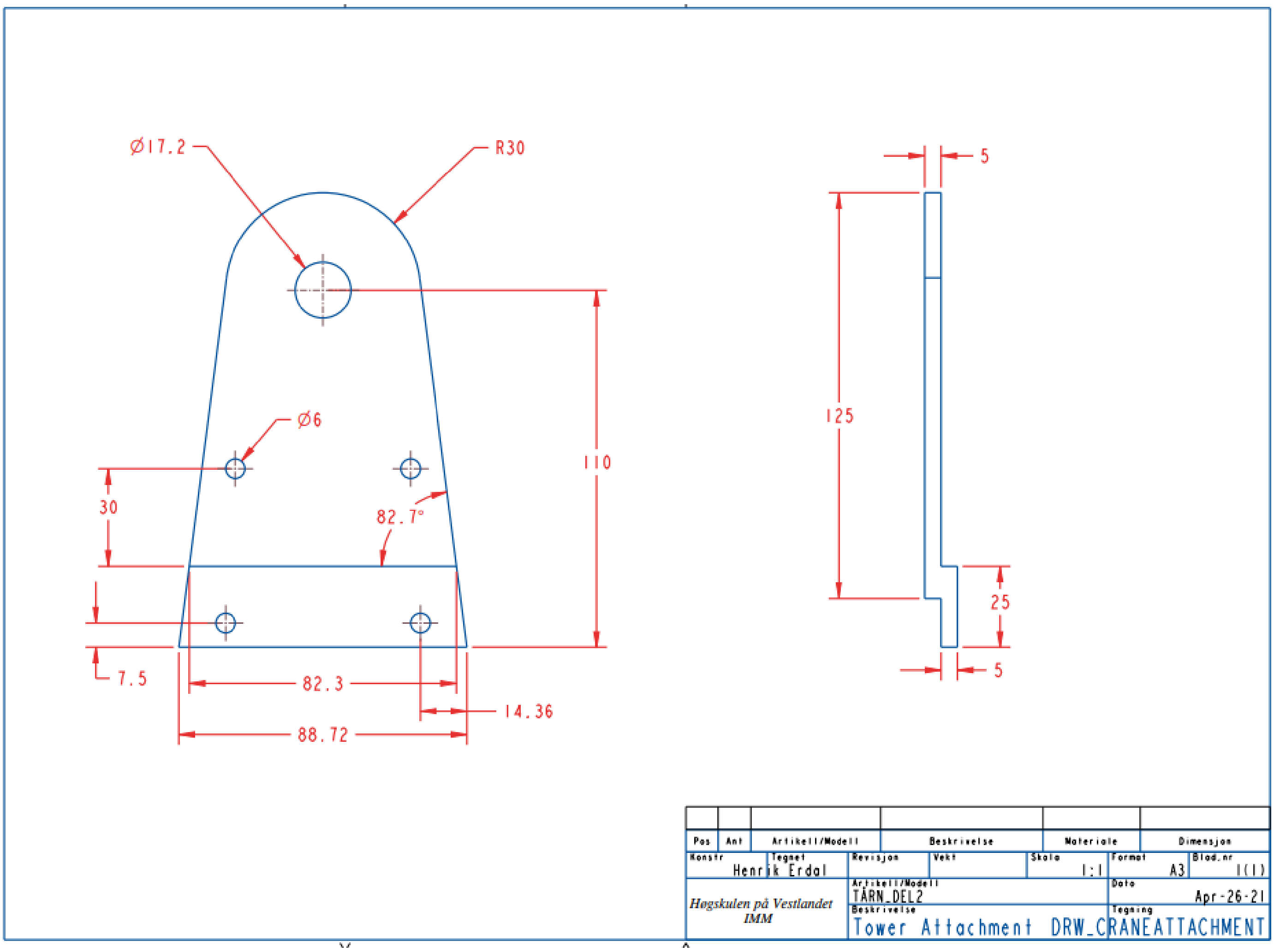


Tower:

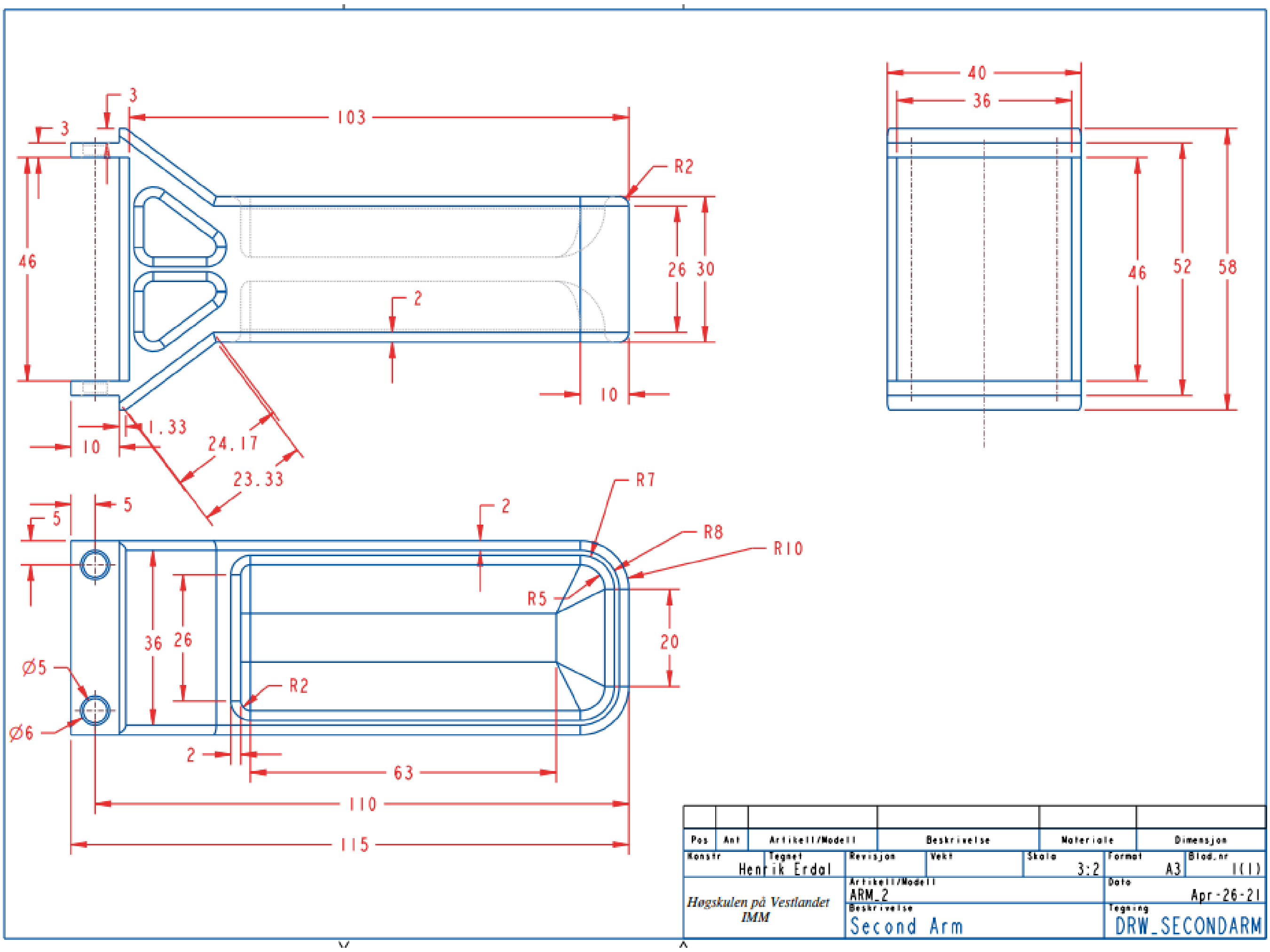


84.87
42.43
7.5
30
Ø6
130
35
82.3°
120
20
Ø6
Ø25
90
10.05
5
43.69
80
86.31
5
Pos
Ant
Artikell/Modell
Beskrivelse
Materiale
Dimensjon
Konstr
Tegnet
Henrik Erdal
Revisjon
Vekt
Skala
1:1
Format
A3
Blad.nr
1(1)
Høgskulen på Vestlandet
IMM
Artikell/Modell
TÅRN_DEL1
Dato
Apr-26-21
Beskrivelse
MAIN TOWER
Tegning
DRW_MAINTOWER

Tower Attachment:

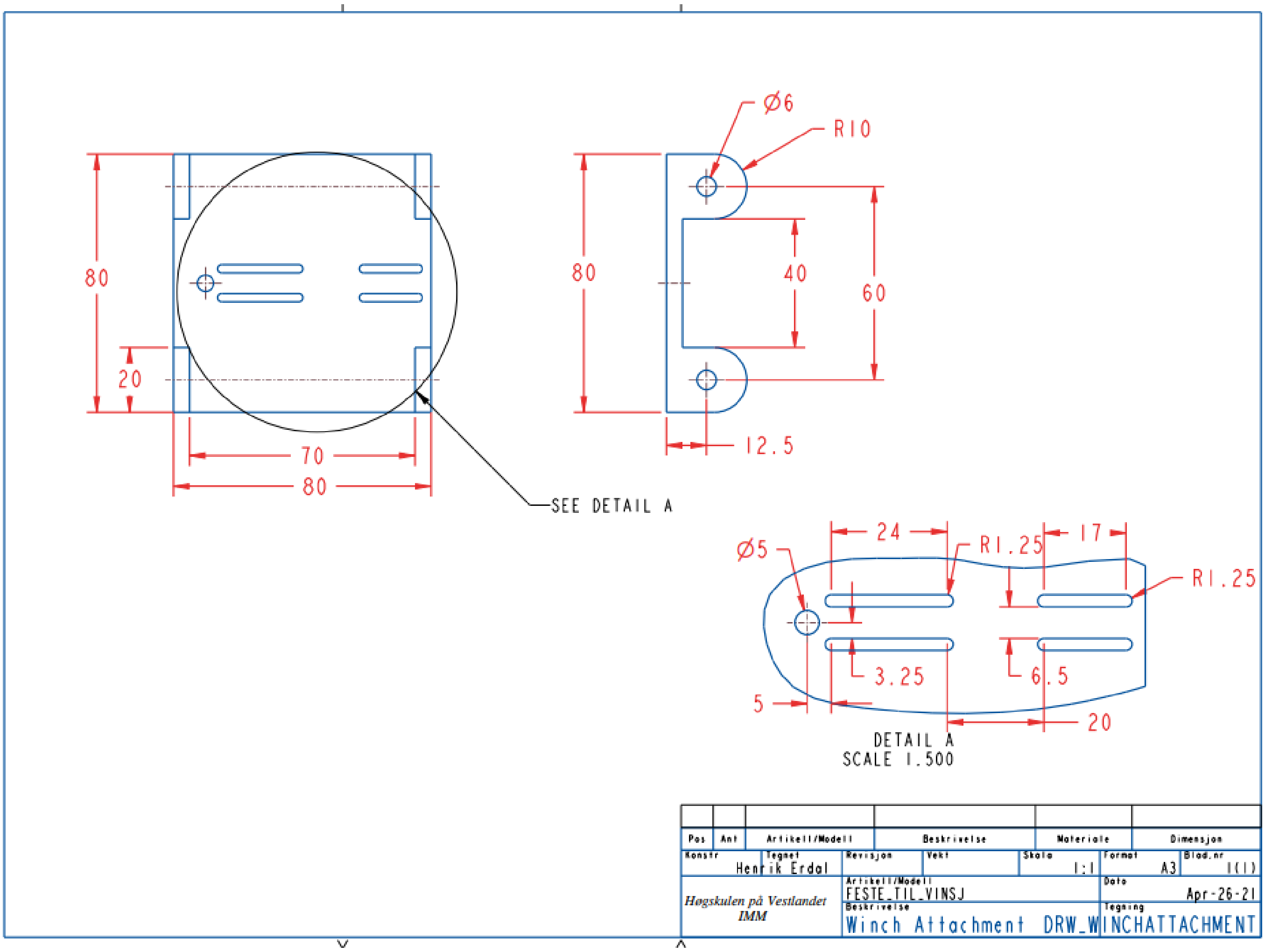


Boom:

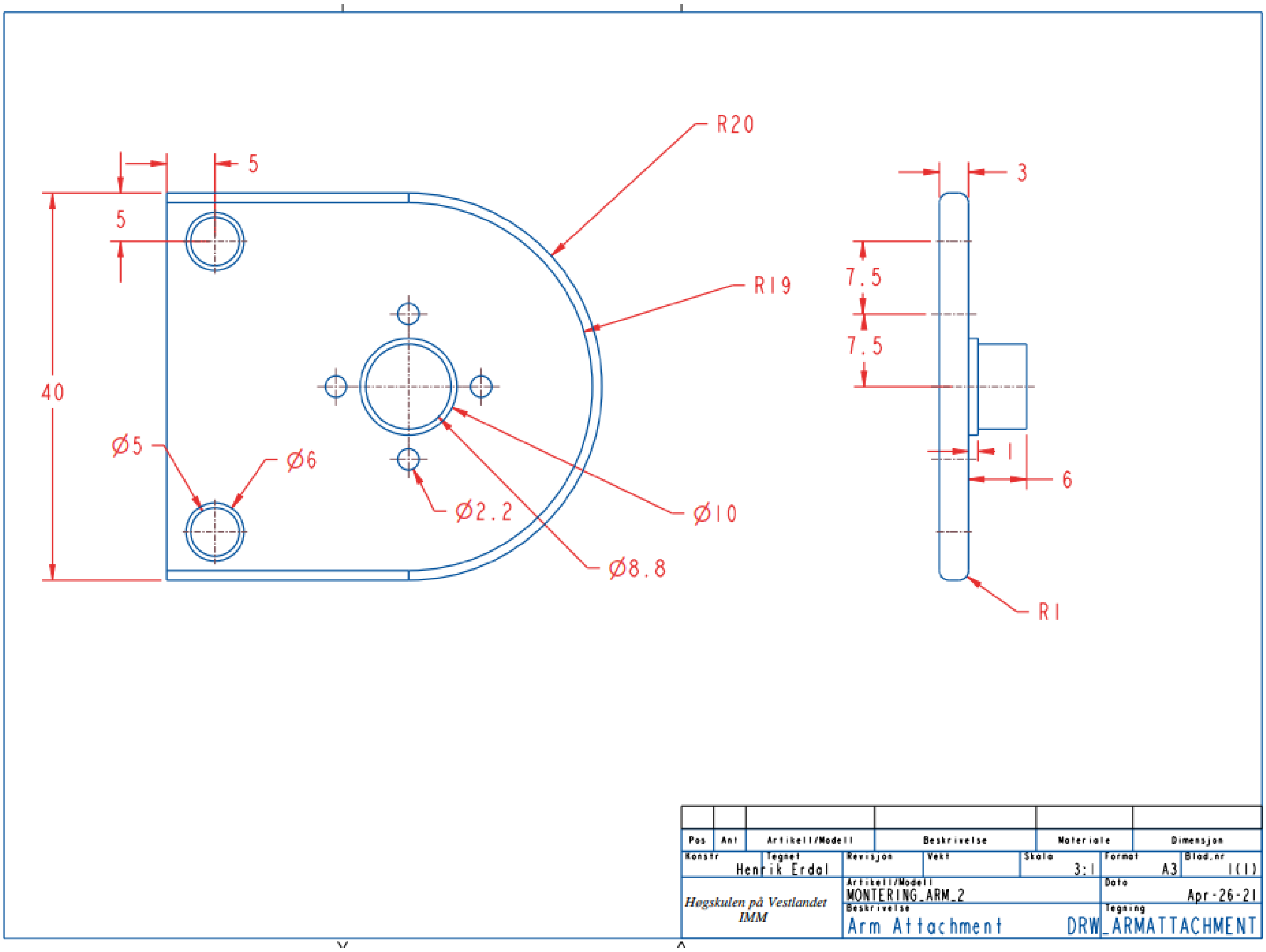


Henrik Erdal
ARM_2
Second Arm
DRW_SECONDARM
Høgskulen på Vestlandet
IMM
3:2
A3
Apr-26-21

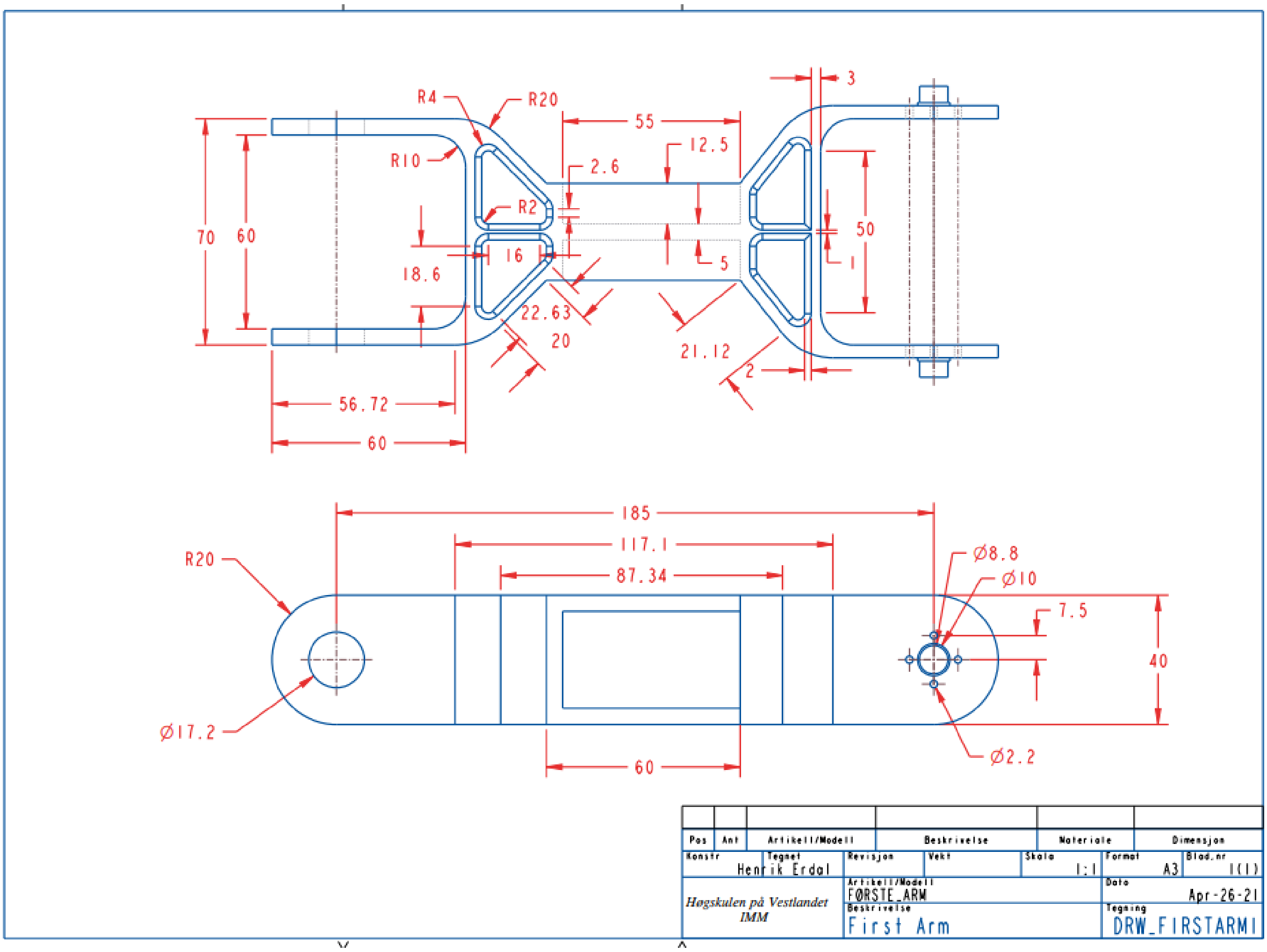
Winch attachment:



Boom attachment:

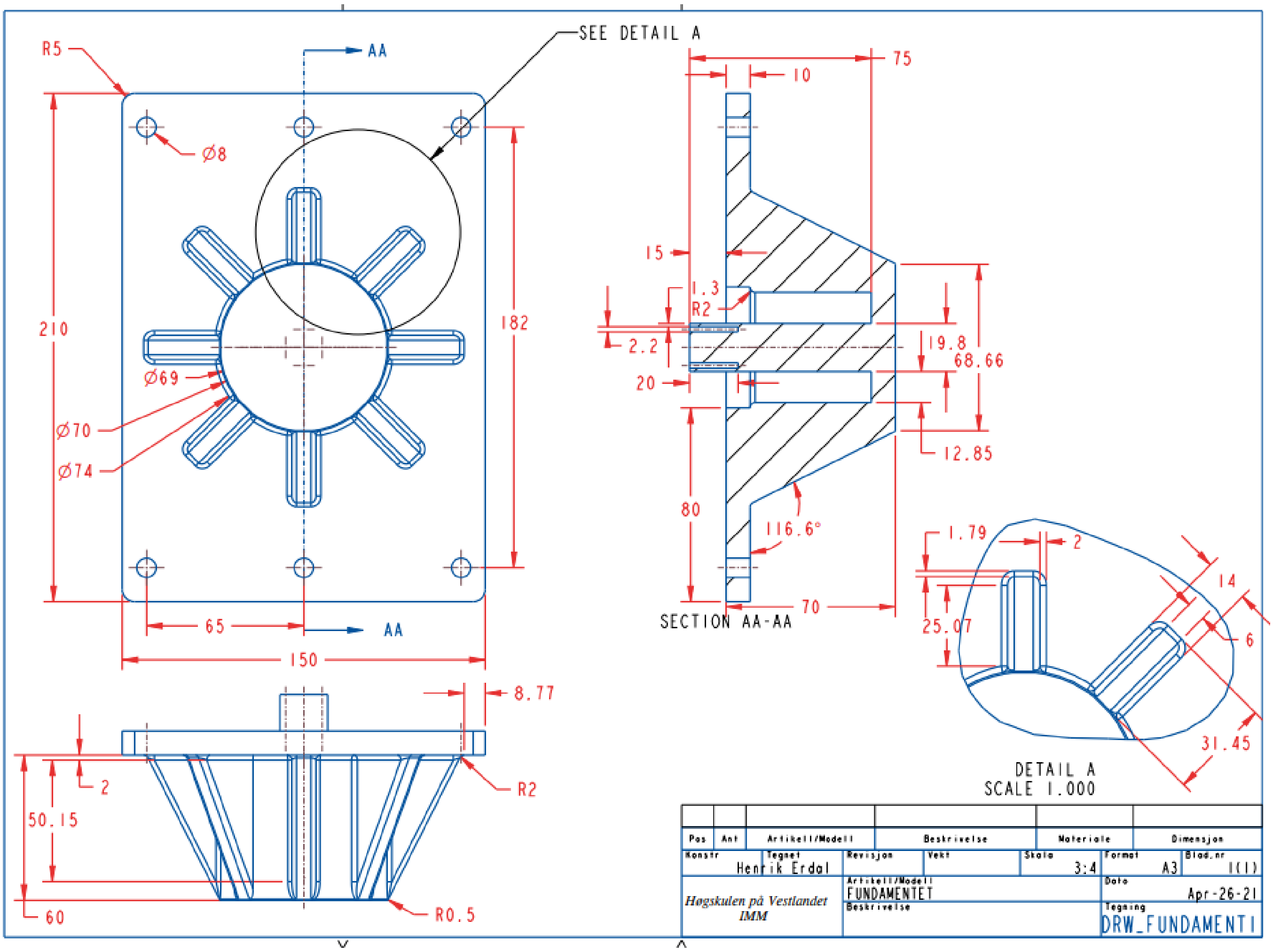


Main Boom:

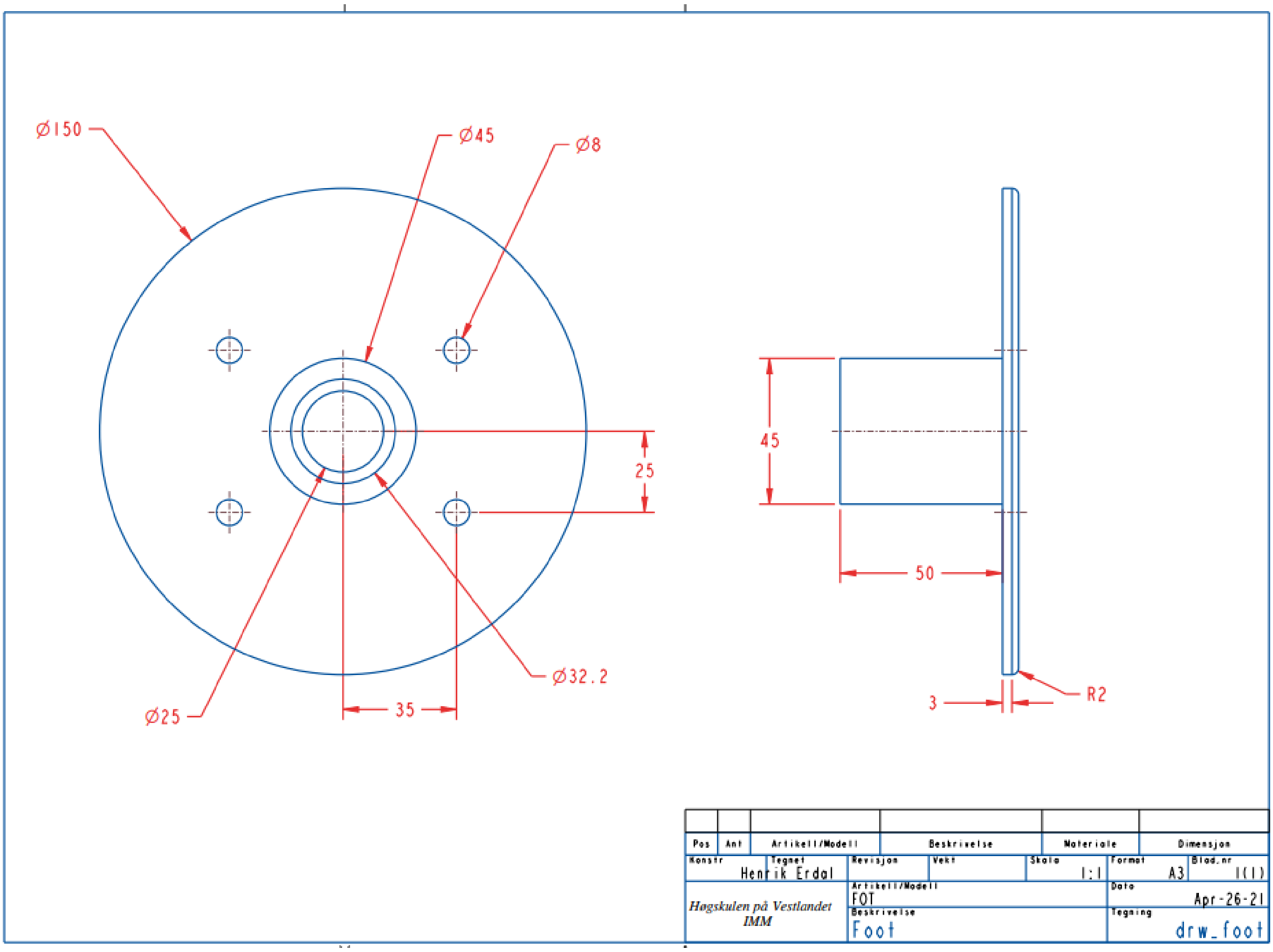


R4
R20
R10
R2
55
12.5
2.6
3
50
1
5
16
18.6
70
60
22.63
20
21.12
2
56.72
60
185
117.1
87.34
R20
Ø17.2
Ø8.8
Ø10
7.5
40
Ø2.2
60
Pos
Ant
Artikell/Modell
Beskrivelse
Materiale
Dimensjon
Konstr
Tegnet
Henrik Erdal
Revisjon
Vekt
Skala
1:1
Format
A3
Blad.nr
1(1)
Høgskulen på Vestlandet
IMM
Artikell/Modell
FØRSTE_ARM
Dato
Apr-26-21
Beskrivelse
First Arm
Tegning
DRW_FIRSTARM1

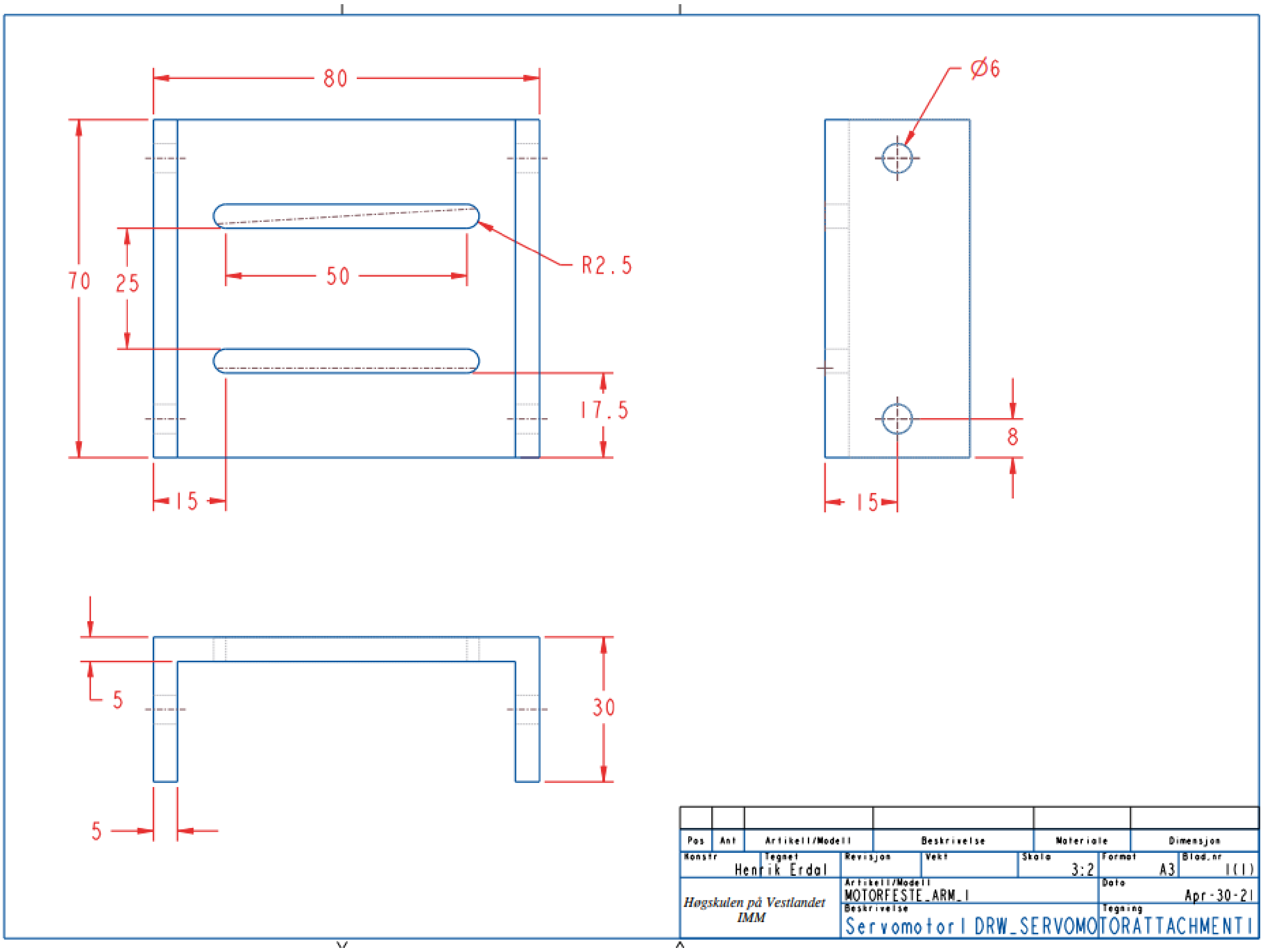
Base:


SEE DETAIL A
R5
AA
Ø8
210
182
Ø69
Ø70
Ø74
65
150
AA
75
10
15
1.3
R2
2.2
20
19.8
68.66
12.85
80
116.6°
70
SECTION AA-AA
1.79
2
14
25.07
6
31.45
DETAIL A
SCALE 1.000
8.77
2
R2
50.15
60
R0.5
Pos
Ant
Artikel/Modell
Beskrivelse
Materiale
Dimensjon
Konstr
Tegnet
Henrik Erdal
Revisjon
Vekt
Skala
3:4
Format
A3
Blad.nr
1(1)
Høgskulen på Vestlandet
IMM
Artikkel/Modell
FUNDAMENTET
Beskrivelse
Dato
Apr-26-21
Tegning
DRW_FUNDAMENT1

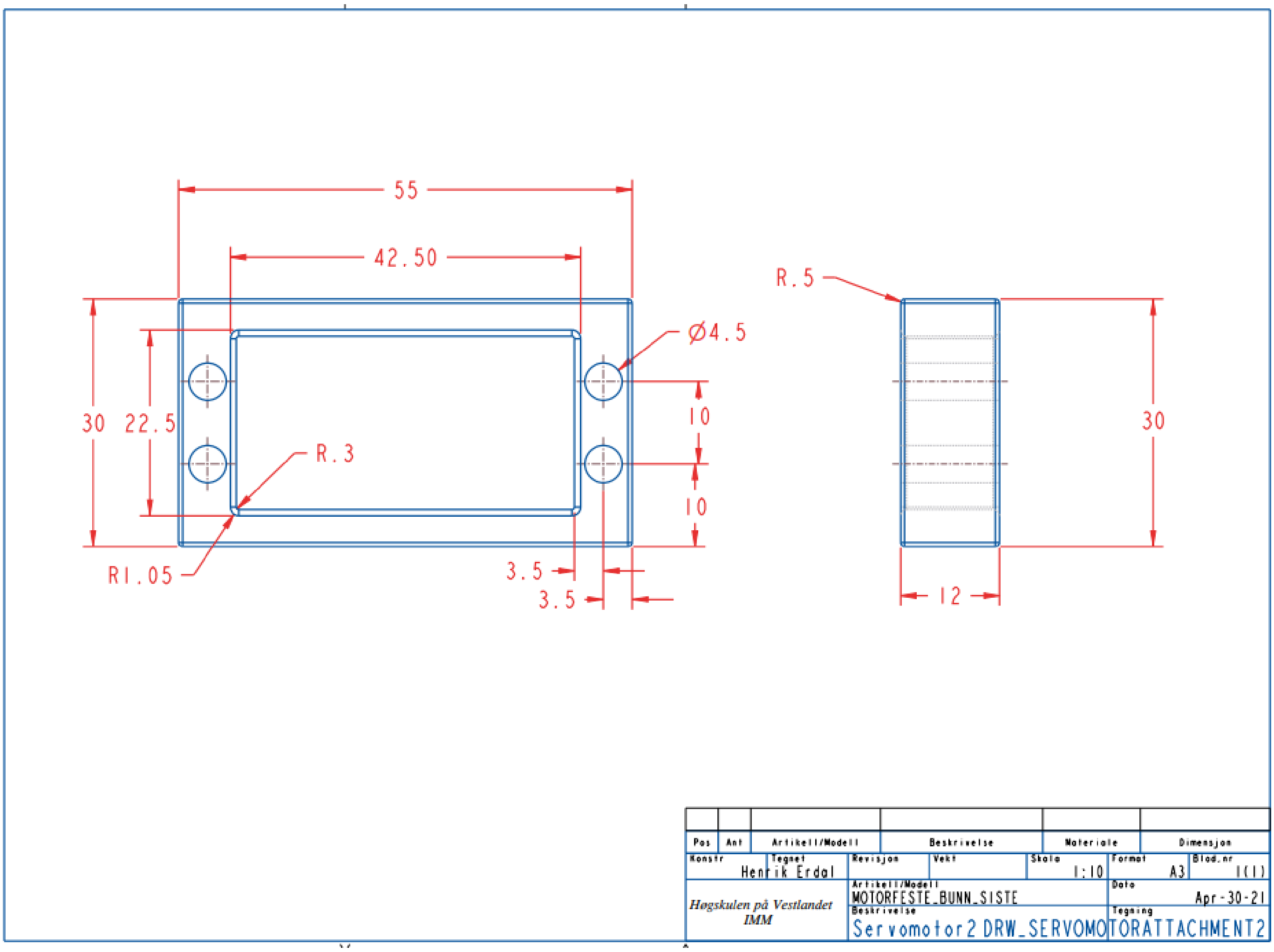
Foot:


Ø150
Ø45
Ø8
25
45
50
Ø32.2
Ø25
35
3
R2
Pos
Ant
Artikell/Modell
Beskrivelse
Materiale
Dimensjon
Konstr
Tegnet
Henrik Erdal
Revisjon
Vekt
Skala
1:1
Format
A3
Blad.nr
1(1)
Høgskulen på Vestlandet
IMM
Artikell/Modell
FOT
Dato
Apr-26-21
Beskrivelse
Foot
Tegning
drw_foot

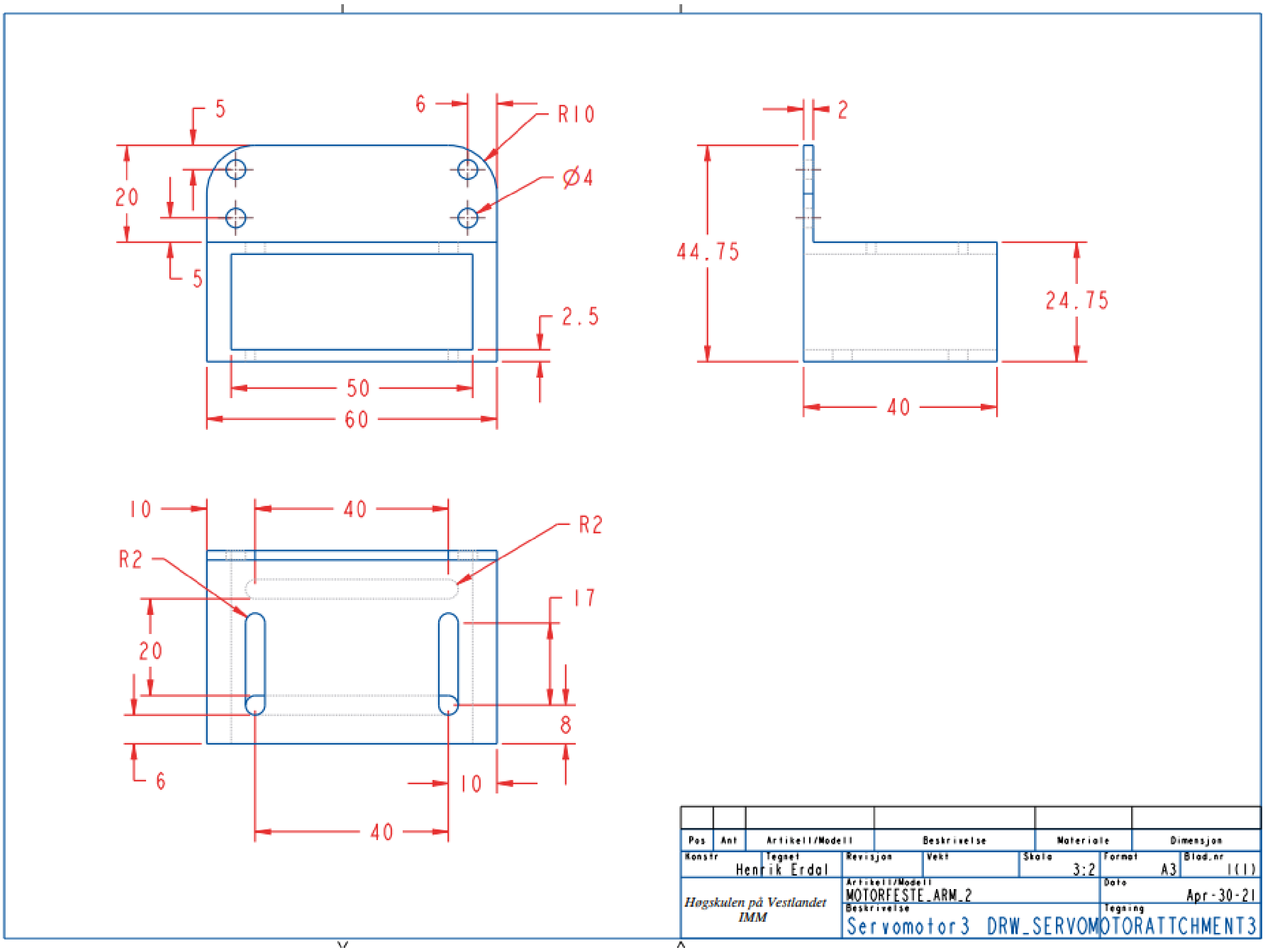
Servo Motor 1 Attachment:



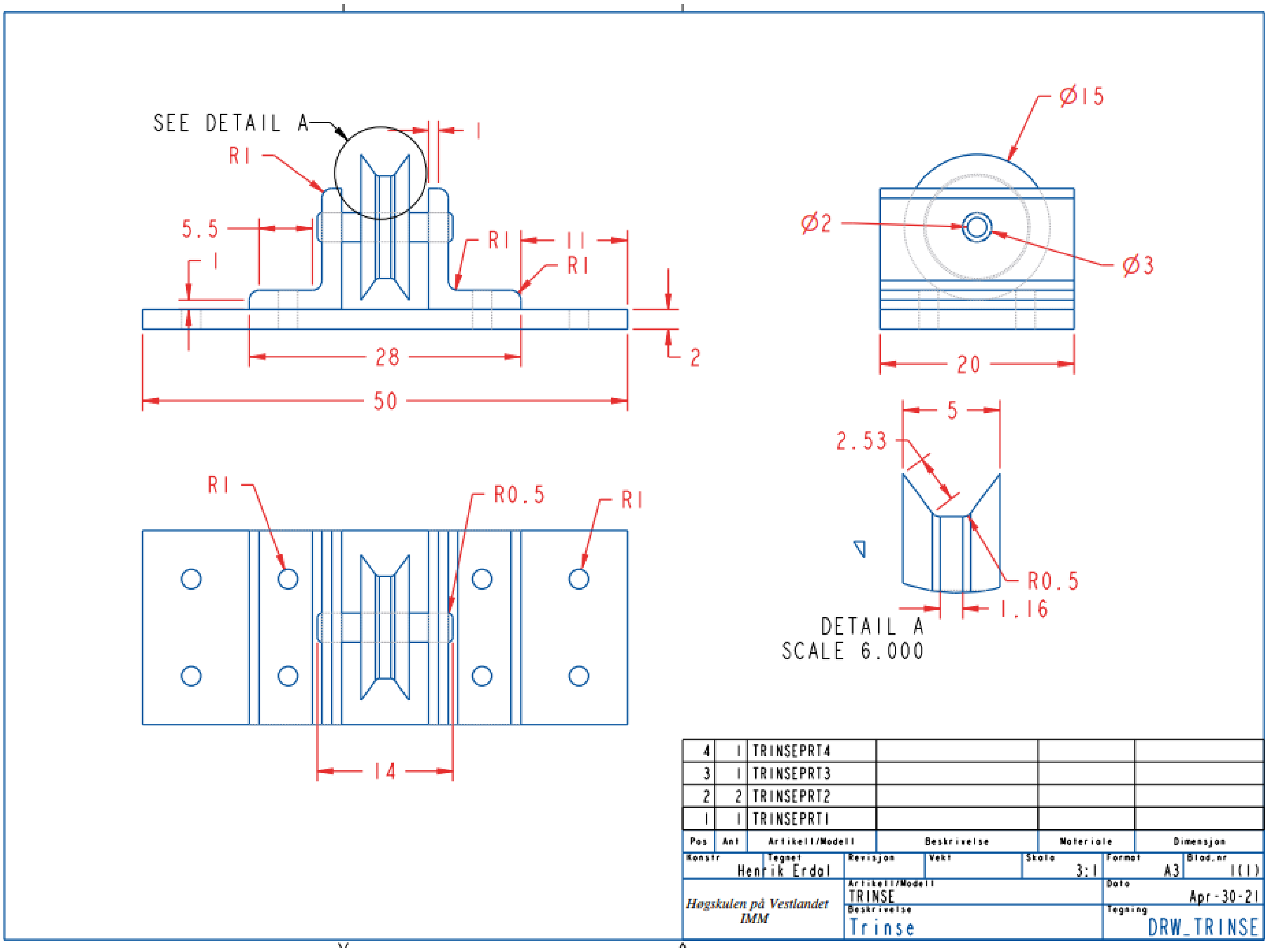
Servo Motor 2 Attachment:



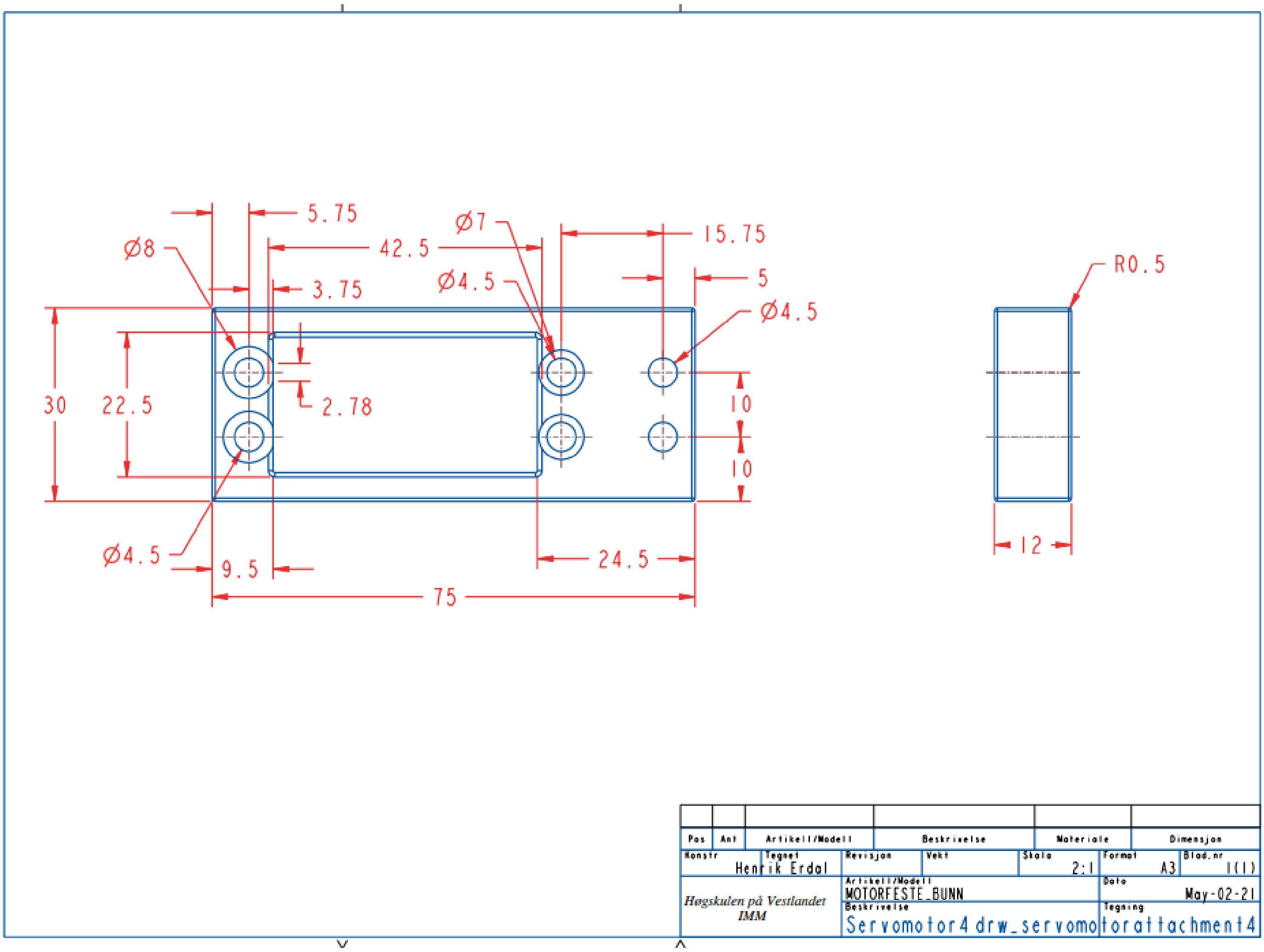
Servo Motor 3 Attachment:



Pulley:


SEE DETAIL A
R1
5.5
1
R1
11
R1
1
2
28
50
Ø15
Ø2
Ø3
20
5
2.53
R0.5
1.16
DETAIL A
SCALE 6.000
R1
R0.5
R1
14
4 1 TRINSEPRT4
3 1 TRINSEPRT3
2 2 TRINSEPRT2
1 1 TRINSEPRT1
Pos Ant Artikkel/Modell Beskrivelse Materiale Dimensjon
Konstr
Tegnet Henrik Erdal
Revisjon
Vekt
Skala 3:1
Format A3
Blad.nr 1(1)
Høgskulen på Vestlandet IMM
Artikkel/Modell TRINSE
Beskrivelse Trinse
Dato Apr-30-21
Tegning DRW_TRINSE

Servo Motor 4 Attachment:



Appendix B – Software C++ (Servomotor Wi-Fi Test)

```
#include <SPI.h>
#include <WiFiNINA.h>
#include <Servo.h>


// Secrets contains the ssid and password.
#include "arduino_secrets.h"


int status = WL_IDLE_STATUS;

WiFiServer server(80);

Servo servoTower;
Servo servoBom;
Servo servoStick;
Servo vinsj;


float angles[3] = {0};
float mappedVal[4] = {0};
int angleData[4] = {0};
int sensorData[3] = {0};


unsigned long prevTime = 0;
unsigned long Time = 0;
unsigned long dt = 0;

float increments[3] = {0};


void setup() {

  Serial.begin(9600);      // initialize serial communication

  //Attatch Servos objects to pins
  servoBom.attach(9); //Green
  servoStick.attach(10);//White
  servoTower.attach(6);//Blue
  vinsj.attach(5);

// Initaial angle positions
angles[0] = 10;
angles[1] = 170;
angles[2] = 10;

//Set the crane to an initital position.
  servoTower.write(angles[0]);
  servoBom.write(angles[1]);
  servoStick.write(angles[2]);

  //Send neutral signal to vinsj, for calibration.
  mappedVal[3] = 90;
  vinsj.write((int)mappedVal[3]);


  // check for the WiFi module:

  if (WiFi.status() == WL_NO_MODULE) {

    Serial.println("Communication with WiFi module failed!");

    // don't continue

    while (true);

  }

  String fv = WiFi.firmwareVersion();

  if (fv < WIFI_FIRMWARE_LATEST_VERSION) {

    Serial.println("Please upgrade the firmware");

  }

  // attempt to connect to Wifi network:

  while (status != WL_CONNECTED) {

    Serial.print("Attempting to connect to Network named: ");

    Serial.println(ssid);                   // print the network name (SSID);

```

```
    // Connect to WPA/WPA2 network. Change this line if using open or WEP network:

    status = WiFi.begin(ssid, pass);

    // wait 10 seconds for connection:

    delay(10000);

  }

  server.begin();                           // start the web server on port 80

  printWifiStatus();                        // you're connected now, so print out the status
}

void loop() {

  WiFiClient client = server.available();   // listen for incoming clients
  if (client) {                             // if you get a client,

    Serial.println("new client");           // print a message out the serial port

    prevTime = millis();
    while (client.connected()) {            // loop while the client's connected
          Time = millis();
          dt= Time-prevTime;

      if (client.available()) {             // if there's bytes to read from the client,

        // Read data from Client
        readInt16(client,angleData,4);

        //Map the data to angular velocity
        mappedVal[0] = (float)map(angleData[0],-32768,32768,-60,60);
        mappedVal[1] = (float)map(angleData[1],-32768,32768,-60,60);
        mappedVal[2] = (float)map(angleData[2],-32768,32768,-60,60);
        mappedVal[3] = (float)map(angleData[3],-32768,32768,30,150);

      }
      //Integrate the angular velocty and add to angular position of the servos.
      angles[0] += mappedVal[0]*((float)dt)*0.001;
      angles[1] += mappedVal[1]*((float)dt)*0.001;
      angles[2] += mappedVal[2]*((float)dt)*0.001;

      //Send mapped value to vinsj
      vinsj.write((int)mappedVal[3]);

      //Check for endpoints on the angle, then write to servo.
      if(0 <= angles[0] && angles[0] <= 179){
      servoTower.write((int)angles[0]);
      }else if((int)angles[0]<0){
        angles[0] = 0;
        servoTower.write((int)angles[0]);
      }else if(angles[0]>179){
        angles[0] = 179;
        servoTower.write((int)angles[0]);
      }

     //Check for endpoints on the angle, then write to servo.
      if(0 <= angles[1] && angles[1] <= 180){
      servoBom.write((int)angles[1]);
      }else if((int)angles[1]<0){
        angles[1] = 0;
        servoBom.write((int)angles[1]);
      }else if((int)angles[1]>180){
        angles[1] = 180;
        servoBom.write((int)angles[1]);
      }

      //Check for endpoints on the angle, then write to servo.
      if(10 <= angles[2] && angles[2] <= 110){
      servoStick.write((int)angles[2]);
      }else if(angles[2]<10){
        angles[2] = 10;
        servoStick.write((int)angles[2]);
      }else if(angles[2]>110){
        angles[2] = 110;
        servoStick.write((int)angles[2]);
      }

      //Save time for next loop
      prevTime = Time;
      }
```

```
    }

    // close the connection:

    client.stop();

    Serial.println("client disonnected");
    delay(1000);

}


void printWifiStatus() {

  // print the SSID of the network you're attached to:

  Serial.print("SSID: ");

  Serial.println(WiFi.SSID());

  // print your board's IP address:

  IPAddress ip = WiFi.localIP();

  Serial.print("IP Address: ");

  Serial.println(ip);

  // print the received signal strength:

  long rssi = WiFi.RSSI();

  Serial.print("signal strength (RSSI):");

  Serial.print(rssi);

  Serial.println(" dBm");

  // print where to go in a browser:


  Serial.print("Ip = ");

  Serial.println(ip);
}

//Read a number of integers from client
void readInt16(WiFiClient client, int* data, int numInts) {
  int dataRead;
  int Size = numInts*sizeof(int);
  int left=Size;
  char* charToRead = (char*)data;
  do{
  dataRead = client.read(charToRead,Size);
  left -= dataRead;
  }while(left>0);

}
```

Appendix C – Software C++ (Joystick Crane Control)

```
#include <fcntl.h>
#include <stdio.h>
#include <unistd.h>
#include <linux/joystick.h>
#include "SendReceive.h"

/**
 * Reads a joystick event from the joystick device.
 *
 * Returns 0 on success. Otherwise -1 is returned.
 */
int read_event(int fd, struct js_event *event)
{
    ssize_t bytes;

    bytes = read(fd, event, sizeof(*event));

    if (bytes == sizeof(*event))
        return 0;

    /* Error, could not read full event. */
    return -1;
}

/**
 * Returns the number of axes on the controller or 0 if an error occurs.
 */
size_t get_axis_count(int fd)
{
    __u8 axes;

    if (ioctl(fd, JSIOCGAXES, &axes) == -1)
        return 0;

    return axes;
}

/**
 * Returns the number of buttons on the controller or 0 if an error occurs.
 */
size_t get_button_count(int fd)
{
    __u8 buttons;
    if (ioctl(fd, JSIOCGBUTTONS, &buttons) == -1)
        return 0;

    return buttons;
}

/**
 * Current state of an axis.
 */
struct axis_state {
    short x, y;
};

/**
 * Keeps track of the current axis state.
 *
 * NOTE: This function assumes that axes are numbered starting from 0, and that
 * the X axis is an even number, and the Y axis is an odd number. However, this
 * is usually a safe assumption.
 *
 * Returns the axis that the event indicated.
 */
size_t get_axis_state(struct js_event *event, struct axis_state axes[4])
{
    size_t axis = event->number / 2;

    if (axis < 4)
    {
        if (event->number % 2 == 0)
            axes[axis].x = event->value;
        else
            axes[axis].y = event->value;
    }

    return axis;
}


#define PORT "80"

int main(int argc,char *argv[]){
```

```
    if(argc<2){
        printf("Please input the ip adress\n");
            return 1;
    }

    int sockfd, check, portnum;
    int angle[4] = {0};
    char ip[16];

    portnum = atoi(PORT);
    printf("Attempting to connect...");
    sockfd = connectToServer(portnum, argv[1]);
    if(sockfd == -1){
        printf("Error in connecting to server");
        return -1;
    }

    const char *device;
    int js;
    struct js_event event;
    struct axis_state axes[4] = {0};
    size_t axis;


    device = "/dev/input/js0";
    js = open(device, O_RDONLY);

    if (js == -1)
        perror("Could not open joystick");

    __u8 buttonCount = get_button_count(js);
    __u8 axisCount =   get_axis_count(js);

    printf("Button count = %d, and axis count: %d\n", buttonCount, axisCount);

/* This loop will exit if the controller is unplugged. */
    while (read_event(js, &event) == 0)
    {
        switch (event.type)
        {
            //Read controller values
            case JS_EVENT_BUTTON:
                printf("Button %u %s\n", event.number, event.value ? "pressed" : "released");
                break;
            case JS_EVENT_AXIS:
                axis = get_axis_state(&event, axes);

            angle[0] = axes[0].x;
            angle[1] = axes[0].y;
            angle[2] = axes[2].x;
            angle[3] = (axes[2].y-axes[1].x)/2;

            if(angle[0]>-4000&&angle[0]<4000){angle[0]=0;}
            if(angle[1]>-4000&&angle[1]<4000){angle[1]=0;}
            if(angle[2]>-4000&&angle[2]<4000){angle[2]=0;}

            //Send controller values
            check = SendSeveralInt16(sockfd,angle,4);
            //Print values to screen for control
            printf("Tower was %d, bom %d and stick %d\nVinsj was %d\n\n",angle[0],angle[1],angle[2],angle[3]);
                    if(check){
                        printf("Error sending angle\n");
                }
            break;
            default:
                /* Ignore init events. */
                break;
        }

        fflush(stdout);

    }

    close(js);
    return 0;



    return 0;
}
```